Sun Java System Portal Server Secure Remote Access 7.2 管理指南

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

文件号码 820–4823
2008 年 5 月

080618@20490

目录

图

表

示例

前言

本指南说明如何管理 Sun Java™ System Portal Server Secure Remote Access 7.2 服务器。

Sun Java System Portal Server Secure Remote Access (SRA) 服务器允许远程用户通过 Internet 安全地访问其组织的网络和服务。此外，SRA 还可为贵组织提供安全的内部门户，从而使所有目标用户（例如员工、业务合作伙伴以及普通公众）都能够访问其内容、应用程序和数据。

本前言包括以下各节：

目标读者

《Sun Java System Portal Server Secure Remote Access 7.2 管理指南》适用于配置和管理 Secure Remote Access 服务器的用户。

《Sun Java System Portal Server Secure Remote Access 7.2 管理指南》假定您是在管理 UNIX 系统和 TCP/IP 网络方面具有丰富经验的网络或系统管理员。即使不是超级用户，您也可以访问所需的机器，以安装 Secure Remote Access 服务器的各种组件。然而确需必需的管理权限才能执行其他操作，如配置用户和服务。

阅读本书之前

Portal Secure Remote Access 服务器管理员应了解以下技术：

- Sun Java System Portal Server
- Sun Java System Directory Server
- Sun Java System Access Manager

- 您的 Web 容器，例如：
  - Sun Java System Application Server 8.2
  - Sun Java System Web Server 7.0
- 您的操作系统
- 基本的 UNIX® 管理过程
- 轻量目录访问协议 (Lightweight Directory Access Protocol, LDAP)
- 远程 Portlet 的 Web 服务 (Web Services for Remote Portlets, WSRP)

您还需达到以下要求才能编写重写器规则：

- 了解超文本标记语言 (Hypertext Markup Language, HTML) 和 HTML 标记
- 熟悉 JavaScript™ 知识
- 基本了解可扩展标记语言 (Extensible Markup Language, XML)

本书的结构

本书的组织方式如下：

- 第 1 部分
  - 第 1 章说明 Sun Java System Portal Server 与 Portal Server Secure Remote Access 之间的关系。
  - 第 2 章说明与网关相关的概念和管理网关的任务。
  - 第 3 章说明可让用户无需解析 Web 页面即可通过网关访问内联网 Web 页面的 Proxylet。
  - 第 4 章说明如何使用 Proxylet 和重写器来通过网关访问内联网 Web 页。
  - 第 5 章说明如何使用 NetFile 访问和操作远程文件系统和目录。
  - 第 6 章说明如何使用 Netlet 在不安全的网络（如 Internet）上安全地运行通用 TCP/IP 服务。
- 第 2 部分
  - 第 7 章说明如何管理对 Portal Server 管理控制台的访问。
  - 第 8 章说明如何从 Portal Server 管理控制台配置网关属性。
  - 第 9 章说明如何使用“重写器”选项卡下的网关服务来执行各种任务。
  - 第 10 章说明管理证书和安装来自证书授权机构的自签署证书。
  - 第 11 章说明从 Portal Server 管理控制台配置 Netlet 属性。
  - 第 12 章说明配置客户机浏览器的 Java 插件，以使 Netlet 可与 PDC 配合使用。
  - 第 13 章说明从 Portal Server 管理控制台配置 Proxylet。
  - 第 14 章说明使用 Portal Server 管理控制台设置 NetFile 选项、权限以及首选项。
  - 第 15 章说明配置用于 Portal Server Secure Remote Access 服务器的各种加速器。
- 第 3 部分

  - 第 16 章说明创建网关配置文件和网关实例的方法。
  - 第 17 章说明维护网络身份的各种方案。
- 附录 A 说明您可以通过 Portal Server 管理控制台为每个 Portal Server Secure Remote Access 组件配置的 Sun Java System Portal Server Secure Remote Access 属性。
- 附录 B 包含调试和其他类型的信息。
- 附录 C 列出您需要在证书管理过程中指定的两个字符的国家/地区代码。

相关书籍

- 《Sun Java System Portal Server 7.2 Deployment Planning Guide》
- 《Sun Java System Portal Server 7.2 Technical Overview》
- 《Sun Java System Portal Server 7.2 管理指南》
- 《Sun Java System Portal Server 7.2 Command-Line Reference》
- 《Sun Java System Portal Server 7.2 发行说明》
- 《Sun Java System Portal Server 7.1 Community Sample Guide》
- 《Sun Java System Portal Server 7.2 Technical Reference》
- 《Sun Java System Portal Server 7.2 Developer's Guide》

《Sun Java System Portal Server 7.2 Technical Overview》 中提供了有关 Portal Server 概念和组件的简介。

其他服务器文档

有关其他服务器文档，请访问：

- Directory Server 文档，位于 `http://docs.sun.com/coll/1224.1` 及 `http://docs.sun.com/coll/1606.1`
- Access Manager 文档，位于 `http://docs.sun.com/coll/1292.2` 及 `http://docs.sun.com/coll/1384.2`
- Web Server 文档，位于 `http://docs.sun.com/coll/1308.3` 及 `http://docs.sun.com/coll/1395.2`
- Application Server 文档，位于 `http://docs.sun.com/coll/1310.3` 及 `http://docs.sun.com/coll/1386.2`
- Web Proxy Server 文档，位于 `http://docs.sun.com/coll/1311.4` 及 `http://docs.sun.com/coll/1579.2`

相关第三方 Web 站点引用

本文档引用了第三方 URL，这些 URL 可提供附加的相关信息。

注 – Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他材料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成或名义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 站点提供关于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.`
`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* 是本地存储的副本。
新词术语强调	新词以及要强调的词。	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。

命令示例中的 Shell 提示符

下表显示了用于 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	`machine_name%`
C shell 超级用户	`machine_name#`
Bourne shell 和 Korn shell	`$`
Bourne shell 和 Korn shell 超级用户	`#`

第 1 部分

Secure Remote Access 服务器组件

◆ ◆ ◆ 第 1 章

1 Portal Server Secure Remote Access 服务器简介

本章介绍 Sun Java™ System Portal Server Secure Remote Access 以及 Sun Java System Portal Server 和 Sun Java System Portal Server Secure Remote Access 组件之间的关系。

本章包括以下主题：

Secure Remote Access 简介

Secure Remote Access 可让远程用户在 Internet 上安全地访问其组织的网络和服务。此外，还可为贵组织提供安全的 Internet 门户，从而使所有目标用户（例如员工、业务合作伙伴以及普通公众）都能够访问其内容、应用程序和数据。

Secure Remote Access 软件可提供基于浏览器的安全远程访问，以从任意远程设备访问门户内容和服务。Secure Remote Access 是一种安全访问解决方案，用户无需客户机软件就可从任意装有启用了 Java™ 技术的浏览器的设备对其进行访问。与 Portal Server 的集成可确保用户对具有访问许可权的内容和服务进行安全加密式的访问。

Secure Remote Access 软件适用于部署高安全远程访问门户的企业。这些门户注重的是内联网资源的安全性、保护性以及保密性。Secure Remote Access 的体系结构正适合这些类型的门户。借助 Secure Remote Access 软件，用户可以通过 Internet 安全地访问内联网资源，而不会使这些资源公诸于 Internet。

Portal Server 能够以两种模式运行，即以下各节中所述的 “开放模式” 和 “安全模式”。

开放模式

在开放模式下，安装 Portal Server 时不会安装 Secure Remote Access。尽管在此模式下 HTTPS 通信仍可进行，但无法实现安全远程访问。因此，用户不能访问安全的远程文件系统和应用程序。

开放门户和安全门户的主要区别在于，由开放门户提供的服务通常驻留在隔离区 (DMZ) 内，而不是驻留在安全的内联网中。DMZ 是公共 Internet 和私有内联网之间的一个小型受保护网络，通常在其两端以防火墙来划界。

如果门户不包含关于部署公共信息和允许访问免费应用程序的含敏感信息，则对大量用户所发出的访问请求的响应速度比使用安全模式更快。

在开放模式下，Portal Server 安装在防火墙后的单台服务器上。多台客户机穿过单台防火墙在 Internet 上访问 Portal Server。

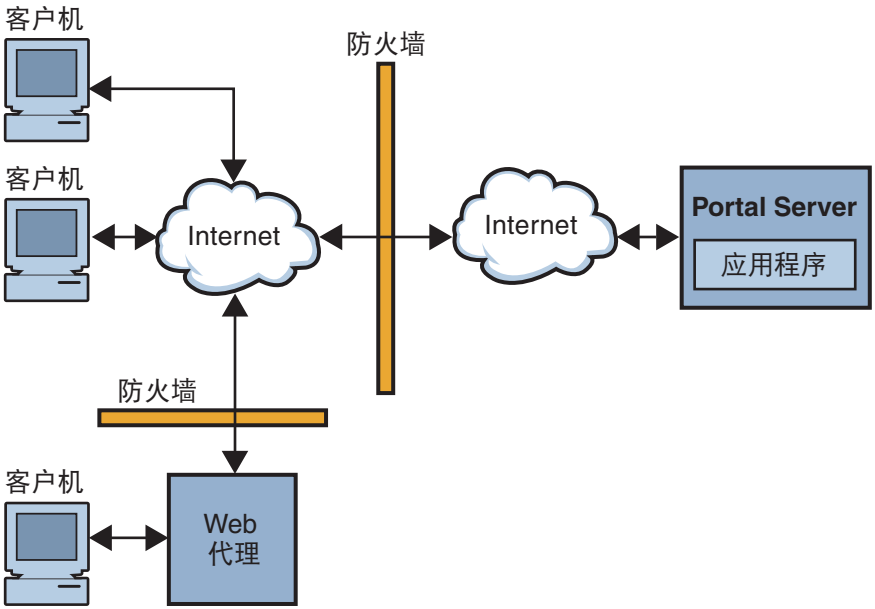


图 1–1 开放模式下的 Portal Server（带有 Secure Remote Access）

安全模式

安全模式可使用户对所需的内联网文件系统和应用程序进行安全远程访问。

网关位于隔离区 (DMZ) 内。网关对所有内联网 URL 和应用程序提供了单个安全访问点，从而减少了防火墙中要打开的端口数。其他所有 Portal Server 服务（如“会话”、

“验证”和标准“Portal 桌面”）均驻留在安全内联网中 DMZ 的后面。从客户机浏览器到网关的通信采用安全套接字层 (Secure Sockets Layer, SSL) 基础之上的 HTTP 进行加密。从网关到服务器和内联网资源的通信既可以是 HTTP，也可以是 HTTPS。

在安全模式下，SSL 用于加密客户机和网关之间的 Internet 连接。SSL 也可用于对网关与服务器之间的连接进行加密。存在于内联网与 Internet 之间的网关使客户机与 Portal Server 之间的安全路径得以延伸。

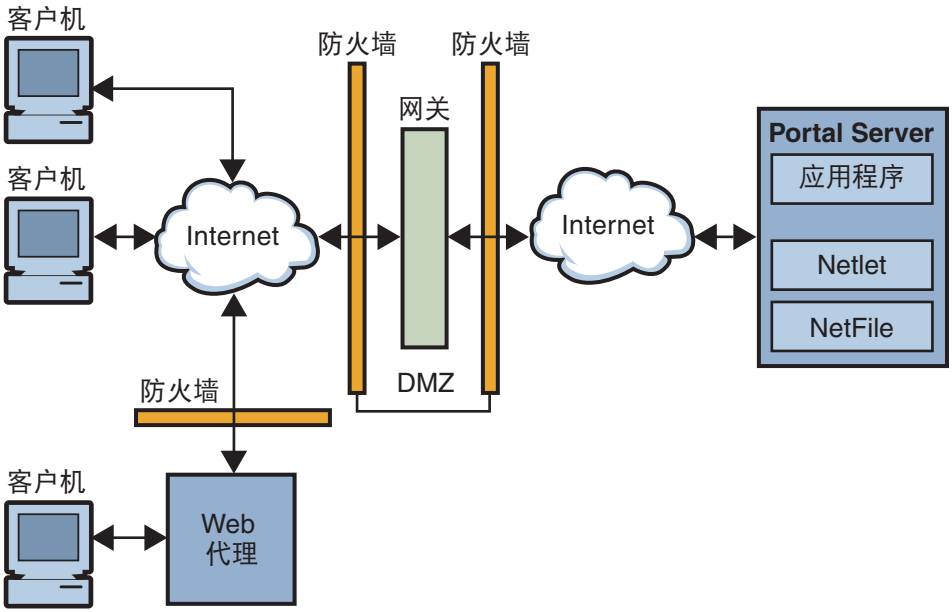


图 1–2 安全模式下的 Portal Server（带有 Secure Remote Access）

可另行添加服务器和网关来扩充站点。Secure Remote Access 软件可根据业务要求以各种方式进行配置。有关如何适应业务要求的详细信息，参见《Sun Java System Portal Server 7.2 Deployment Planning Guide》。

Secure Remote Access 服务

Secure Remote Access 软件有五个主要组件：

- **网关**

  SRA 网关在源自 Internet 的远程用户会话与公司内联网之间提供了接口和安全屏障。网关可通过单一接口将来自内部 Web 服务器和应用程序服务器的内容安全地呈现给远程用户。

Web 服务器使用基于 Web 的资源（例如 HTML、JavaScript 和 XML）在客户机与网关之间进行通信。重写器是一个网关组件，用于使 Web 内容变为可用。

应用服务器使用二进制协议（如 telnet 和 FTP）在客户机与网关之间进行通信。驻留在网关上的 Netlet 便用于该用途。有关详细信息，参见第 2 章。

- **重写器**

  重写器使最终用户可以浏览内联网，并使这些页面上的链接和其他 URL 引用正确发挥作用。重写器预先考虑 Web 浏览器位置字段中的网关 URL，进而通过网关重定向内容请求。有关详细信息，参见第 4 章。

- **Netfile**

  NetFile 是一个文件管理器应用程序，能够远程访问和操作文件系统和目录。NetFile 包括一个基于 Java 的用户界面。有关详细信息，参见第 5 章。

- **Netlet**

  Netlet 有助于安全地在远程桌面上运行常用的或公司特定的应用程序。在您的站点实现 Netlet 后，用户可安全地运行公共 TCP/IP 服务（如 Telnet 和 SMTP）及基于 HTTP 的应用程序（如 pcANYWHERE 或 Lotus Notes）。有关详细信息，参见第 6 章。

- **Proxylet**

  Proxylet 是一个在客户机上运行的动态代理服务器。Proxylet 通过读取和修改客户机上浏览器的代理设置使其指向本地代理服务器或 Proxylet，从而使 URL 重定向到网关。

配置 Secure Remote Access 属性

可在 Portal Server 管理控制台中使用以下服务配置 Secure Remote Access 属性：

- 访问控制

  此服务使您能够允许或限制对特定 URL 的访问，并能对单点登录功能进行管理。有关详细信息，参见第 7 章。

- 网关

  配置文件（网关实例）。此服务使您能够配置所有与网关相关的属性，如启用组件、cookie 管理、代理管理、安全性设置、性能调试、重写器映射管理。有关详细信息，参见第 8 章。

- NetFile

  此服务使您能够配置所有与 NetFile 相关的属性，如公共主机、MIME 类型，以及对不同类型主机的访问。有关详细信息，参见第 14 章。

- Netlet

  此服务使您能够配置所有与 Netlet 相关的属性，如 Netlet 规则、对所需规则的访问、组织和主机，以及默认算法。有关详细信息，参见第 11 章。

- 重写器

  此服务使您能够下载、上载以及删除所有重写器规则集。

- Proxylet

  此服务使您能够配置与 Proxylet 相关的属性，如“Proxylet Applet 绑定 IP”地址及端口号。有关详细信息，参见第 13 章。

注意 – 网关不会收到有关在其运行期间对属性所作更改的通知。重新启动网关以使更新的配置文件属性（属于网关或任何其他服务）生效。有关详细信息，参见第 169 页中的“使用命令行选项配置网关属性”。

设置冲突解决方案

▼ 设置冲突解决方案级别

1. **登录 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后单击所需的服务选项卡：Netlet、Netfile 或 Proxylet。**
3. **从“ 选择 DN ”下拉式菜单中选择“ 组织 ”或“ 角色 ”。**
4. **从“ COS 优先级 ”下拉列表框中选择所需的“ 冲突解决级别 ”。**
5. **单击“ 保存 ”以完成修改。**

支持的应用程序

SRA 支持以下应用程序：

- Sun Java System Calendar Server Release 5.1.1 及更高版本
- Sun Java System Messenger Express 6 2005Q1 - Sun Java System Messaging Server 5.2 及更高版本
- Sun Java System Communications Express 6 2005Q1

开始之前

▼ 为门户启用 SRA

1 通过使用 `PortalServer_base/psadmin switch-sra-status -u amadmin -f <passwordfile> on` 命令转换 SRA 状态。

2 通过使用 `PortalServer_base/psadmin provision-sra -u amadmin -f <passwordfile> -p <portal-id> --gateway-profile <profile-name> --enable` 命令置备 SRA 状态。

◆ ◆ ◆ 第 2 章

使用网关

本章说明与网关有关的概念。有关管理网关的信息，参见第 16 章。有关配置网关的信息，参见第 8 章。

本章包括以下主题：

网关简介

网关在源自 Internet 的远程用户会话与公司内联网之间提供了接口和安全屏障。网关可通过单个接口将来自内联网服务器和应用程序服务器的内容安全地呈现给远程用户。

对于每个网关实例，您必须完成以下任务：

其他与网关有关的主题包括：

- 第 33 页中的 “配置网关监视程序”
- 第 33 页中的 “指定虚拟主机”
- 第 33 页中的 “指定与 Access Manager 联络的代理”

创建网关配置文件

网关配置文件包含与网关配置相关的所有信息，包括网关监听时使用的端口、SSL 选项和代理选项。安装网关时，如果选择默认值，就会创建一个名为 “default” 的默认网关配置文件。与默认配置文件相对应的配置文件位于 `: /etc/opt/SUNWportal/platform.conf.default`。

其中 `/etc/opt/SUNWportal` 是所有 `platform.conf.*` 文件的默认位置。有关 `platform.conf` 文件的详细信息，参见第 34 页中的 “了解 platform.conf 文件” 。

当使用配置文件时，可执行以下任务：

- 创建多个配置文件、定义每个配置文件的属性，并根据需要将这些配置文件分配给不同的网关。
- 将单个配置文件分配给在不同机器上安装的网关。
- 将不同的配置文件分配给在同一机器上运行的单个网关实例。

注意 – 不要将同一配置文件分配给在同一机器上运行的不同网关实例。由于端口号相同，此设置会造成冲突。

不要在为同一网关创建的不同配置文件中指定相同的端口号。以相同端口运行同一网关的多个实例会造成冲突。

创建网关的多个实例

要创建网关的多个实例，参见 《Sun Java System Portal Server 7.2 Installation and Configuration Guide》 中的第 4 章 “Installing and Configuring a Gateway With Portal Server”

创建多宿主网关实例

多宿主网关实例是一个 Portal Server 上的多个网关。要创建这些实例，请按照以下方式修改 `platform.conf` 文件：

```
gatewaybindipaddress = 0.0.0.0
```

使用同一 LDAP 创建网关实例

如果是创建使用同一 LDAP 的多个网关实例，请在创建第一个网关之后，在所有后续网关上执行以下操作：

在 /etc/opt/SUNWam/config/ 中，修改 AMConfig-*instance-name* .properties 中的以下区域，以便与第一个安装的网关实例一致。

参见第 230 页中的 “使用同一 LDAP 创建网关实例”

重新启动网关

通常，您无需重新启动网关。只有在发生以下任一事件时，才需要重新启动网关：

- 已创建一个新的配置文件并且需要将新配置文件分配给网关。
- 已修改现有配置文件中的一些属性并要使更改生效。
- 网关因“内存不足”之类的错误而崩溃。
- 网关停止响应，且不对任何请求提供服务。

配置网关监视程序

可以配置监视程序监控网关状态的时间间隔。要启动或停止监视程序，请运行命令 ./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off。该时间间隔默认设置为 60 秒。要更改该值，请在 crontab 实用程序中编辑下面一行：

```
0-59 * * * * gateway-install-root/SUNWportal/bin/
/var/opt/SUNWportal/.gw. 5 > /dev/null 2>&1
```

参见 crontab 的手册页以便配置 crontab 条目。

指定虚拟主机

虚拟主机是指向同一机器 IP 和主机名的附加主机名。例如，如果某一主机名 abc 指向主机 IP 地址 192.155.205.133，则您可以添加指向同一 IP 地址的另一个主机名 cde。

指定与 Access Manager 联络的代理

您可以指定一个代理主机，网关将使用它来联络在 Portal Server 之上部署的“SRA 核心”(RemoteConfigServlet)。网关使用此代理可访问到 Portal Server 和 Access Manager。参见第 233 页中的 “指定代理”。

了解 platform.conf 文件

默认情况下，platform.conf 文件位于： /etc/opt/SUNWportal。

platform.conf 文件包含网关所需要的详细信息。本节提供了一个范例 platform.conf 文件，并且说明了所有条目。

在配置文件中包含所有机器特定细节的优势在于：一个通用的配置文件可以被多个机器上运行的各个网关共享。

以下是 platform.conf 文件的样例。

```
Tue May 30 11:51:23 IST 2006
debug.com.sun.portal.rewriter.original.level=INFO
gateway.favicon=
gateway.bindipaddress=10.12.154.236
debug.com.sun.portal.sra.rproxy.toFromServer.handler.java.util.logging.FileHandler.pattern=
/var/opt/SUNWportal/logs/sra/default/Gateway.toFromServer.%u.%g.log
gateway.port=443
rewriterproxy.jvm.flags=-ms64m -mx128m
portal.server.instance=default
debug.com.sun.portal.handler.java.util.logging.FileHandler.filter=
gateway.jdk.dir=/usr/jdk/entsys-j2se
gateway.ignoreURIList=/MSOffice/cltreq.asp,/_vti_bin/owssvr.dll
debug.com.sun.portal.rewriter.rest.level=INFO
gateway.trust_all_server_certs=true
debug.com.sun.portal.handler.java.util.logging.FileHandler.append=true
gateway.cdm.cacheCleanupTime=300000
gateway.httpurl=
debug.com.sun.portal.handler.java.util.logging.FileHandler.count=1
gateway.jvm.classpath=
debug.com.sun.portal.setserverlogs=false
gateway.protocol=https
debug.com.sun.portal.sra.rproxy.toFromServer=java.util.logging.FileHandler
rewriterproxy.jvm.classpath=
gateway.enable.customurl=false
debug.com.sun.portal.sra.rproxy.toFromBrowser=java.util.logging.FileHandler
debug.com.sun.portal.handler.java.util.logging.FileHandler.formatter=com.sun.portal.
log.common.PortalLogFormatter
debug.com.sun.portal.sra.rproxy.toFromBrowser.handler.java.util.logging.FileHandler.pattern=
/var/opt/SUNWportal/logs/sra/default/Gateway.toFromBrowser.%u.%g.log
debug.com.sun.portal.level=INFO
debug.com.sun.portal.rewriter.unaffected.separatefile=true
gateway.enable.accelerator=false
debug.com.sun.portal.rewriter.original.separatefile=true
gateway.virtualhost=nicp236.india.sun.com 10.12.154.236
debug.com.sun.portal.stacktrace=true
gateway.host=nicp236.india.sun.com
```

```
debug.com.sun.portal.handler.java.util.logging.FileHandler.pattern=
/var/opt/SUNWportal/logs/sra/default/%logger.%sraComponentType.%u.%g.log
gateway.certdir=/etc/opt/SUNWportal/cert/default
gateway.sockretries=3
gateway.allow.client.caching=true
debug.com.sun.portal.rewriter.unaffected.level=INFO
debug.com.sun.portal.rewriter.uriinfo.separatefile=true
log.config.check.period=2000
debug.com.sun.portal.rewriter.rewritten.level=INFO
gateway.userProfile.cacheSize=1024
debug.com.sun.portal.rewriter.rulesetinfo.level=INFO
netletproxy.jvm.classpath=
gateway.userProfile.cacheSleepTime=60000
debug.com.sun.portal.rewriter.uriinfo.level=INFO
debug.com.sun.portal.rewriter.rest.separatefile=true
gateway.notification.url=notification
debug.com.sun.portal.rewriter.rulesetinfo.separatefile=true
gateway.logdelimiter=&&
gateway.ignoreServerList=false
gateway.jvm.flags=-ms64m -mx128m
debug.com.sun.portal.handler.java.util.logging.FileHandler.limit=5000000
gateway.dsame.agent=http\://sunone216.india.sun.com\:8080/portal/RemoteConfigServlet
gateway.httpsurl=
gateway.retries=6
gateway.userProfile.cacheCleanupTime=300000
gateway.logging.password=X03MO1qnZdYdgyfeuILPmQ\=\= UX9x0jIua3hx1YOVRG/TLg\=\=
netletproxy.jvm.flags=-ms64m -mx128m
debug.com.sun.portal.rewriter.rewritten.separatefile=true
gateway.user=noaccess
gateway.external.ip=10.12.154.236
debug.com.sun.portal.handler=java.util.logging.FileHandler
gateway.cdm.cacheSleepTime=60000
rewriterproxy.accept.from.gateways=
rewriterproxy.checkacl=false
```

下表列出并介绍了 `platform.conf` 文件中的所有字段。

表 2–1 文件属性

表项	默认值	描述
`gateway.user`	noaccess	以该用户身份运行网关。 必须以超级用户身份启动网关，在初始化之后将丧失超级用户权限而成为该用户。
`gateway.jdk.dir`		这是网关使用的 JDK 目录的位置。

表 2–1 文件属性 （续）

表项	默认值	描述
gateway.dsame.agent		这是网关启动时为获取其配置文件所联络的 Access Manager 的 URL。
portal.server.protocol portal.server.host portal.server.port		这是 Portal Server 默认安装正在使用的协议、主机和端口。
gateway.protocolgateway.hostgateway.port		这是网关的协议、主机和端口。这些值与您在安装期间所指定的模式和端口相同。这些值用于构造通知用的 URL。
gateway.trust_all_server_certs	true	该项指示网关是否必须信任所有服务器证书，或者仅信任那些位于网关证书数据库中的证书。
gateway.trust_all_server_cert_domains	false	当网关与服务器之间进行 SSL 通信时，会将服务器证书提交给网关。默认情况下，网关检查服务器主机名是否与服务器证书 CN 相同。 如果该属性值被设为 true，则网关将禁止对所接收的服务器证书进行域检查。
gateway.virtualhost		如果网关机器具有多个已配置的主机名，则可以在此字段中指定一个不同的名称和身份认证提供者地址。
gateway.virtualhost.defaultOrg=org		该项指定用户登录到的默认 Org。 例如，假设虚拟主机字段条目如下所示： gateway.virtualhost=test.com employee.test.com Managers.test.com 而默认 Org 条目为： test.com.defaultOrg = o=root,dc=test,dc=com employee.test.com.defaultOrg = o=employee,dc=test,dc=com Manager.test.com.defaultOrg = o=Manager,dc=test,dc=com 用户可以使用 https://manager.test.com，而不是 https://test.com/o=Manager,dc=test,dc=com 登录到管理人员的 Org。 **注 –** virtualhost 和 defaultOrg 在 platform.conf **文件**中区分大小写，但在 URL 中使用时却不区分。

表 2–1 文件属性 （续）

表项	默认值	描述
gateway.notification.url		网关主机、协议和端口的组合用于构造通知用的 URL。它用于从 Access Manager 接收会话通知。 请确保通知用的 URL 与任何组织名称都不相同。如果通知 URL 与某一组织名匹配，则尝试连接至该组织的用户得到的是空白页，而不是登录页。
gateway.retries		这是启动时网关试图联络 Portal Server 的次数。
gateway.debug	error	该项可设置网关的调试级别。调试日志文件位于 *debug-directory*/files。调试文件的位置在 gateway.debug.dir 条目中指定。 调试级别为： ■ error - 只有严重的错误才会记录到调试文件中。当出现此类错误时，网关通常停止运行。 ■ warning - 记录警告消息。 ■ message - 记录所有调试消息。 ■ on - 在控制台上显示所有调试信息。 调试文件为： srapGateway.*gateway-profile-name* - 包含网关调试消息。 Gateway_to_from_server.*gateway-profile-name* - 在消息模式下，该文件包含网关与内部服务器之间的所有请求和响应标题。 要生成该文件，请更改 /var/opt/SUNWportal/debug 目录的写权限。 Gateway_to_from_browser.*gateway-profile-name* - 在消息模式下，该文件包含网关和客户机浏览器间的所有请求和响应标题。 要生成该文件，请更改 /var/opt/SUNWportal/debug 目录的写权限。
gateway.debug.dir		这是生成所有调试文件的目录。 此目录必须具有足有的权限以将 gateway.user 中所提及的用户写入文件。
gateway.logdelimiter		当前未使用。
gateway.external.ip		对于多宿主网关机器（一部机器具有多个 IP 地址），需要在此指定外部 IP 地址。Netlet 使用该 IP 用于运行 FTP。
gateway.certdir		该项指定证书数据库的位置。

表 2–1 文件属性 （续）

表项	默认值	描述
gateway.allow.client.caching	true	允许或禁止客户机高速缓存。 如果允许，则客户机浏览器就可以为实现更好的性能而高速缓存静态页和图像（通过已缩减的网络通信量）。 如果不允许，则不会缓存任何内容且安全性更高，但性能会随着网络负载增高而下降。
gateway.userProfile.cacheSize		这是在网关获取高速缓存的用户配置文件条目数。如果条目数超过了该值，则会出现频繁重试以便清除高速缓存。
gateway.userProfile.cacheSleepTime		设置清除高速缓存的睡眠时间（以秒为单位）。
gateway.userProfile.cacheCleanupTime		以秒为单位的最大时间，超过该时间后就可以删除一个配置文件条目。
gateway.bindipaddress		在多宿主机器上，这是 IP 地址，网关将其 serversocket 绑定到该地址。要配置网关以侦听所有接口，请替换该 IP 地址以使 gateway.bindipaddress=0.0.0.0。
gateway.sockretries	3	当前未使用。
gateway.enable.accelerator	false	如果设置为 true，则允许支持外部加速器。
gateway.enable.customurl	false	如果设置为 true，则允许管理员为网关指定一个自定义的 URL 以便将页重写至该 URL。
gateway.httpurl		与网关将页面重写到的自定义 URL 相应的 HTTP 反向代理 URL。Proxylet 启用时使用此条目。
gateway.httpsurl		与网关将页面重写到的自定义 URL 相应的 HTTPS 反向代理 URL。如果 Proxylet 已启用，则不使用此条目。
gateway.favicon		网关将 favicon.icon 文件请求重定向到的 URL。 它用于 Internet Explore 和 Netscape 7.0 及更高版本中的 “收藏图标” 。 如果留为空白，网关会向浏览器发回一条 404 未找到消息。
gateway.logging.password		用户 amService-srapGateway 的 LDAP 密码，网关使用它创建其应用程序会话。 该字段既可以是加密文本，也可以是明文。
http.proxyHost		此代理主机用于联络 Portal Server。
http.proxyPort		这是主机用于联络 Portal Server 的端口。
http.proxySet		如果需要代理主机，将此属性设置为 true。如果将该属性设置为 false，则忽略 http.proxyHost 和 http.proxyPort 。

表 2–1 文件属性 （续）

表项	默认值	描述
portal.server.*instance*		该属性的值是相应的 /etc/opt/SUNWam/config/AMConfig-*instance-name*.properties 文件。如果该值为默认值，则其指向 AMConfig.properties。
gateway.cdm.cacheSleepTime	60000	从 Access Manager 发送至网关的高速缓存“客户机检测模块”响应的超时值。
gateway.cdm.cacheCleanupTime	300000	从 Access Manager 发送至网关的高速缓存“客户机检测模块”响应的超时值。
netletproxy.port	10555	Netlet 代理守护进程在此端口侦听请求。
rewriterproxy.port	10555	“重写器代理”守护进程在此端口侦听请求。
gateway.ignoreServerList	false	如果设置为 true，则使用 AMConfig.properties 文件中指定的值来构造 Access Manager 服务器 URL。当 Access Manager 服务器躲在负载平衡器后面时，将此属性设置为 true。
rewriterproxy.accept.from.gateways		此属性是指 IP 地址列表，可使重写器代理接受来自该列表的请求。它在 HTTP 和 HTTPS 模式下均可运行。它用于提高安全性，只接受来自该集合的请求，而不处理所有其他请求。IP 地址可使用逗号分隔。默认值为空，被视为传统模式，即接受所有送至重写器代理的请求。
rewriterproxy.checkacl=	false	启用该属性，可使重写器代理如同网关一样检查 ACL 值。传统模式值为“false”。当设置为 true 时，重写器代理将在指定的 DN 上根据网关访问服务中指定的值来检查 URL，并将根据位于该处的列表集合允许/拒绝请求。该值在 HTTP 和 HTTPS 模式下均非常有用。

使用 Web 代理

您可以使用第三方 Web 代理 配置网关以联系 HTTP 资源。Web 代理位于客户机和 Internet 之间。

Web 代理配置

不同的代理可能用于不同的域和子域。这些条目告诉网关使用哪个代理去联络特定域中的特定子域。在网关中指定的代理配置具有如下功能：

- 创建域和子域列表，同时在网关服务中的“域和子域代理”字段中创建所需的代理。
- 启用“使用代理”选项：
  - 在“域和子域代理”字段中指定的代理被用于指定的主机。

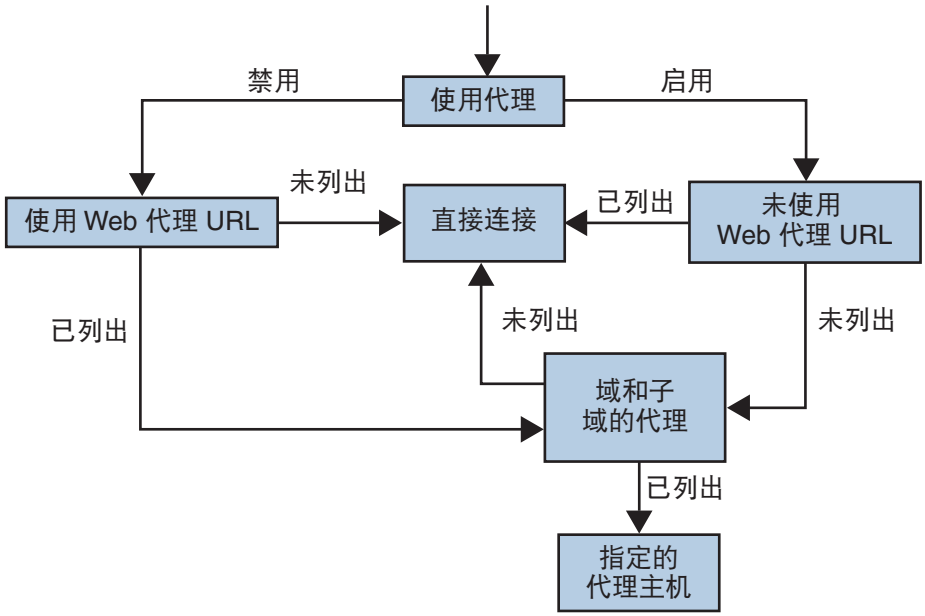
- 要实现直接连接“域和子域代理”列表中指定的域和子域内的特定 URL，请在“不使用 Web 代理 URL”字段中指定这些 URL。

禁用“使用代理”选项：

- 要确保代理被用于在“域和子域代理”字段中指定的域和子域内的特定 URL，请在“使用 WebProxy URL”列表中指定这些 URL。

  虽然已禁用“使用代理”选项，但是代理仍用于连接在“使用 Webproxy URL”下列出的 URL。这些 URL 的代理从“域和子域代理”列表中获得。

  以下图示说明显示了如何根据网关服务中的代理配置来解析 Web 代理信息。



在第 39 页中的 “Web 代理配置”中，如果启用“使用代理”，并且请求的 URL 在“不可使用 Web 代理 URL”列表中列出，则网关直接连接到目标主机。

如果启用“使用代理”，并且请求的 URL 未在“不可使用 Webproxy URL”列表中列出，则网关通过指定的代理连接到目标主机。如果已经指定代理，则可以在“域和子域代理”列表中查寻。

如果禁用“使用代理”，并且请求的 URL 在“使用 Webproxy URL”列表中列出，则网关使用在“域和子域代理”列表中的代理信息连接到目标主机。

如果禁用“使用代理”，并且请求的 URL 未在“使用 Webproxy URL”列表中列出，则网关直接连接到目标主机。

如果上述条件都无法满足，则不可能进行直接连接，网关显示一条错误信息，说明不可能进行连接。

注 – 如果您正通过标准“Portal 桌面”的“书签”频道访问 URL，但未满足上述任一条件，则网关将向浏览器发送一条重定向指令。浏览器使用它自己的代理设置访问 URL。

语法

```
domainname [web_proxy1:port1]|subdomain1 [web_proxy2:port2]|
```

示例

```
sesta.com wp1:8080|red wp2:8080|yellow|* wp3:8080
```

* 通配符可匹配一切内容

其中：

`sesta.com` 是域名而 `wp1` 是在端口 8080 上用于联络的代理。

`red` 是子域而 `wp2` 是在端口 8080 上用于联络的代理。

`yellow` 是子域。由于没有指定代理，因此使用为域指定的代理，即端口 8080 上的 `wp1`。

* 指示其他所有子域都需要在端口 8080 上使用 `wp3`。

注 – 如果未指定端口，则默认使用 8080 端口。

处理网络代理信息

当客户机尝试访问特定 URL 时，URL 中的主机名会与“域和子域的代理”列表中的条目匹配。要考虑与所请求主机名的最长后缀相匹配的条目。例如，假设所请求的主机名为 `host1.sesta.com`。会按顺序进行以下搜索，直到找到匹配条目为止。

- 扫描 `host1.sesta.com` 中的域和子域的代理。如果发现匹配的条目，则根据该条目所指定的代理将用于连接该主机。
- 或者，在列表中扫描 `*.sesta.com`。如果找到匹配条目，则使用相应的代理。
- 或者，在列表中搜索 `sesta.com`。如果找到匹配条目，则使用相应的代理。
- 或者，在列表中搜索 `*.com`。如果找到匹配条目，则使用相应的代理。
- 或者，在列表中搜索 `com`。如果找到匹配条目，则使用相应的代理。
- 或者，在列表中搜索 `*`。如果找到匹配条目，则使用相应的代理。
- 如果没有找到匹配条目，则尝试直接连接。

考虑以下“域和子域代理”列表中的条目：

```
com p1| host1 p2 | host2 | * p3
sesta.com p4 | host5 p5 | * p6
florizon.com | host6
abc.sesta.com p8 | host7 p7 | host8 p8 | * p9
host6.florizon.com p10
host9.sesta.com p11
siroe.com | host12 p12 | host13 p13 | host14 | * p14
siroe.com | host15 p15 | host16 | * p16
* p17
```

网关会按照以下的表格所示，将这些条目内部映射到表格。

表 2–2 域和子域代理列表中条目的映射

数量	域和子域代理列表中的条目	代理	描述
1	com	p1	如列表中指定。
2	host1.com	p2	如列表中指定。
3	host2.com	p1	由于没有为 host2 指定任何代理，因而使用域的代理。
4	*.com	p3	如列表中指定。
5	sesta.com	p4	如列表中指定。
6	host5.sesta.com	p5	如列表中指定。
7	*.sesta.com	p6	如列表中指定。
8	florizon.com	直接	有关详细信息，参见条目 14 的说明。
9	host6.florizon.com	–	有关详细信息，参见条目 14 的说明。
10	abc.sesta.com	p8	如列表中指定。
11	host7.abc.sesta.com	p7	如列表中指定。
12	host8.abc.sesta.com	p8	如列表中指定。
13	*.abc.sesta.com	p9	如列表中指定。在 `abc.sesta.com` 域下，所有主机（`host7` 和 `host8` 除外）都会使用 `p9` 作为代理。
14	host6.florizon.com	p10	此条目与条目 9 相同。条目 9 指示直接连接，但是此条目指示应当使用代理 p10。倘若有两个这样的条目，含有代理信息的条目被视为有效条目。另一个条目将被忽略。
15	host9.sesta.com	p11	如列表中指定。

表 2–2 域和子域代理列表中条目的映射 （续）

数量	域和子域代理列表中的条目	代理	描述
16	siroe.com	直接	由于没有为 siroe.com 指定任何代理，所以尝试进行直接连接。
17	host12.siroe.com	p12	如列表中指定。
18	host13.siroe.com	p13	如列表中指定。
19	host14.siroe.com	直接	由于没有为 host14 指定任何代理，所以尝试进行直接连接。
20	*.siroe.com	p14	参见表项 23 的说明。
21	host15.siroe.com	p15	如列表中指定。
22	host16.siroe.com	直接	由于没有为 host16 或 siroe.com 指定任何代理，所以尝试进行直接连接。
23	*.siroe.com	p16	类似于条目 20，但是所指定的代理却不同。在此情况下，无法知道网关的确切行为。可能使用两个代理中的任意一个。
24	*	p17	如果没有其他条目与所请求的 URL 匹配，则使用 p17 作为代理。

提示 – 您可以将“域和子域代理”列表中的各个条目放在列表中单独的行上，而不是使用 | 符号分隔代理条目。例如，不使用一个条目：

```
sesta.com p1 | red p2 | * p3
```

您可以按以下方式指定此信息：

```
sesta.com p1
red.sesta.com p2
*.sesta.com p3
```

这种列表格式更容易跟踪重复的条目或任何其他多义的情况。

基于“域和子域代理”列表进行重写

重写器也使用“域和子域代理”列表中的条目。重写器重写所有 URL（它们的域与“域和子域代理”列表中列出的域相匹配）。

注意 – 不会考虑重写“域和子域的代理”列表中的 * 条目。例如，不会考虑条目 24。

有关重写器的信息，参见第 4 章。

默认域和子域

当 URL 中的目标主机不是全限定主机名时，默认的域和子域将用于到达全限定名。

假设管理控制台的“默认域”字段中的值是：

```
red.sesta.com
```

注 – 需要在“域和子域代理”列表中具有相应的条目。

在上面的示例中，`sesta.com` 是默认域而默认子域是 `red`。

如果请求的 URL 是 `host1`，则使用默认的域和子域将此条目解析为 `host1.red.sesta.com`。然后，在“域和子域代理”列表中查找 `host1.red.sesta.com`。

使用自动代理配置

要忽略“域和子域代理”列表中的信息，请启用“自动代理配置”功能。

当使用代理自动配置 (Proxy Auto Configuration, PAC) 文件时：

- Portal Server、网关、Netlet 和 Proxylet 使用 *Rhino* 软件解析 PAC 文件。您可以从 Java Enterprise System Accessory CD 安装 `SUNWrhino` 软件包。

  此软件包包含 `/usr/share/lib` **目录**中必须存在的 `js.jar` 文件。将此目录添加到网关和 Portal Server 机器上的 `webserver/appserver` 类路径，否则 Portal Server、网关、Netlet 和 Proxylet 无法解析 PAC 文件。
- `js.jar` 必须存在于网关机器上的 `$JRE_HOME/lib/ext` 目录中，否则网关无法解析 PAC 文件。
- 启动时，网关从网关配置文件“自动代理配置文件”位置字段中所指定的位置获取 PAC 文件。
- 网关使用 URLConnection API 到达该位置。如果需要配置代理才能访问到该网关，需按以下方式配置代理：

  1. 在命令行中编辑以下文件：

     `/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*
  2. 添加下列条目：

     `http.proxyHost=`*web-proxy-hostname*

     `http.proxyPort=`*web-proxy-port*

     `http.proxySet=true`

3. 重新启动网关以使用指定的代理：

   ```
   ./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>–t <gateway>
   ```

- 如果 PAC 文件初始化失败，则网关使用“域和子域代理”列表中的信息。
- 如果从 PAC 文件返回 ""（空字符串）或“null”，则网关假设该主机不属于内联网。这类似于主机不在“域和子域代理”列表中。

  如果要网关使用直接到主机的连接，将返回“DIRECT”。参见第 45 页中的 “返回 DIRECT 或 NULL 的示例”。
- 当指定多个代理时，网关仅使用返回的第一个代理。它不会尝试在为主机指定的各个代理间进行故障转移或负载平衡。
- 网关忽略 SOCKS 代理并且尝试直接连接，并假设主机是内联网的一部分。
- 要指定用于到达任何不属于内联网的主机的代理，请使用代理类型 STARPROXY。此代理类型是 PAC 文件格式的扩展，类似于网关配置文件的“域和子域代理”部分中的 `* proxyHost:port` 条目。参见第 46 页中的 “返回 STARPROXY 的示例”

PAC 文件用法示例

以下示例显示在“域和子域代理”列表中列出的 URL 及相应的 PAC 文件。

返回 DIRECT 或 NULL 的示例

如果将这些代理用于域和子域：

```
*intranet1.com proxy.intranet.com:8080

intranet2.com proxy.intranet1.com:8080
```

相应的 PAC 文件是：

```
// Start of the PAC File
function FindProxyForURL(url, host) {
         if (dnsDomainIs(host, ".intranet1.com")) {
             return "DIRECT";
         }
          if (dnsDomainIs(host, ".intranet2.com")) {
              return "PROXY proxy.intranet1.com:8080";
          }
          return "NULL";
}
//End of the PAC File
```

返回 STARPROXY 的示例

如果将这些代理用于域和子域：

```
intranet1.com

intranet2.com.proxy.intranet1.com:8080

internetproxy.intranet1.com:80
```

相应的 PAC 文件是：

```
// Start of the PAC File
function FindProxyForURL(url, host) {
          if (dnsDomainIs(host, ".intranet1.com")) {
              return "DIRECT";
          }
          if (dnsDomainIs(host, ".intranet2.com")) {
              return "PROXY proxy.intranet1.com:8080;" +
                  "PROXY proxy1.intranet1.com:8080";
          }
          return "STARPROXY internetproxy.intranet1.com:80";
}
//End of the PAC File
```

在此情况下，如果请求 `.intranet2.com` **域**中的主机，网关会联络 `proxy.intranet1.com:8080`。如果 proxy.intranet1.com:8080 停用，则请求失败。网关不进行故障转移而联络 `proxy1.intranet1.com:8080`。

指定 PAC 文件的位置

指定 PAC 文件位置的格式取决于其所在的位置，如下所示：

- 如果 PAC 文件驻留于 Web 服务器上，则 PAC URL 为：

  `http://`hostname`/`*pacfile_name*`.pac`

- 如果 PAC 文件是 Java 1.4.1_x 的本地文件（例如 `c:\\pacfile\\sample.pac`），则输入如下格式的 PAC URL：

  `file://c:/pacfile/sample.pac`

- 如果 PAC 文件是 Java 1.4.2_x 的本地文件（例如 `c:\\pacfile\\sample.pac`），则输入如下格式的 PAC URL：

  `file:///c:/pacfile/sample.pac`

在单独的会话中添加服务

当您在单独的会话中添加 Portal Server 服务时：

- 在 Portal Server 管理控制台的“网关”>“核心”下列出 Portal Server。
- 在“网关”>“安全”下的“免验证 URL”中会列出 Portal Server URL。

使用 Netlet 代理

Netlet 信息包在网关处解码并被发送至目标服务器。然而，网关需要通过隔离区 (DMZ) 和内联网之间的防火墙，才能访问所有的 Netlet 目标主机。此设置需要在防火墙中打开许多端口。Netlet 代理可于将防火墙中打开的端口数量降至最低。

Netlet 代理通过延展客户机至网关最终至内联网中的 Netlet 代理之间的安全隧道，从而增强了网关和内联网之间的安全性。通过使用代理，Netlet 信息包将被代理解码然后发送到目标服务器地。

使用 Netlet 代理的优点如下：

- 添加了额外的安全层。
- 在大规模的部署环境中，尽量减少了网关通过内部防火墙使用额外 IP 地址和端口的情况。
- 将网关和 Portal Server 之间的打开端口数限制为 1。该端口数可以在安装期间配置。
- 扩展了客户机与网关之间的安全通道，最多可扩展至 Portal Server，如第 47 页中的“使用 Netlet 代理”的“使用配置的 Netlet 代理”一节中所示。通过数据加密 Netlet 代理提供的安全性得到改善，但是可能会增加系统资源的使用量。有关安装 Netlet 代理的信息，参见《Sun Java System 安装指南》。

您可以执行以下任务：

- 在 Portal Server 节点或在单独的节点上安装 Netlet 代理。
- 使用管理控制台安装多个 Netlet 代理并且为单个网关配置这些代理。这有利于负载平衡。
- 在单个机器上配置 Netlet 代理的多个实例。
- 将多个网关实例指向单个安装的 Netlet 代理。
- 通过 Web 代理开通 Netlet 通道。

显示在安装和未安装 Netlet 代理的情况下，网关和 Portal Server 的三个实现示例。组件包括一台客户机、两个防火墙、驻留在两个防火墙之间的网关、Portal Server 和 Netlet 目标服务器。

第一种方案显示网关和未安装 Netlet 代理的 Portal Server。数据加密仅从客户机扩展至网关。对于每个 Netlet 连接请求，都会在第二个防火墙中打开一个端口。

在第二种方案显示的网关和 Portal Server 中，Portal Server 上安装了 Netlet 代理。数据加密从客户机一直扩展到 Portal Server。由于所有 Netlet 连接都通过 Netlet 代理路由，因此在第二个防火墙中仅需为 Netlet 请求打开一个端口。

第三种方案显示网关以及在单独节点上安装 Netlet 代理的 Portal Server。在单独节点上安装 Netlet 代理将减少 Portal Server 节点上的负载。此外，仅需在第二个防火墙中打开两个端口。一个端口将请求送至 Portal Server，另一个端口将 Netlet 请求发送至 Netlet 代理服务器。

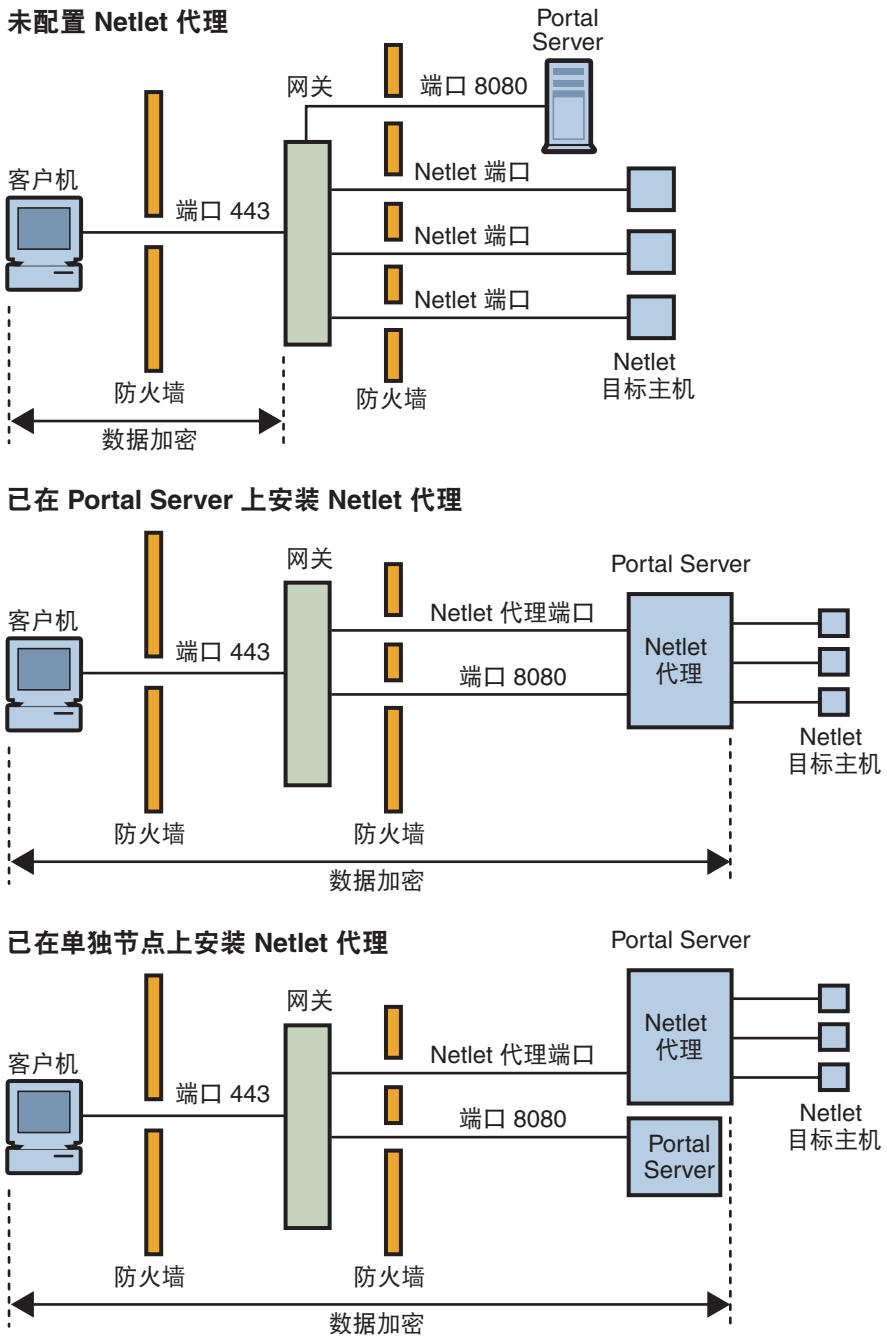


图 2–1 Netlet 代理的实现

启用 Netlet 代理

通过使用 Portal Server 管理控制台中的网关服务可启用 Netlet 代理。

重新启动 Netlet 代理

可将 Netlet 代理配置为只要代理被意外终止就重新启动。可以计划监视程序进程时间表来监控 Netlet 代理，只要它停止运行就重新启动。

也可以手动重新启动 Netlet 代理。有关步骤，参见第 234 页中的 “重新启动 Netlet 代理”。

配置 Netlet 代理监视器

可以配置监视程序监控 Netlet 代理状态的时间间隔。该时间间隔默认设置为 60 秒。要更改此间隔，请将以下一行添加至 `crontab` 文件：

0-59 * * * * *netlet-install-dir*`/bin/checkgw /var/opt/SUNWportal/.gw 5 > /dev/null 2>&1`

注 – 要启动或停止监视程序，请运行命令 `./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off`。

使用重写器代理

重写器代理安装于内联网中。网关不会直接尝试检索内容，而是将所有请求转发至重写器代理，由重写器代理获取内容并将其返回至网关。

使用重写器代理的优点如下：

- 如果网关与服务器之间有防火墙，防火墙只需打开两个端口--一个为网关与重写器代理之间的端口，另一个为网关与 Portal Server 之间的端口。
- HTTP 通信在网关和内联网之间是安全的，即使目标服务器仅支持 HTTP 协议（不支持 HTTPS）。

如果未指定重写器代理，那么当用户尝试访问内联网计算机时，网关组件将会直接连接到内联网计算机。

如果您将重写器代理用作负载平衡器，请确保重写器的 `platform.conf.instance_name` 指向负载平衡器的 URL。此外，请在 Portal Server 列表中指定负载平衡器主机。

如果每个网关实例都有多个重写器代理实例（不必位于门户节点上），请在 `platform.conf` 文件中以 *host-name:port* 形式提供每个重写器代理的详细信息，不是重写器代理的单个端口条目。

创建重写器代理的实例

使用 `rwpmultiinstance` 脚本在 Portal Server 节点上创建新的重写器代理实例。创建网关配置文件后运行此脚本。

参见第 234 页中的 “创建重写器代理实例” 。

启用重写器代理

在 Access Manager 管理控制台中，通过“SRA 配置”下的网关服务启用重写器代理。

重新启动重写器代理

可以将重写器代理配置为只要代理被意外终止就重新启动。可以制定监视程序进程计划来进行监控，只要出现此种情况就重新启动。

也可以手动重新启动重写器代理。

参见第 235 页中的 “重新启动重写器代理” 。

配置重写器代理监视程序

可以配置监视程序监控重写器代理状态的时间间隔。该时间间隔默认设置为 60 秒。要更改时间间隔，请将以下一行添加至 `crontab` 文件中：

0-59 * * * * *rewriter-proxy-install-root* `/bin/checkgw /var/opt/SUNWportal/.gw 5 > /dev/null 2>&1`

注 – 要启动或停止监视程序，请运行命令 `./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off`。

与网关一起使用反向代理

代理服务器将 Internet 内容传送至内联网，而反向代理将内联网内容传送至 Internet。您可以配置反向代理的部署，以实现负载平衡和高速缓存。

如果部署中网关前面有第三方反向代理，必须用反向代理的 URL 重写响应内容，而不是网关的 URL。对此，需要进行下列配置。

参见第 235 页中的 “启用反向代理” 。

获取客户机信息

当网关将客户机请求转发至任何内部服务器时，它将 HTTP 标题添加到 HTTP 请求中。可以使用这些报头来获取额外的客户机信息并检测网关是否存在。

要查看 HTTP 请求报头，请将 `platform.conf` 文件中的条目设置为 `gateway.error=message`。然后，使用 servlet API 的 `request.getHeader()`。下表列出了 HTTP 报头中的信息。

表 2–3 HTTP 报头中的信息

报头	语法	描述
PS-GW-PDC	`X-PS-GW- PDC: true/false`	指示是否已在网关启用 PDC。
PS-Netlet	`X-PS-Netlet:enabled=true/false`	指示网关是否已启用或禁用 Netlet。 如果 Netlet 已启用，则会填充加密选项，指示网关是以 HTTPS (`encryption=ssl`) 模式还是以 HTTP 模式 (`encryption=plain`) 运行。 例如： ■ `PS-Netlet: enabled=false` Netlet 已禁用。 ■ `PS-Netlet: enabled=true; encryption=ssl` Netlet 通过以 SSL 模式运行的网关启用。 当未启用 Netlet 时，`encryption=ssl` 或 `encryption=plain` 不会被填充。
PS-GW-URL	`X-PS-GW-URL: http(s)://gatewayURL(:port)`	指示客户机连接的 URL。 如果端口为非标准端口，例如，如果网关是 HTTP/HTTPS 模式而端口并非 80/443，则也会填充 `:port`。

表 2–3 HTTP 报头中的信息 （续）

报头	语法	描述
PS-GW-Rewriting-URL	`X-PS-GW-URL: http(s)://gatewayURL(:port)/[SessionInfo]`	指示网关将全部页重写至的 URL。 1. 当浏览器支持 cookie 时，此报头的值与 PS-GW-URL 报头相同。 2. 当浏览器不支持 cookie 时： ■ 如果目标主机列于“用户会话 Cookie 被转发到的用户会话”字段中，该值为网关将页面重写到的实际 URL（包括已编码的 SessionID 信息）。 ■ 如果目标主机未列于“用户会话 Cookie 被转发到的用户会话”字段中，则 `SessionInfo` 字符串为 `$SessionID`。 **注 –** 作为响应的一部分，如果用户的 Access Manager sessionId 发生变化（就像来自验证页的响应那样），则使用该值（而不是先前在报头中指示的值）重写页面。 例如： ■ 如果浏览器支持 cookie： PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/ ■ 如果浏览器不支持 cookie 且端服务器列于“用户会话 Cookie 被转发到的用户会话”字段中。 PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/SessIDValCustomEncodedValue/ ■ 如果浏览器不支持 cookie 且端服务器未列于“用户会话 Cookie 被转发到的用户会话”字段中。 PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/$SessionID
PS-GW-CLientIP	`X-PS-GW-CLientIP:` *IP*	表示网关从 `recievedSocket.getInetAddress().getHostAddress()` 获取的 IP。 如果直接连接至网关，则该值可提供客户机的 IP。

使用验证链

验证链提供了比常规验证机制更高级别的安全性。可用多个验证机制验证用户。

这里所描述的过程仅用于在网关同时启用验证链和个人数字证书 (Personal Digital Certificate, PDC) 验证。有关网关上无 PDC 验证的验证链信息，参见 Access Manager 管理指南。

例如，如果将 PDC 和 Radius 验证模块链在一起，用户须得通过全部三个模块的验证才能访问标准“Portal 桌面”。

有关步骤，参见第 236 页中的 “向现有 PDC 实例添加验证模块”。

注 – 启用后，PDC 总是第一个显示给用户的验证模块。

使用通配符证书

通配符证书接受具有主机的全限定 DNS 名中通配符的单个证书。

通过使用证书，可以保护同一域内的多个主机。例如，`*.domain.com` 的证书可用于 `abc.domain.com` 和 `abc1.domain.com`。此证书对于 `domain.com` 域中的任何主机都有效。

禁用浏览器高速缓存

由于可通过网关组件仅使用 Web 浏览器从任何地方安全地访问后端公司数据，因此客户机不应在本地对信息进行高速缓存。

通过修改特定网关的 `platform.conf` 文件中的属性，可以禁止通过网关对重定向的页面进行高速缓存。

禁用该选项会影响网关性能。每当刷新标准“Portal 桌面”，网关就必须检索该页所引用的所有内容，例如先前可能已经被浏览器高速缓存的图像。然而，启用该功能意味着远程访问安全内容将不会在客户端站点留下高速缓存的痕迹。如果从不受公司 IT 控制的网吧或类似的远程位置访问公司网络，此因素可能比性能问题更重要。

参见第 236 页中的 “禁用浏览器高速缓存”。

自定义网关服务用户界面

本节讨论可以编辑的各种网关属性文件。

修改 srapGateway.properties 文件

您可以编辑该文件用于以下目的：

- 自定义网关运行时可能会出现的错误消息。
  - HTML-CharSets=ISO-8859-1 指定用于创建此文件的字符集。
  - 括号中的数字（例如 {0}）表示在运行时显示的值。可以更改与该数字关联的标签，或者按照需要重新排列标签。确保标签与要显示的消息相对应，因为数字和消息是关联的。

自定义日志信息。

默认情况下，srapGateway.properties 文件位于 *portal-server-install-root*/SUNWportal/locale 目录之下。网关计算机上出现的所有消息都在该文件中，而与消息语言无关。

要更改客户机标准“Portal 桌面”上显示的消息的语言，请将该文件复制到相应的语言环境的目录中，例如 *portal-server-install-root*/SUNWportal/locale_en_US。

修改 srapgwadminmsg.properties 文件

您可以出于以下目的编辑该文件：

- 定制在管理控制台中网关服务的按钮上显示的标签。
- 定制配置网关时所显示的状态消息和错误消息。

共享 LDAP 目录

当 Portal Server 和 Access Manager 服务器的两个实例共享相同的 LDAP 目录时，Portal Server、Access Manager 和网关的所有后续实例均共享相同的 LDAP 目录。参见第 237 页中的 “共享 LDAP 目录”。

◆ ◆ ◆ 第 3 章

3 使用 Proxylet

本章说明 Proxylet，它使用户能够通过网关访问内联网 Web 页而无需解析 Web 页。

使用 Proxylet

Proxylet 概述

Proxylet 是可以将自身设置为客户机上的代理服务器的 Java applet。Proxylet 读取并修改客户机上“代理自动配置”(Proxy Auto Config, PAC) 文件中的代理设置，以使代理设置指向本地代理服务器 (Proxylet)。

Proxylet 从网关继承传输模式。如果将网关配置为在 SSL 上运行，则 Proxylet 会在客户机和网关或目标服务器之间建立一个安全通道。对于加密，如果客户机 JVM 为 1.4 或更高版本，或者所需的 jar 文件位于客户机上，则 Proxylet 使用 JSSE API。否则使用 KSSL API。解密在客户机上进行。

在网关配置文件中指定定向到网关的 URL 的域和子域。如果 URL 不是网关处理的域的一部分，则请求将重定向到 Internet。如果一个特定 URL 域在网关配置文件中列出，则 Proxylet 会重置客户机代理设置以指向网关。

如果在网关处启用了“个人数字证书”(Personal Digital Certificate, PDC)，则 Proxylet 支持客户端验证。要检查是否启用了 PDC，参见第 52 页中的 “获取客户机信息”。

Proxylet 是从 Portal Server 管理控制台中启用的，也可以在管理控制台中指定客户机 IP 地址或代理主机名和端口。如果启用了 Proxylet，则它将检查客户机的以下信息：

- 相应的浏览器权限
- 浏览器是否为 IE 6.0 sp2、IE 7 以及 Firefox 2.0
- 计算机或设备是否可以运行服务器应用程序

如果满足所有要求，则会下载 applet 并在客户机上启动。如果客户机没有安装 JRE 1.4.2 或更高版本，将自动使用 Proxylet 下载 JRE（如果您拥有 Internet 连接和管理权限）。

使用 Proxylet 时，它会从“代理自动配置”(Proxy Auto Configuration, PAC) 文件或代理配置列表检索代理设置。

注 – 确保用户了解当使用 Proxylet applet 时，必须禁用浏览器弹出式窗口拦截器。

HTTPS 支持

Proxylet 支持 HTTPS，具有以下功能：

- 在客户端服务器执行解密。
- 可以访问以 SSL 模式运行的目标服务器。
- 客户机证书直接提交给目标服务器。
- 网关不支持基本验证单点登录 (SSO)。（网关不能在 http 报头中插入 SSO 信息。）
- 仅支持基于主机的访问控制，不支持基于 URL 的访问控制。
- 当前不支持网关之前的外部加速器和外部反向代理。

注 – 当 Portal Server 使用 HTTPS 时，此支持不适用于 Proxylet。

使用 Proxylet 的优点

与重写器不同，Proxylet 很少或根本不需要进行安装后更改。与第三方软件（如 Microsoft Exchange Server）的集成非常简单。另外，Proxylet 不触及 Web 内容，因而网关性能得到了提高。由于 Proxylet 不会修改内容或更改数据，因此用户可以下载任何类型的内容，如 `tar` 和 `gzip` 文件。

配置 Proxylet

有关启用和配置 Proxylet 的信息，参见第 13 章。

注 – 如果用户没有相应的 Java 虚拟机 (Java Virtual Machine, JVM) 来运行 Proxylet，浏览器会连接到 Sun Web 站点以下载“Java 运行时环境”。如果用户的浏览器设置未包含正确的值，或用户正在使用直接代理设置而未访问至 Internet，则无法下载 Proxylet。

◆ ◆ ◆ 第 4 章

使用重写器

Secure Remote Access 的重写器组件允许用户利用解析 Web 页面来通过网关访问内联网 Web 页面。

本章包括以下主题：

重写器简介

借助 Secure Remote Access 的重写器组件，最终用户可以浏览内联网，方法是修改 Web 页上引用的“统一资源标识符”(Uniform Resource Identifier, URI)，以使其指向网关。URI 定义了一种在任何注册名称空间中封装名称的方法，并用名称空间对名称进行标记。最常用的 URI 类型为统一资源定位符 (Uniform Resource Locators, URL)。重写器仅支持 HTTP 或 HTTPS。此支持与协议的大小写无关。重写器仅支持出现在相对 URL 中的反斜线符号。

示例 4–1 重写 URL

`http://abc.sesta.com\\index.html` 会被重写。

而以下这些 URL 不会被重写：`http:\\\\abc.sesta.com`。 *http:/abc.com*

字符集编码

HTTP 标准要求 HTTP 报头或 HTML 元标记为网页指定一个字符集。但有时无法获得此信息。字符集必须已知，以便设置数据编码并按照创建者的意图显示数据。

要检测字符集，请从 Java Enterprise System Accessory CD 安装 SUNWjchdt 软件包。如果安装了此产品，重写器会检测到它，并在需要时使用。

注 – 使用此产品可能会影响性能，因此只有需要时才应安装。有关安装、配置以及使用的详细信息，参见 jcharset_readme.txt。

重写器使用方案

当用户想通过网关访问内联网网页时，可使用重写器来获得网页。重写器由 URLScraper 和网关使用。

URLScraper

URL Scraper 提供者可从配置的 URI 获取内容。将这些 URI 发送到浏览器之前，它会将所有相对 URI 扩展为绝对 URI。

例如，如果用户尝试用以下方式访问站点：

```
<a href="../mypage.html">
```

重写器会将其转换为：

```
<a href="http://yahoo.com/mypage.html">
```

其中 http://yahoo.com/test/ 为页面的基 URL。

有关 URLScraper 提供者的详细信息，参见 Sun Java System Portal Server 管理指南。

网关

网关从 Internet 门户获取内容。在将内容发送到浏览器之前，它将网关 URI 追加到现有 URI 之前，使来自浏览器的后续 URI 请求可到达网关。

例如，如果有某位用户尝试用以下方式访问某台 Internet 机器上的 HTML 页面：

```
<a href="http://mymachine.intranet.com/mypage.html>"
```

重写器会在该 URL 前加上一个指向网关的引用，如下所示：

```
<a href="https://gateway.company.com/http://mymachine.intranet.com/ mypage.html>"
```

当用户单击与该锚定点相关联的某个链接时，浏览器便会与网关联络。网关会从 mymachine.intranet.com 中获取 mypage.html 的内容。

网关使用若干规则来确定要重写已获取网页的哪些要素。

编写规则集

有关定义规则集的详细信息，参见 Portal Server 管理指南。创建新规则集后，需要定义所需的规则。

本节涵盖以下主题：

公共接口（规则集 DTD）

规则集 DTD：

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
The following constraints are not represented in DTD, but taken care of programmatically
    1. In a Rule, All Mandatory attributes cannot be "*".
    2. Only one instance of the below elements is allowed, but in any order.
    1)HTMLRules
    2)JSRules
    3)XMLRules
    3. ID should always be in lower case.
-->
<!ENTITY % eURL 'URL'>
<!ENTITY % eEXPRESSION 'EXPRESSION'>
<!ENTITY % eDHTML 'DHTML'>
<!ENTITY % eDJS 'DJS'>
<!ENTITY % eSYSTEM 'SYSTEM'>

<!ENTITY % ruleSetElements '(HTMLRules | JSRules | XMLRules)?'>
<!ENTITY % htmlElements '(Form | Applet | Attribute)*'>
```

```
<!ENTITY % jsElements '(Variable | Function)*'>
<!ENTITY % xmlElements '(Attribute | TagText)*'>

<!ELEMENT RuleSet (%ruleSetElements;,%ruleSetElements;,%ruleSetElements;)>
<!ATTLIST RuleSet
    id ID #REQUIRED
    extends CDATA "none"
>

<!-- Rules for identifying rules in HTML content -->
<!ELEMENT HTMLRules (%htmlElements;)>
<!ELEMENT Form EMPTY>
<!ATTLIST Form
    name CDATA #REQUIRED
    field CDATA #REQUIRED
    valuePatterns CDATA ""
    source CDATA "*"
>

<!ELEMENT Applet EMPTY>
<!ATTLIST Applet
    code CDATA #REQUIRED
    param CDATA "*"
    valuePatterns CDATA ""
    source CDATA "*"
>

<!-- Rules for identifying rules in JS content -->
<!ELEMENT JSRules (%jsElements;)>
<!ELEMENT Variable EMPTY>
<!ATTLIST Variable
    name CDATA #REQUIRED
    type (%eURL; | %eEXPRESSION; | %eDHTML; | %eDJS; | %eSYSTEM;) "EXPRESSION"
    source CDATA "*"
>

<!ELEMENT Function EMPTY>
<!ATTLIST Function
    name CDATA #REQUIRED
    paramPatterns CDATA #REQUIRED
    type (%eURL; | %eEXPRESSION; | %eDHTML; | %eDJS;) "EXPRESSION"
    source CDATA "*"
>

<!-- Rules for identifying rules in XML content -->
<!ELEMENT XMLRules (%xmlElements;)>
<!ELEMENT TagText EMPTY>
<!ATTLIST TagText
```

```
    tag CDATA #REQUIRED
    attributePatterns CDATA ""
    source CDATA "*"
>

<!ELEMENT Attribute EMPTY>
<!ATTLIST Attribute
    name CDATA #REQUIRED
    tag CDATA "*"
    valuePatterns CDATA ""
    type (%eURL; | %eDHTML; | %eDJS; ) "URL"
    source CDATA "*"
>
```

注 – 除必需的属性值不能只为 * 以外，您可以使用 * 作为规则值的一部分。此类规则会被忽略，但会将消息记录在 RuleSetInfo 日志文件中。有关该日志文件的信息，参见第 89 页中的 “调试文件名称” 。

XML DTD 示例

本节包含一个示例规则集。第 140 页的 “案例研究” 用于举例说明重写器是如何解释这些规则的。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
Rules for integrating a mail client with the gateway.
-->
<!DOCTYPE RuleSet SYSTEM "jar://rewriter.jar/resources/RuleSet.dtd">
<RuleSet type="GROUPED" id="owa">
<HTMLRules>
<Attribute name="action" />
<Attribute name="background" />
<Attribute name="codebase" />
<Attribute name="href" />
<Attribute name="src" />
<Attribute name="lowsrc" />
<Attribute name="imagePath" />
<Attribute name="viewClass" />
<Attribute name="emptyURL" />
<Attribute name="draftsURL" />
<Attribute name="folderURL" />
<Attribute name="prevMonthImage" />
<Attribute name="nextMonthImage" />
<Attribute name="style" />
<Attribute name="content" tag="meta" />
</HTMLRules>
<JSRules>
```

```
<!-- Rules for Rewriting JavaScript variables in URLs -->
<Variable name="URL"> _fr.location </Variable>
<Variable name="URL"> g_szUserBase </Variable>
<Variable name="URL"> g_szPublicFolderUrl </Variable>
<Variable name="URL"> g_szExWebDir </Variable>
<Variable name="URL"> g_szViewClassURL </Variable>
<Variable name="URL"> g_szVirtualRoot </Variable>
<Variable name="URL"> g_szBaseURL </Variable>
<Variable name="URL"> g_szURL </Variable>
<Function name="EXPRESSION" name="NavigateTo" paramPatterns="y"/>
</JSRules>
<XMLRules>
<Attribute name="xmlns"/>
<Attribute name="href" tag="a"/>
<TagText tag="baseroot" />
<TagText tag="prop2" />
<TagText tag="prop1" />
<TagText tag="img" />
<TagText tag="xsl:attribute"
attributePatterns="name=src" />
</XMLRules>
</RuleSet>
```

规则编写步骤

编写规则的一般程序为：

- 确定哪些目录包含内容需重写的 HTML 页。
- 在这些目录中，确定需重写的页。
- 确定各页需重写的 URL。确定大多数 URL 的简便方法是搜索“http”和“/”。
- 确定 URL 的内容类型：HTML、JavaScript 或 XML。
- 编写上述各 URL 所需的重写规则，这可通过在 Access Manager 管理控制台的 Portal Server 配置下编辑重写器服务中的必需规则集来完成。
- 将所有规则合并到该域的规则集中。

规则集指导原则

当创建规则集时，请谨记以下内容：

- 特定主机的优先顺序以匹配最长的 URL 为基础。例如，对于以下规则集

  ```
  mail1.central.abc.com|iplanet_mail_ruleset
  *.sfbay.abc.com|sfbay_ruleset
  *.abc.com|generic_ruleset
  ```

  由于 `sfbay_ruleset` 匹配最长，因此使用它。

- 规则集中的规则会依次应用于页面中的每条语句，直到有一项规则与某条语句相匹配为止。

  编写规则时，切记不要忘了规则的顺序。规则是按它们在规则集中的出现顺序应用于页中的语句的。如果既有特定规则又有包含 "*" 的一般规则，要先定义特定规则，然后再定义一般规则。否则，一般规则将先于特定规则应用于所有语句。
- 所有规则都需要包括在 <RuleSet> </RuleSet> 标记内。
- 在规则集的 <HTMLRules> </HTMLRules> 部分加入需要重写 HTML 内容的所有规则。
- 在规则集的 <JSRules> </JSRules> 部分加入需要重写 JavaScript 内容的所有规则。
- 在规则集的 <XMLRules> </XMLRules> 部分加入需要重写 XML 内容的所有规则。
- 在内联网页中，确定需要重写的 URL，并在规则集的适当部分（HTML、JSRules 或 XMLRules）加入所需规则。
- 将规则集分配给所需的域。
- 重新启动网关以使所有更改生效：

  gateway-install-root/SUNWportal/bin/gateway -n *gateway-profile-name* start

定义规则集根元素

规则集根元素有两个属性：

- RuleSetName，例如，default_ruleset。在规则集到 URI 的映射中会引用此名称。
- Extends。该属性是指规则集的继承功能。该值指向您希望从中派生出规则集的规则集。

  可使用值 none 表示这个新的独立规则集不依赖于其他任何规则集，或者指定 *RuleSetName* 表示您的规则集依赖于另一规则集。

使用递归功能

重写器使用递归功能对同一模式进行搜索，直到匹配字符串模式的末尾。

例如，重写器解析以下字符串时：

```
<a href="src=abc.jpg,src=bcd.jpg,src=xyz.jpg>
```

规则

```
<Attribute name="href" valuePatterns="*src=**"/>
```

仅重写首先出现的模式，其形式如下：

```
<a href="src=http://jane.sun.com/abc.jpg>
```

如果使用递归选项

```
<Attribute name="href" valuePatterns="REC:*src=**"/>;
```

重写器会对同一模式进行搜索，直到匹配字符串模式的末尾，因此输出为：

```
<a
href="src=http://jane.sun.com/abc.jpg,src=http://jane.sun.com/bcd.jpg,src=http://jane.sun
.com/xyz.jpg>
```

定义基于语言的规则

规则建立在下列语言基础上：

- HTML
- JavaScript
- XML

HTML 内容规则

网页中的 HTML 内容可进一步分成属性、表单和 applet。相应地，HTML 内容规则分为以下几类：

- 第 66 页中的 “HTML 内容属性规则”
- 第 68 页中的 “HTML 内容的表单规则”
- 第 69 页中的 “HTML 内容的 Applet 规则”

HTML 内容属性规则

该规则用于确定标记都有哪些属性的值需要重写。属性值可以是简单 URL，也可以是 JavaScript 或 DHTML 内容。例如：

- "img" 标记中指向某个图像位置的 `src` 属性（简单 URL）
- href 属性中用于处理链接单击操作的 `onClick` 属性 (DJS)

本节说明以下内容：

- 第 66 页中的 “属性规则语法”
- 第 67 页中的 “属性规则示例”
- 第 67 页中的 “DJS 属性示例”

属性规则语法

```
<Attribute name="attributeName" [tag="*" valuePatterns="" source="*" type="URL|DHTML|DJS"]/>
```

其中：

`attributeName` 是属性的名称（强制项）

`tag` 是属性所属的标记（可选项，默认值是 `*`，指任何标记）

`valuePatterns` 参见第 71 页中的 “在规则中使用模式匹配”

`source` 指定在其中定义该属性的页的 URI（可选项，默认值是 `*`，指在任何页中）

`type` 指定值的类型（可选项）。它们可以是：

`URL` - 简单 URL（默认值）。

`DHMTL` - DHTML 内容。这种内容以标准 HTML 内容的形式显示，并且用于 Microsoft HTC 格式文件。

`DJS` - JavaScript 内容。所有 HTML 事件处理程序（如 onClick 和 onMouseover）都用此 HTML 属性嵌入 JavaScript。

属性规则示例

假定页的基 URL 为：

```
http://mymachine.intranet.com/mypage.html
```

页面内容：

```
<a href="http://mymachine.intranet.com/mypage.html">
```

规则

```
<Attribute name="href"/>
或
<Attribute name="href" tag="a"/>
```

输出

```
<a href=gateway-URL/http://mymachine.intranet.com/myhome.html>
```

描述

由于要重写的 URL 已是一个绝对 URL，所以只在此 URL 前加上了网关 URL。

DJS 属性示例

假定页的基 URL 为：

```
http://abc.sesta.com/focus.html
```

页面内容：

```
<Form>

<input TYPE=TEXT SIZE=20 value=focus
onClick="Check(\q/focus.html\q,\qfocus\q);return;">
```

```
</Form>
```

规则

```
<Attribute name="onClick" type="DJS"/>
<Function type="URL" name="Check" paramPatterns="y,"/>
```

输出

```
<Form>

<INPUT TYPE=TEXT SIZE=20 value=focus onClick="Check(\q
gateway-URL
/http://abc.sesta.com/focus.html\q,\qfocus\q);return;">

</Form>
```

描述

需要两项规则来重写指定的页内容。第一项规则确定 `onClick` JavaScript 标志。第二项规则确定 `check` 函数需要重写的参数。在本例中，只会重写第一个参数，因为 `paramPatterns` 用值 `y` 代替了第一个参数。

会在所需参数前加上网关 URL 以及这些 JavaScript 标志所在页的基 URL。

HTML 内容的表单规则

用户浏览的 HTML 页可能会包含表单。一些表单元素可能会以 URL 作为值。

本节分为下列各小部分：

- 第 68 页中的 “表单规则语法”
- 第 69 页中的 “表单规则示例”

表单规则语法

```
<Form name="form1" field="visit" [valuePatterns="" source="*"]/>
```

其中

`name` 是表单的名称（强制项）

`field` 是表单中的字段，其中含有需要重写的值（强制项）

`valuePatterns` 参见第 71 页中的 “在规则中使用模式匹配”

`source` 是该表单定义所在 html 页的 URL（可选项，默认值是 `*`，指在任何页中）

表单规则示例

假定页的基 URL 为：

```
http://test.siroe.com/testcases/html/form.html
```

页内容

假定页 URI 是 `form.html`，并且它位于服务器的根目录下。

```
<form name=form1  method=POST action=
"http://test.siroe.com/testcases/html/form.html">
<input type=hidden name=abc1 value="0|1234|/test.html">
</form>
```

要重写 `form1` 中名为 `abc1` 的隐藏字段值中出现的 `/text.html`，需要下列规则。

规则

```
<Form source="*/form.html" name="form1"
field="abc1" valuePatterns="0|1234|"/>
<Attribute name="action"/>
```

输出

```
<FORM name="form1"
method="POST" action="gateway-URL/
http://test.siroe.com/testcases/html/form.html">
<input type=hidden name=abc1
value="0|1234|gateway-URL/
http://test.siroe.com/test.html">
</FORM>
```

描述

`action` 标记是用某些已定义的 HTML 属性规则进行重写的。

输入标记属性值的 `value` 的重写方式如输出中所示。将会查找指定的 `valuePatterns`，并通过在前面加上网关 URL 以及页的基 URL 来重写紧随在匹配的 `valuePatterns` 之后的所有内容。参见第 71 页中的 “在规则中使用模式匹配”。

HTML 内容的 Applet 规则

单个网页可以包含许多 applet，而且每个 applet 可以包含许多参数。重写器将用规则中指定的值与 applet 的 HTML 定义进行匹配，并修改 applet 参数定义中出现的 URL 值。此替换在服务器处执行，用户浏览特定网页时并不会执行。此项规则会确定并重写 applet 以及 HTML 内容对象标记中的参数。

本节分为下列各小部分：

Applet 规则语法

```
<Applet code="ApplicationClassName/ObjectID
" param="parametername" [valuePatterns="" source="*"] />
```

其中

`code` 是 applet 或对象类的名称（强制项）

`param` 是值需要重写的参数的名称（强制项）

`valuePatterns` 参见第 71 页中的 “在规则中使用模式匹配”

`source` 是包含 applet 定义的页的 URL（可选项，默认值是 `*`，指在任何页中）

Applet 规则示例

假定页的基 URL 为：

```
http://abc.siroe.com/casestudy/test/HTML/applet/rule1.html
```

页面内容：

```
<applet codebase="appletcode" code="RewriteURLinApplet.class" archive="/test.jar">
<param name=Test1 value="/index.html">
</applet>
```

规则

```
<Applet source="*/rule1.html" code=
"RewriteURLin*.class" param="Test*"/>
```

输出

```
<APPLET codebase="gateway-URL
/http://abc.siroe.com/casestudy/test/HTML/
applet/appletcode" code="RewriteURLinApplet.class"
archive="/test.jar"><param name="Test1" value="
gateway-URL/http:
//abc.siroe.com/index.html">
</APPLET>
```

描述

由于 `<Attribute name="codebase"/>` 是 `default_gateway_ruleset` 中的一项已定义规则，因此会重写 `codebase` **属性**。

名称以 Test 开头的所有参数均会被重写。并且，会在 param 标记的 value 属性前加上 applet 代码所在页的基 URL 以及网关 URL。

在规则中使用模式匹配

您可以使用 valuePatterns 字段实现模式匹配并指定语句中需要重写的特定部分。

如果指定 valuePatterns 作为规则的一部分，将会重写紧随在匹配模式之后的所有内容。

请考虑下面的表单规则示例。

```
<Form source="*/source.html
" name="form1" field="visit
" [valuePatterns="0|1234|"]/>
```

其中

source 是显示表单的 html 页的 URL。

name 是表单的名称。

field 是表单中的字段，需要重写它的值。

valuePatterns 指示需要重写字符串中的哪个部分。将会重写 valuePatterns 后出现的所有内容（可选项，默认值是 ""，指需要重写整个值）

在 valuePatterns 中指定专用字符

可以通过使用反斜杠对专用字符进行转义来指定这些字符。例如：

```
<Form source="*/source.html " name="form1" field=" visit" [valuePatterns="0|1234| \\
;original text|changed text"]/>
```

在 valuePatterns 中使用通配符

您可以使用通配符星号 (*) 字符来实现重写的模式匹配。

不能在 valuePatterns 字段中仅指定一个 *。因为 * 表示与所有文本匹配，所以 valuePattern 后面没有文本。这样，重写器就没有要重写的文本。您必须将 * 与另一字符串连起来使用，例如 *abc。此时，会重写紧随在 *abc 之后的所有内容。

注 – 可在规则的任何字段中使用星号 (*) 作为通配符。不过，规则中的所有字段不能全都包含 *。如果所有字段都包含 *，则会忽略该规则。不会显示错误消息。

可以与原始语句中出现的分隔符（分号或逗号）联合使用 * 或 ** 来分隔多个字段。一个星号 (*) 匹配任何不进行重写的字段，而两个星号 (**) 匹配任何需要重写的字段。

第 71 页中的 “在 valuePatterns 中使用通配符” 列出了 * 通配符的一些用法示例。

表 4–1 * 通配符用法示例

URL	valuePatterns	描述
url1, url2, url3, url4	valuePatterns = "**, *, **, *"	由于 ** 指示了要重写的部分，因此会重写 url1 和 url3。
XYZABChttp://host1.sesta.com/dir1.html	valuePatterns = "*ABC"	仅重写 http://host1.sesta.com/dir1.html 部分。需要重写 *ABC 后的所有内容。
"0|dir1|dir2|dir3|dir4|test|url1	valuePatterns = "*|*|**|*|**|*|"	重写 dir2、dir4 和 url1。最后一个需要重写的字段不必用 ** 指出。

JavaScript 内容规则

JavaScript 可以在各种不同位置包含 URL。重写器不能直接解析 JavaScript 并确定出 URL 部分。需要编写一组特殊的规则来帮助 JavaScript 处理器确定和转换 URL。

具有 URL 类型的 JavaScript 元素分类如下：

- 第 72 页中的 “变量”
- 第 79 页中的 “函数参数”

变量

变量的通用语法为：

<Variable name="variableName" [type="URL|EXPRESSION|DHTML|DJS|SYSTEM" source="*"]>

根据 JavaScript 变量所含值的类型，可将它们细分为以下 5 类：

- 第 72 页中的 “URL 变量”
- 第 74 页中的 “EXPRESSION 变量”
- 第 75 页中的 “DHTML（动态 HTML）变量”
- 第 76 页中的 “DJS（动态 JavaScript）变量”
- 第 77 页中的 “SYSTEM 变量”

URL 变量

变量值为可作为 URL 对待的简单字符串。

本节分为下列各小部分：

- 第 73 页中的 “URL 变量语法”

- 第 73 页中的 “URL 变量示例”

URL 变量语法

```
<Variable name="variableName" type="URL" [source="*"]>
```

其中

`variableName` 是变量的名称。`variableName` 的值会被重写（强制项）

`type` 是 URL 变量（强制项，其值必须是 URL）

`source` 是该 JavaScript 变量所在页的 URI（可选项，默认值是 `*`，指在任何页中）

URL 变量示例

假定基 URL 为：

```
http://abc.siroe.com/tmp/page.html
```

页内容

```
<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc1="/tmp/tmp.jpg";
var imgsrc2="http://srap.sesta.com/tmp/tmp.jpg";
var imgsrc3=imgsrc2;
//-->
</SCRIPT>
```

规则

```
<Variable name="imgsrc*" type="URL"/>
```

输出

```
<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc="gateway-URL/http://abc.siroe.com/tmp/tmp.jpg";
var imgsrc="gateway-URL/http://srap.sesta.com/tmp/tmp.jpg";
var imgsrc3=imgsrc2;
//-->
</SCRIPT>
```

描述

会重写 URL 类型并且名称以 imgsrc 开头的所有变量。对于输出中的第一行，在其前面加上了网关 URL 以及变量所在页的基 URL。第二行已包含绝对路径，因此只在其前面加上了网关 URL。由于第三个 var imagsrc2 的值不是字符串，而是其他 JavaScript 值，所以不会对其进行重写。

EXPRESSION 变量

表达式变量的右侧是一个表达式。该表达式的结果是一个 URL。由于重写器无法对服务器上的此类表达式求值，所以它会向 HTML 页追加一个 JavaScript 函数 (psSRAPRewriter_convert_expression)。该函数将此表达式视为一个参数，并在客户机浏览器中对其进行求值以得出所需的 URL。

如果不确定语句中包含的是简单 URL 还是 EXPRESSION URL，则使用 EXPRESSION 规则，因为它可以处理这两种情形。

本节分为下列各小部分：

- 第 74 页中的 “EXPRESSION 变量语法”
- 第 74 页中的 “EXPRESSION 变量示例”

EXPRESSION 变量语法

```
<Variable name="variableName" [type="EXPRESSION" source="*"]/>
```

其中

variableName 是值为表达式的 JavaScript 变量的名称（强制项）

type 是 JavaScript 变量的类型（可选项，默认值是 EXPRESSION）

source 是页的 URI（可选项，默认值是 *，指任何源）

EXPRESSION 变量示例

假定页的基 URL 为：

```
http://abc.siroe.com/dir1/dir2/page.html
```

页内容

```
<script LANGUAGE="Javascript">
<!--
//Expression variables
var expvar= getURIPreFix() + "../../images/graphics"+".gif";
document.write("<A HREF="+expvar+">Link to XYZ content</A><P>")
var expvar="../../images/graphics"+".gif";
//-->
</SCRIPT>
```

规则

```
<Variable name="expvar" type="EXPRESSION"/>
或
<Variable name="expvar"/>
```

输出

```
var expvar=psSRAPRewriter_convert_expression(getURIPreFix()
 + "../../images/graphics"+".gif");document.write("<a href="+expvar+">>
Link to XYZ content</A><P>")var expvar="gateway-URL/http://abc.siroe.com/images/graphics"+".gif";
```

描述

会在第一行的表达式变量 `expvar` 右侧的前面加上函数 `psSRAPRewriter_convert_expression`。该函数会在运行时处理此表达式并重写相应内容。在第三行中，值被重写为一个简单的 URL。

DHTML（动态 HTML）变量

这些变量是包含 HTML 内容的 JavaScript 变量。

本节分为下列各小部分：

- 第 75 页中的 “DHTML 语法”
- 第 75 页中的 “DHTML 示例”

DHTML 语法

```
<Variable name="variableName" type="DHTML" [source="*"]/>
```

其中

`variableName` 是具有 DHTML 内容的 JavaScript 变量的名称（强制项）

`type` 是变量的类型（强制项，其值必须是 DHTML）

`source` 是页的 URL（可选项，默认值是 `*`，指在任何页中）

DHTML 示例

假定页的基 URL 为：

```
http://abc.sesta.com/graphics/set1/
graphics/jsscript/JSVAR/page.html
```

页内容

```
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a href=../../images/test.html>"
var dhtmlVar="<a href=/images/test.html>"
var dhtmlVar="<a href=images/test.html>"
//-->
</SCRIPT>
```

规则

```
<Variable name="dhtmlVar" type="DHTML"/>
<Attribute name="href"/>
或
<Attribute name="href" tag="a"/>
```

输出

```
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a href=gateway-URL
/http://abc.sesta.com/graphics/
set1/graphics/images/test.html>"
var dhtmlVar="<a href=gateway-URL/
http
://abc.sesta.com/images/test.html>"
var dhtmlVar="<a href=gateway-URL/
http://abc.sesta.com/graphics/set1/
graphics/jscript/JSVAR/images/test.html>"
//--></SCRIPT>
```

描述

JavaScript 解析器会将 dhtmlVar 的值作为 HTML 内容读取，并通过 HTML 解析器发送此内容。HTML 解析器会应用其中有 href 属性规则匹配的 HTML 规则，因此 URL 会被重写。

DJS（动态 JavaScript）变量

这些变量是包含 JavaScript 内容的 JavaScript 变量。

本节分为下列各小部分：

- 第 77 页中的 “DJS 语法”
- 第 77 页中的 “DJS 示例”

DJS 语法

```
<Variable name="variableName" type="DJS" [source="*"]/>
```

其中

`variable` 是值为 javascript 的 JavaScript 变量。

DJS 示例

假定页的基 URL 为 :

```
http://abc.sesta.com/dir1/dir2/dir3/jscript/dir4/page.html
```

页内容

```
//DJS Var
var dJSVar="var dJSimgsrc=\q/tmp/tmp.jpg\q;"
var dJSVar="var dJSimgsrc=\q../tmp/tmp.jpg\q;"
var dJSVar="var dJSimgsrc=
\qhttp://abc.sesta.com/tmp/tmp.jpg\q;"
```

规则

```
<Variable name="DJS">dJSVar/>
<Variable name="URL">dJSimgsrc/>
```

输出

```
//DJS Var - need 2 rules
var dJSVar="var dJSimgsrc=\qgateway-URL
/http://abc.sesta.com/tmp/tmp.jpg\q;"var dJSVar="var dJSimgsrc=\q
gateway-URL/http
://abc.sesta.com/dir1/dir2/dir3/jscript/tmp/tmp.jpg\q;"
var dJSVar="var dJSimgsrc=\qgateway-URL/
http://abc.sesta.com/tmp/tmp.jpg\q;"
```

描述

这里需要两项规则。第一项规则用于查找动态 JavaScript 变量 `dJSVar`。该变量的值同样是 `URL` 类型的 JavaScript。第二项规则用于重写该 JavaScript 变量的值。

SYSTEM 变量

这些变量是指不是由用户声明且只得到有限支持的变量。它们可用作 JavaScript 标准的一部分。例如，`window.location.pathname`。

本节分为下列各小部分 :

SYSTEM 变量语法

```
<Variable name="variableName" type="SYSTEM" [source="*"]/>
```

其中

variableName 是 JavaScript 系统变量（强制项，其值可以是与以下模式匹配的任何值：document.URL、document.domain、location、doument.location、location.pathname、location.href、location.protocol、location.hostname、location.host 和 location.port。上述所有模式都存在于 generic_ruleset 中。不要修改这些系统变量规则。）

type 指定系统类型值（强制项，且值为 DJS）

source 是此页的 URI（可选项，默认值是 *，指在任何页中）

SYSTEM 变量示例

假定页的基 URL 为：

```
http://abc.siroe.com/dir1/page.html
```

页内容

```
<script LANGUAGE="Javascript">
<!--
//SYSTEM Var
alert(window.location.pathname);
//-->
</SCRIPT>
```

规则

```
<Variable name="window.location.pathname" type="SYSTEM"/>
```

输出

```
</SCRIPT>
<SCRIPT LANGUAGE="Javascript">
<!--
//SYSTEM Var
alert(psSRAPRewriter_convert_pathname(window.location.pathname));
//-->
</SCRIPT>
```

描述

重写器会查找与规则匹配的系统变量，然后在其前面加上 `psSRAPRewriter_convert_system` 函数。该函数会在运行时处理此系统变量并相应地重写最终得到的 URL。

函数参数

值需重写的函数参数分为以下 4 类：

- 第 79 页中的 “URL 参数”
- 第 81 页中的 “EXPRESSION 参数”
- 第 82 页中的 “DHTML 参数”
- 第 84 页中的 “DJS 参数”

一般语法

```
<Function name="functionName " paramPatterns="y,y,"
[type="URL|EXPRESSION|DHTML|DJS" source="*"]/>
```

其中

`name` 是 JavaScript 函数的名称（强制项）

`paramPatterns` 指定需要重写的参数（强制项）

`y y` 的位置指示需要重写的参数。例如，在语法中，第一个参数需要重写，但不能重写第二个参数

`type` 指定该参数所需值的类型（可选项，默认值是 EXPRESSION 类型）

`source` 是页的源 URI（可选项，默认值是 `*`，指在任何页中）

URL 参数

函数将该参数视为一个字符串并且该字符串可以作为 URL 对待。

本节分为下列各小部分：

- 第 79 页中的 “URL 参数语法”
- 第 80 页中的 “URL 参数示例”

URL 参数语法

```
<Function name="functionName" paramPatterns="y,," type="URL" [source="*"]/>
```

其中

`name` 是具有 URL 类型参数的函数的名称（强制项）

`paramPatterns` 指定需要重写的参数（强制项）

y y 的位置指示需要重写的参数。例如，在语法中，第一个参数需要重写，但不能重写第二个参数

type 是函数的类型（强制项，其值必须是 URL）

source 是具有该函数调用的页的 URL（可选项，默认值是 *，指在任何 URL 中）

URL 参数示例

假定页的基 URL 为：

```
http://abc.sesta.com/test/rewriter/test1/jscript/test2/page.html
```

页内容

```
<script language="JavaScript">
<!--
function test(one,two,three){
alert(one + "##" + two + "##" +three);
}
test("/test.html","../test.html","123");
window.open("/index.html","gen",width=500,height=500);
//-->
</SCRIPT>
```

规则

```
<Function name="URL" name="test" paramPatterns="y,y,"/>
<Function name="URL" name="window.open" paramPatterns="y,,,"/>
```

输出

```
<SCRIPT language="JavaScript">
<!--
function test(one,two,three) {
alert(one + "##" + two + "##" +three);
}
test("gateway-URL/http://abc.sesta.com/test.html","
gateway-URL/http://abc.sesta.com/test/rewriter/
test1/jscript/test.html","123");window.open("gateway-URL/
http://abc.sesta.com/index.html","gen",width=500,height=500);
//-->
</SCRIPT>
```

描述

第一项规则指定需要重写名为 test 的函数中的前两个参数。因此会重写 test 函数的前两个参数。第二项规则指定需要重写 window.open 函数的第一个参数。会在 window.open 函数中的 URL 前面加上网关 URL 以及包含函数参数的页的基 URL。

EXPRESSION 参数

这些参数接受表达式值，表达式计算结果为 URL。

本节分为下列各小部分：

EXPRESSION 参数语法

```
<Function name="functionName" paramPatterns="y" [type="EXPRESSION" source="*"]/>
```

其中

`name` 是函数的名称（强制项）

`paramPatterns` 指定需要重写的参数（强制项）

`y` `y` 的位置指示需要重写的函数参数。在上述语法中，只会重写第一个参数

`type` 指定 EXPRESSION 值（可选项）

`source` 是其中调用了该函数的页的 URI

EXPRESSION 参数示例

假定页的基 URL 为：

```
http://abc.sesta.com/dir1/dir2/page.html
```

页内容

```
<script language="JavaScript">
<!--
function jstest2(){
return ".html";
}
function jstest1(one){
return one;
}
var dir="/images/test"
var test1=jstest1(dir+"/test"+jstest2());
document.write("<a HREF="+test1+">TEST</a>");
alert(test1);
//-->
</SCRIPT>
```

规则

```
<Function type="EXPRESSION" name="jstest1" paramPatterns="y"/>
或
<Function name="jstest1" paramPatterns="y"/>
```

输出

```
<script language="JavaScript">
<!--
function jstest2(){
return ".html";
}
function jstest1(one){
return one;
}
var dir="/images/test"
var test1=jstest1(psSRAPRewriter_convert_expression(dir+"/test"+jstest2()));
document.write("<a HREF="+test1+">TEST</a>");
alert(test1);
//-->
</SCRIPT>
```

描述

此规则将 `jstest1` 函数的第一个参数视为 EXPRESSION 函数参数，以此来指定需要重写该参数。在示例页内容中，第一个参数是一个表达式，只会在运行时对其进行求值。重写器会在该表达式前加上 `psSRAPRewriter_convert_expression` 函数。此表达式要进行求值，并且 `psSRAPRewriter_convert_expression` 函数会在运行时重写输出结果。

注 – 在上述示例中，不需要在 JavaScript 变量规则中包含 `test1`。`jstest1` 的函数规则会负责执行重写工作。

DHTML 参数

值为 HTML 的函数参数

本机 JavaScript 方法（如可动态生成 HTML 页的 `document.write()`）归属于这一类别。

本节分为下列各小部分：

- 第 83 页中的 “DHTML 参数语法”
- 第 83 页中的 “DHTML 参数示例”

DHTML 参数语法

```
<Function name="functionName" paramPatterns="y" type="DHTML" [source="*"]/>
```

其中

`name` 是函数的名称

`paramPatterns` 指定需要重写的参数（强制项）

`y y` 的位置指示需要重写的函数参数。在上述语法中，只会重写第一个参数

DHTML 参数示例

假定页的基 URL 为：

```
http://xyz.siroe.com/test/rewriter/test1/jscript/JSFUNC/page.html
```

页内容

```
<script>
<!--
document.write(\q<a href="/index.html">write</a><BR>\q)
document.writeln(\q<a href="index.html">writeln</a><BR>\q)
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
//-->
</SCRIPT>
```

规则

```
<Function name="DHTML" name="document.write" paramPatterns="y"/>
<Function name="DHTML" name="document.writeln" paramPatterns="y"/>
<Attribute name="href"/>
```

输出

```
<SCRIPT>
<!--
document.write(\q<a href="gateway-URL/
http://xyz.siroe.com/index.html">write</a><BR>\q)
document.writeln(\q<a href="gateway-URL/
http://xyz.siroe.com/test/rewriter/test1/
jscript/JSFUNC/index.html">writeln</a><BR>\q)
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
//-->
</SCRIPT>
```

描述

第一项规则指定需要重写函数 document.write 中的第一个参数。第二项规则指定需要重写函数 document.writeln 中的第一个参数。第三项规则是一项简单的 HTML 规则，它指定需要重写名为 href 的所有属性。在示例中，DHTML 参数规则将会确定函数中需要重写的参数。然后会应用 HTML 属性规则来实际重写已确定的参数。

DJS 参数

值为 JavaScript 的函数参数。

本节分为下列各小部分：

- 第 84 页中的 “DJS 参数语法”
- 第 84 页中的 “DJS 参数示例”

DJS 参数语法

```
<Function name="functionName" paramPatterns="y" type="DJS" [source="*"]/>
```

其中

name 是含有一个参数 DJS 的函数的名称（强制项）

paramPatterns 指定上述函数中的哪个参数是 DJS（强制项）

y y 的位置指示需要重写的函数参数。在上述语法中，只会重写第一个参数

type 为 DJS（强制项）

source 是页的 URI（可选项，默认值为 *，指任何 URI）

DJS 参数示例

假定页的基 URL 为：

```
http://abc.sesta.com/page.html
```

页内容

```
<script>
menu.addItem(new NavBarMenuItem("All Available Information","JavaScript:top.location=\qhttp://abc.sesta.com\q"));
</script>
```

规则

```
<Function name="DJS" name="NavBarMenuItem" paramPatterns=",y"/>
<Variable name="URL">top.location</Variable>
```

输出

```
<script>
menu.addItem(new NavBarMenuItem("All Available Information",
"JavaScript:top.location=\qgateway-URL/
http://abc.sesta.com\q"));
</script>
```

描述

第一项规则指定需要重写函数 `NavBarMenuItem` 中的第二个包含 JavaScript 的参数。在 JavaScript 中，变量 `top.location` 也需要重写。该变量是使用第二项规则来重写的。

XML 内容规则

网页可以包含 XML 内容，而后者又可以包含 URL。需要重写的 XML 内容分为以下两类：

- 第 85 页中的 “标记文本”（与标记的 PCDATA 或 CDATA 相同）
- 第 86 页中的 “属性”

标记文本

本规则用于重写标记元素的 PCDATA 或 CDATA。

本节分为下列各小部分：

- 第 85 页中的 “标记文本语法”
- 第 86 页中的 “标记文本示例”

标记文本语法

```
<TagText tag="tagName"
[attributePatterns="attribute_patterns_for_ this_tag" source="*"]/>
```

其中

`tagName` 是标记的名称

`attributePatterns` 是与该标记相应的属性及其值模式（可选项，指该标记根本无任何属性）

`source` 是该 xml 文件的 URI（可选项，默认值是 `*`，指任何 xml 页）

标记文本示例

假定页的基 URL 为 :

```
http://abc.sesta.com/test/rewriter/test1/xml/page.html
```

页内容

```
<xml>
<Attribute name="src">test.html</attribute>
<attribute>abc.html</attribute>
</xml>
```

规则

```
<TagText tag="attribute" attributePatterns="name=src"/>
```

输出

```
<xml>
<Attribute name="src">gateway-URL/
http://abc.sesta.com/test/rewriter/test1/
xml/test.html</attribute><attribute>abc.html</attribute>
</xml>
```

描述

页内容的第一行包含第 87 页中的 “属性示例” 。页内容中的第二行不包含具有属性呼叫名称且属性名称值为 `src` 的属性，因此不会进行任何重写。要重写该属性，也需要有 `<TagText tag="attribute"/>`

属性

XML 属性规则与 HTML 的属性规则类似。二者的区别在于，XML 的属性规则区分大小写，而 HTML 属性规则不区分大小写。这同样是因为 XML 中内置了对大小写的敏感性而 HTML 中则没有。

重写器会基于属性名称来转换属性值。

本节分为下列各小部分 :

属性语法

```
<Attribute name="attributeName " [tag="*" type="URL" valuePatterns="*" source="*"
]/>
```

其中

`attributeName` 是属性的名称（强制项）

`tag` 是该属性所在标记的名称（可选项，默认值是 `*`，指任何标记）

`valuePatterns` 参见第 71 页中的 “在规则中使用模式匹配”

`source` 是该 XML 页的 URI（可选项，默认值是 `*`，指在任何 XML 页中）

属性示例

假定页的基 URL 为：

```
http://abc.sesta.com/test/rewriter/test1/xml/page.html
```

页内容

```
<xml>
<baseroot href="/root.html"/>
<img href="image.html"/>
<string href="1234|substring.html"/>
<check href="1234|string.html"/>
</xml>
```

规则

```
<Attribute name="href"tag="check" valuePatterns="1234|"/>
```

输出

```
<xml>
<baseroot href="/root.html"/><img href="image.html"/>
<string href="1234|substring.html"/><check href="1234|
gateway-URL
/http://abc.sesta.com/test/rewriter/test1/xml/string.html"/></xml>
```

描述

在上述示例中，只会重写第四行，因为它满足规则中指定的所有条件。参见第 71 页中的 “在规则中使用模式匹配”。

层叠样式表规则

HTML 页中的“层叠样式表”（包括 CCS2）会进行转换。由于 URL 只在 CSS 的 `url()` 函数和导入语法中出现，因此没有为这种转换定义任何规则。

WML 规则

WML 与 HTML 类似，因此 HTML 规则适用于 WML 内容。使用 WML 内容的一般规则集。参见 第 66 页中的 “HTML 内容规则” 。

使用递归功能

重写器使用递归功能对同一模式进行搜索，直到匹配字符串模式的末尾。

例如，重写器解析以下字符串时：

```
<a href="src=abc.jpg,src=bcd.jpg,src=xyz.jpg>
```

规则

```
<Attribute name="href" valuePatterns="*src=**"/>
```

仅重写首先出现的模式，形式如下：

```
<a href="src=http://jane.sun.com/abc.jpg>
```

但是，如果您如下使用递归选项

```
<Attribute name="href" valuePatterns="REC:*src=**"/>;
```

重写器会对同一模式进行搜索，直到匹配字符串模式的末尾，因此输出为：

```
<a href="src=http://jane.sun.com/abc.jpg,src=
http://jane.sun.com/bcd.jpg,src=http://jane.sun.com/xyz.jpg>
```

使用调试日志排除故障

要排除重写器故障，需要启用调试日志。

“调试消息”分为以下几类。

- 错误 – 重写器无法从中恢复的错误。
- 警告 – 对重写器的功能没有严重影响的警告。重写器能够恢复这类错误，但无法保证是否会造成异常行为。警告中显示的一些消息是为了提供信息。例如，“未重写图像内容”会作为警告消息被记录下来。这很合理，因为不允许重写器重写图像。
- 消息 – 重写器提供的最高级别的信息。

设置重写器调试级别

▼ 设置重写器调试级别

1 **以根用户身份登录到网关机器并编辑以下文件：**

gateway-install-root/SUNWam/config/AMConfig-*instance-name*.properties

2 **设置调试级别：**

com.iplanet.services.debug.level=

调试级别为：

error - 只将严重错误记录到调试文件中。出现此类错误时，重写器通常会停止工作。

warning - 记录警告消息。

message - 记录所有调试消息。

off - 不记录任何调试消息。

3 **在 AMConfig-*instance-name* .properties 文件的以下属性中，为调试文件指定目录：**

com.iplanet.services.debug.directory=/var/opt/SUNWam/debug

其中 /var/opt/SUNWam/debug 是默认调试目录。

4 **从终端窗口中重新启动网关：**

./psadmin start-sra-instance –u amadmin – f <*password file*> –N <*profile name*>– t <*gateway*>

调试文件名称

当调试级别设置为 message 时，调试会生成一组文件。第 89 页中的 “调试文件名称”列出了重写器文件及其中包含的信息。

表 4–2 重写器调试文件

文件名	信息
RuleSetInfo	包含重写时已使用的所有规则集。
Original Pages	包含页面 URI、resolveURI（若不同于页面 URI）、内容 MIME、已应用于该页的规则集、解析器 MIME，以及原始内容。 与解析有关的特定错误/警告/消息也出现在本文件中。 在消息模式下，会记录全部内容。在警告和错误模式下，只记录重写期间出现的异常。

表 4–2 重写器调试文件 （续）

文件名	信息
Rewritten Pages	包含页面 URI、resolveURI（若不同于页 URI）、内容 MIME、已应用于该页的规则集、解析器 MIME，以及重写后的内容。 当将调试模式设置为消息时，将会填写本文件。
Unaffected Pages	包含未经修改的页列表。
URIInfo Pages	包含已找到并经过转换的 URL。该文件会记录内容仍与原始数据相同的所有页的详细信息。 所记录的详细信息有：页 URI、MIME 及编码数据、重写时所用的规则集 ID，以及解析器 MIME。

除了上述文件以外，重写器还会为调试消息生成一个文件，该文件未收入上述文件中。此文件名由两部分组成：第一部分是 `pwRewriter` 或 `psSRARewriter`；第二部分是使用 `portal` 或 *gateway-profile-name* 的扩展名。

调试文件在门户或网关中显示。这些文件在 `AMConfig-`*instance-name*`.properties` 文件指定的目录中。

重写器组件会生成下面的一组文件来帮助进行调试：

*prefix*_RuleSetInfo.*extension*

*prefix*_OrginalPages.*extension*

*prefix*_RewrittenPages.*extension*

*prefix*_UnaffectedPages.*extension*

*prefix*_URIInfo.*extension*

其中

prefix 对于 URLScraper 使用日志为 `psRewriter`，对于网关使用日志为 `psSRAPRewriter`。

extension 对于 URLScraper 使用为 `portal`，对于网关使用为 `gateway-profile-name`。

例如，如果使用网关上的重写器来转换页并且使用了默认网关配置文件，则调试时会创建下列文件：

```
psSRAPRewriter_RuleSetInfo.default

psSRAPRewriter_OriginalPages.default

psSRAPRewriter_RewrittenPages.default

psSRAPRewriter_UnaffectedPages.default
```

```
psSRAPRewriter_URIInfo.default

psSRAPRewriter.default
```

工作示例

本节包括：

- 含有需重写内容的简单 HTML 页
- 重写内容所需的规则
- 相应的已重写 HTML 页

这些示例页在 *portal-server-URL* `/rewriter` 目录下获得。在应用规则之前可以先浏览页面，然后再通过网关查看含有已重写输出的文件，以了解规则的工作方式。在一些示例中，规则已包含在 `default_gateway_ruleset` 中。在一些示例中，您可能需要将规则加入到 `default_gateway_ruleset` 中。这一点会在适当的地方提及。

注 – 某些语句以粗体形式出现，表示已对它们进行了重写。

提供了下列示例：

HTML

- 第 92 页中的 “HTML 属性示例”
- 第 96 页中的 “HTML 表单示例”
- 第 98 页中的 “HTML Applet 示例”

JavaScript

- 变量
  - 第 99 页中的 “JavaScript URL 变量示例”
  - 第 99 页中的 “JavaScript 内容示例”
  - 第 103 页中的 “JavaScript DHTML 变量示例”
  - 第 105 页中的 “JavaScript DJS 变量示例”
  - 第 107 页中的 “JavaScript SYSTEM 变量示例”

函数

- 第 109 页中的 “JavaScript URL 函数示例”
- 第 110 页中的 “JavaScript EXPRESSION 函数示例”
- 第 112 页中的 “JavaScript DHTML 函数示例”
- 第 114 页中的 “JavaScript DJS 函数示例”

XML

- XML 属性示例

HTML 内容示例

HTML 属性示例

▼ 使用 HTML 属性示例

1 **可从以下位置访问本示例：**

portal-server-URL /rewriter/HTML/attrib/attribute.html

2 **确保在网关服务的“域和子域的代理”列表中定义了** `abc.sesta.com` **和** `host1.siroe.com`。

如果没有定义该项，则假定采用直接连接，不会在其前面加网关 URL。

不需要将本范例中指定的规则添加到 `default_gateway_ruleset` 中，因为该规则已定义。

重写前的 HTML

```
<html>
Rewriting starts
<head>
<title>TEST PAGE () </title>
</head>
ID-htmlattr.1
<br><br>
1.a href <a href="http://abc.sesta.com/images/logo.gif">http://..</a>
<br><br>
2. href <a href="https://host1.siroe.com">https://..</a>
<br><br>
3. href <a href="../images/logo.gif">../images/</a>
<br><br>
4. href <a href="images/logo.gif">images/..</a> <br><br>
5. href <a href="../../images/logo.gif">../../images/</a> <br><br>
Rewriting ends
</html>
```

规则

```
<Attribute name="href"/>
```

重写后的 HTML

```
<html>
Rewriting starts
<head>
<title>TEST PAGE () </title>
</head>
ID-htmlattr.1
<br><br>
1. a href <a href="gateway-URL/http://abc.sesta.com/images/logo.gif">http://..</a> <br>
```

// 由于已在 default_gateway_ruleset 中定义了 <Attrib name="href"/> 规则，所以会重写这个 URL。由于此 URL 已是绝对的，因此只在其前面加上了网关 URL。确保在网关服务的“域和子域的代理”列表中定义了 abc.sesta.com。否则，不会在其前面加网关 URL，因为此时将假定采用直接连接。

```
2. href <a href="gateway-URL/https://host1.siroe.com">https://..</a>
```

// 同样，需要在网关服务的“域和子域的代理”列表中定义 host1.siroe.com。否则，不会在其前面加网关 URL，因为此时将假定采用直接连接。

```
<br><br>
3. href <a href="gateway-URL/portal-server-URL/rewriter/HTML/images/logo.gif">../images/</a>
```

// 由于指定了相对路径，因此会随所需子目录一同在其前面加上网关 URL 和 portal-server-URL。此链接不会起作用，因为在所提供的示例结构中，HTML 目录下没有指定名为 images 的目录。

```
<br><br>
```

```
4 href <a href="gateway-URL/portal-server-URL/rewriter/HTML/attrib/images/
logo.gif">images/..</a> <br><br>
```

// 由于指定了相对路径，因此会随所需子目录一同在其前面加上网关 URL 和 Portal Server URL。

```
5. href <a href="gateway-URL/portal-server-URL/rewriter/images/logo.gif">
../../images/</a> <br><br>
```

// 由于指定了相对路径，因此会随所需子目录一同在其前面加上网关 URL 和 Portal Server URL。此链接不会起作用，因为在所提供的示例结构中，Rewriter 目录下没有指定名为 images 的目录。

```
Rewriting ends</html>
```

HTML 动态 JavaScript 标志示例

本节讨论使用 HTML JavaScript 标志示例

▼ 使用 HTML JavaScript 标志示例：

1. 可从以下位置访问本示例：
   portal-server-URL /rewriter/HTML/jstokens/JStokens.html
2. 将本示例中指定的规则添加到“ JavaScript 源重写规则 ”一节的 `default_gateway_ruleset` 中。
3. 在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的 `default_gateway_ruleset`。
4. 从终端窗口中重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML

```
<html>
<head>
Rewriting starts
<script language="javascript">
function Check(test,ind){
if (ind == \qblur\q)
{alert("testing onBlur")}
if (ind == \qfocus\q)
{alert("testing onFocus")}
}
</SCRIPT>
</head>
<body>
<form>
<input TYPE=TEXT SIZE=20 value=blur onAbort="Check
(\q/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=blur onBlur="Check
(\q/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onFocus="Check
(\q/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onChange="Check
(\q/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onClick="Check
(\q/focus.html\q,\qblur\q);return;">
<br><br>
</form>
</body>
```

```
Rewriting ends
</html>
```

规则

```
<Attribute name="onClick" type="DJS"/>
<Function type="URL" name="Check" paramPatterns="y"/>
```

注 – `<Function name="URL" name="Check" paramPatterns="y"/>` 是 JavaScript 函数规则，在 JavaScript 函数示例中对其进行了详细解释。

重写后的 HTML

```
<html>
<head>
Rewriting starts
<script language="javascript">
function Check(test,ind){
if (ind == \qblur\q)
{alert("testing onBlur")}
if (ind == \qfocus\q)
{alert("testing onFocus")}
}
</SCRIPT>
</head>
<body>
<form>
<input TYPE=TEXT SIZE=20 value=blur onAbort="Check
(\qgateway URL/portal-server-URL/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=blur onBlur="Check
(\qgateway URL/portal-server-URL/indexblur.html\q,\qblur\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onFocus="Check
(\qgateway URL/portal-server-URL/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onChange="Check
(\qgateway URL/portal-server-URL/focus.html\q,\qfocus\q);return;">
<input TYPE=TEXT SIZE=20 value=focus onClick="Check
(\qgateway URL/portal-server-URL/focus.html\q,\qblur\q);return;">
```

// 在本示例中所有语句均会被重写，并且在每种情况下都会在前面加上网关 URL 和 Portal Server URL。这是因为在 `default_gateway_ruleset` 文件中定义了 `onAbort`、`onBlur`、`onFocus`、`onChange` 和 `onClick` 的相应规则。重写器会检测 JavaScript 标志，并将其传递给 JavaScript 函数规则以便做进一步处理。示例中所列的第二项规则会通知重写器要重写哪个参数。

```
</body>
<br>
```

```
Rewriting ends

</html>
```

HTML 表单示例

▼ 使用表单示例

1. 从以下位置访问此示例：
   portal-server-URL/rewriter/HTML/forms/formrule.html
2. 确保在网关服务的“域和子域的代理”列表中定义了 `abc.sesta.com`。
   如果没有定义该项，则假定采用直接连接，不会在其前面加网关 URL。
3. 将本示例中指定的规则添加到“HTML 属性重写规则”一节的 `default_gateway_ruleset` 中。
4. 在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的 `default_gateway_ruleset`。
5. 从终端窗口中重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
</head>
<body>
RW_START
<p>
<form name="form1" method="Post" action=
"http://abc.sesta.com/casestudy/html/form.html">
<input type="hidden" name="name1" value="0|1234|/test.html">
<input type="hidden" name="name3" value="../../html/test.html">
<form name="form2" method="Post" action="
http://abc.sesta.com/testcases/html/form.html"><br>
<input type="hidden" name="name1" value="0|1234|
../../html/test.html"></form>
RW_END </p>
</body>
</html>
```

规则

```
<Form source="*" name="form1" field="name1" valuePatterns="0|1234|"/>
```

重写后的 HTML 页

```
<HTML>
<HEAD>
RW_START
</HEAD>
<BODY>
<P>
<FORM name=form1  method=POST action="gateway-URL/http://abc.sesta.com/casestudy/html/form.html">
```

// 由于在 default_gateway_ruleset 中将 <Attribute name="action"/> 定义为 HTML 规则的一部分，所以会重写这个 URL。由于此 URL 已是绝对的，因此只需在其前面加上网关 URL。确保在网关服务的“域和子域的代理”列表中定义了 abc.sesta.com。否则，不会在其前面加网关 URL，因为此时将假定采用直接连接。

```
<input type=hidden name=name1 value=
"0|1234|gateway URL/portal-server-URL/test.html">
```

// 这里，表单名是 form1，字段名是 name1。这与规则中指定的表单名和字段名相匹配。规则将 valuePatterns 规定为 0|1234|，该值与本语句中的 value 相匹配。因此，会重写 valuePattern 后出现的 URL。在其前面加上 Portal Server URL 和网关 URL。有关 valuePatterns 的详细信息，参见第 71 页中的 “在规则中使用模式匹配”。

```
<input type=hidden name=name3 value="../../html/test.html">
```

// 由于 name 不匹配规则中指定的 field 名称，所以不会重写这个 URL。

```
</FORM>
<FORM name=form2 method=POST action=
"gateway-URL/http://abc.sesta.com/casestudy/html/form.html"><BR>
```

// 由于在默认规则集中将 <Attribute name="action"/> 定义为 HTML 规则的一部分，所以会重写这个 URL。由于此 URL 已是绝对的，因此只需在其前面加上网关 URL。

```
<input type=hidden name=name1 value="0|1234|../../html/test.html">
```

// 由于表单名不匹配规则中指定的名称，所以不会重写这个 URL。

```
</FORM>
</BODY>
RW_END
</HTML>
```

HTML Applet 示例

▼ 使用 Applet 示例

1. **获得 applet 类文件。** `RewriteURLinApplet.class` 文件位于以下位置：

   portal-server-URL/rewriter/HTML/applet/appletcode

   applet 代码所在页的基 URL 是：

   portal-server-URL/rewriter/HTML/applet/rule1.html

2. **将本示例中指定的规则添加到“ HTML 属性重写规则 ”一节的** `default_gateway_ruleset` **中。**

3. **在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的** `default_gateway_ruleset`。

4. **重新启动网关：**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML

```
<html>
Rewriting starts
<br>
<applet codebase=appletcode code=RewriteURLinApplet.class archive=/test>
<param name=Test1 value="/index.html">
<param name=Test2 value="../index.html">
<param name=Test3 value="../../index.html">
</applet>
Rewriting ends
</html>
```

规则

```
<Applet source="*/rule1.html" code="RewriteURLinApplet.class" param="Test*" />
```

重写后的 HTML

```
<HTML>
Rewriting starts
<BR>
<APPLET codebase=gateway-URL/portal-server-URL
/rewriter/HTML/applet/appletcode=RewriteURLinApplet.class archive=/test>
```

// 由于规则 <Attribute name="codebase"/> 已呈现为 default_gateway_ruleset 文件的一部分，所以会重写这个 URL。网关和 Portal Server URL 会连同到 appletcode 目录的路径一起被加在其前面。

```
<param name=Test1 value=
"gateway-URL/portal-server-URL/index.html">
```

// 由于页的基 URL 是 rule1.html，并且参数名与规则中指定的参数 Test* 相匹配，所以会重写此 URL。由于 index.html 被指定位于根层级，因此会直接在其前面加上网关 URL 和 Portal Server URL。

```
<param name=Test2 value="gateway-URL
/portal-server-URL/rewriter/HTML/index.html">
```

// 由于页的基 URL 是 rule1.html，并且参数名与规则中指定的参数 Test* 相匹配，所以会重写此 URL。会根据需要在其前面加上相应的路径。

```
<param name=Test3 value="gateway-URL
/portal-server-URL/rewriter/index.html">
```

// 由于页的基 URL 是 rule1.html，并且参数名与规则中指定的参数 Test* 相匹配，所以会重写此 URL。会根据需要在其前面加上相应的路径。

```
</APPLET>
Rewriting ends
</HTML>
```

JavaScript 内容示例

JavaScript URL 变量示例

▼ 使用 JavaScript URL 变量示例

1. **可从以下位置访问本示例：**
   portal-server-URL /rewriter/JavaScript/variables/url/js_urls.html
2. **确保在网关服务的“域和子域的代理”列表中定义了 abc.sesta.com。**
   如果没有定义该项，则假定采用直接连接，不会在其前面加网关 URL。
3. **将本示例中指定的规则添加到“JavaScript 源重写规则”一节的 default_gateway_ruleset 中。**
4. **在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的 default_gateway_ruleset。**

5 如果添加了此规则，请重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<html>
Rewriting starts
<head>
<title>JavaScript Variable test page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//URL Variables
var imgsrc="/tmp/tmp.jpg";
var imgsrc="./tmp/tmp.jpg";
var imgsrc="../tmp/tmp.jpg";
var imgsrc="../../tmp/tmp.jpg";
var imgsrc="http://abc.sesta.com/tmp/tmp.jpg";
var imgsrc="../../../tmp/tmp.jpg";
var imgsrc="tmp/tmp.jpg";
//-->
</SCRIPT>
<br>
Testing JavaScript variables!
<br>
<img src="images/logo.gif">
<br>
Image
</body>
<br>
Rewriting ends
</html>
```

规则

```
<Variable name="imgsrc" type="URL"/>
```

重写后的 HTML 页

```
<html>
Rewriting starts
<head>
<title>JavaScript Variable test page</title>
</head>
<body>
<script LANGUAGE="Javascript">
```

```
<!--
//URL Variables
var imgsrc="gateway-URL/portal-server-URL/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL
/rewriter/JavaScript/variables/url/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL
/rewriter/JavaScript/variables/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL
/rewriter/JavaScript/tmp/tmp.jpg";
var imgsrc="gateway-URL/http://abc.sesta.com/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL/rewriter/tmp/tmp.jpg";
var imgsrc="gateway-URL/portal-server-URL
/rewriter/JavaScript/variables/url/tmp/tmp.jpg";
```

// 如规则中指定的那样，所有上述 URL 都是 URL 类型且名称为 `imgsrc` 的 JavaScript 变量。因此会在它们前面加上网关 URL 和 Portal Server URL。根据需要，会在其前面加上跟在 Portal Server URL 后面的路径。

```
//-->
</SCRIPT>
<br>
Testing JavaScript variables!
<br>
<img src="gateway URL/portal-server-URL/rewriter
/JavaScript/variables/url/images/logo.gif">
```

// 由于在 `default_gateway_ruleset` 中定义了规则 `<Attribute name="src"/>`，所以会重写该行

```
<br>
Image
</body>
<br>
Rewriting ends
</html>
```

JavaScript EXPRESSION 变量示例

▼ 使用 JavaScript 表达式变量示例

1. 可从以下位置访问本示例：
   portal-server-URL /rewriter/JavaScript/variables/expr/expr.html
2. 将本示例中指定的规则（如果尚不存在）添加到“ JavaScript 源重写规则 ”一节的 `default_gateway_ruleset` 中。

3 在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的 default_gateway_ruleset。

4 如果添加了此规则，请重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<html>
<head>
<title>JavaScript EXPRESSION Variables Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//Expression variables
var expvar1="images";
var expvar2="/logo.gif";
var expvar = expvar1 + expvar2;
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
var expvar="/images/logo"+".gif";
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
//-->
</SCRIPT>
Testing JavaScript EXPRESSION variables
</body>
</html>
```

规则

```
<Variable type="EXPRESSION" name="expvar"/>
```

重写后的 HTML 页

```
<html>
<head>
<title>JavaScript EXPRESSION Variables Test Page</title>
</head>
<body>
<SCRIPT>
// Rewriter appends the wrapper function
psSRAPRewriter_convert_expression here
</SCRIPT>
<script LANGUAGE="Javascript">
<!--
//Expression variables
var expvar1="images";
```

```
var expvar2="/logo.gif";
var expvar =psSRAPRewriter_convert_expression( expvar1 + expvar2);
```

// 重写器会将该语句的右侧部分识别为 JavaScript EXPRESSION 变量。重写器不能在服务器端求解该表达式的值。因此，会在此表达式前面加上函数 psSRAPRewriter_convert_expression。在客户机端对此表达式进行求值，并根据需要对其进行重写。

```
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
```

// 使用了上一语句中 expvar 重写后的值来得出该表达式的值。由于结果是一个有效的 URL（在示例中，该位置有图形存在），因此链接将会起作用。

```
var expvar="gateway URL/portal-server-URL/images/logo"+".gif";
```

// 重写器会将 expvar 的右侧部分识别为字符串表达式。该表达式可以在服务器一方求解，因而会直接对其进行重写。

```
document.write("<A HREF="+expvar+">EXPRESSION</A><P>")
```

// 使用了上一语句中 expvar 重写后的值来得出该表达式的值。由于结果不是一个有效的 URL（在最终得出的位置不存在图形），因此链接将不起作用。

```
//-->
</SCRIPT>
Testing JavaScript EXPRESSION variables
</body>
</html>
```

JavaScript DHTML 变量示例

▼ 使用 JavaScript DHTML 变量示例

1. 可从以下位置访问本示例：

   portal-server-URL /rewriter/JavaScript/variables/dhtml/dhtml.html

2. 确保在网关服务的“域和子域的代理”列表中定义了 abc.sesta.com。如果没有定义该项，则假定采用直接连接，不会在其前面加网关 URL。

3. 将本示例中指定的规则（如果尚不存在）添加到“JavaScript 源重写规则”一节的 default_gateway_ruleset 中。在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的 default_gateway_ruleset。

4. 如果添加了此规则，请重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<html>
<head>
<title>JavaScript DHTML Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a href=../../images/test.html>"
var dhtmlVar="<a href=/../images/test.html>"
var dhtmlVar="<a href=/images/test.html>"
var dhtmlVar="<a href=images/test.html>"
var dhtmlVar="<a href=http://abc.sesta.com/images/test.html>"
var dhtmlVar="<img src=http://abc.sesta.com/images/test.html>"
//-->
</SCRIPT>
<br><br>
Testing DHTML Variables
<br><br>
<img src="images/logo.gif">IMAGE
</body>
</html>
```

规则

```
<Variable name="DHTML">dhtmlVar</Variable>
```

重写后的 HTML 页

```
<html>
<head>
<title>JavaScript DHTML Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//DHTML Var
var dhtmlVar="<a href=gateway-URL/portal-server-URL
/rewriter/JavaScript/images/test.html>"
```

// JavaScript DHTML 规则将 `dhtmlVar` 的右侧部分确定为动态 HTML 内容。因而，会应用 `default_gateway_ruleset` 文件中的 HTML 规则。动态 HTML 包含 `href` 属性。`default_gateway_ruleset` 定义了规则 `<Attribute name="href"/>`。因此，会重写 `href` 属性的值。但此 URL 不是绝对的；因此，会用页的基 URL 以及所需子目录来替换这个相对的 URL。接着在其前面加上网关 URL 以得出最终重写后的输出。

```
var dhtmlVar="<a href=gateway-URL
/portal-server-URL/../images/test.html>"
```

// 虽然附加了页的基 URL，并且在前面加上了网关 URL，但最终得到的 URL 不会起作用。这是因为初始 URL `/../images/test.html` 是错误的。

```
var dhtmlVar="<a href=gateway-URL
/portal-server-URL/images/test.html>"
```

// 这里，JavaScript DHTML 规则同样将右侧部分确定为动态 HTML 内容，并将其传递给 HTML 规则。应用 `default_gateway_ruleset` 中的 HTML 规则 `<Attribute name="href"/>`，并按所示方式重写该语句。在其前面加上网关 URL 和 Portal Server URL。

```
var dhtmlVar="<a href=gateway URL/portal-server-URL/
rewriter/JavaScript/variables/dhtml/images/test.html>"
var dhtmlVar="<a href=gateway URL/http://abc.sesta.com/images/test.html>"
var dhtmlVar="<img src=gateway-URL/
http://abc.sesta.com/images/test.html>"
```

// JavaScript DHTML 规则会确定出右侧的动态 HTML 内容，并将此语句传递给 HTML 规则。此时会应用 `default_gateway_ruleset` 中的 `<Attribute name="src"/>` 规则。由于此 URL 是绝对的，因此只需在其前面加上网关 URL。为重写该 URL，请确保在“域和子域的代理”列表中定义了 `abc.sesta.com`。

```
//-->
</SCRIPT>
<br><br>
Testing DHTML Variables
<br><br>
<img src="gateway-URL/portal-server-URL/
rewriter/JavaScript/variables/dhtml/images/logo.gif">
```

// 由于在 `default_gateway_ruleset` 中定义了规则 `<Attribute name="src"/>`，所以会重写该行。

```
<br><br>
Image
</body>
</html>
```

JavaScript DJS 变量示例

▼ 使用 JavaScript DJS 变量示例

1 **可从以下位置访问本示例：**

portal-server-URL /rewriter/JavaScript/variables/djs/djs.html

2. 确保在网关服务的“域和子域的代理”列表中定义了 abc.sesta.com。如果没有定义该项，则假定采用直接连接，不会在其前面加网关 URL。

3. 将本示例中指定的两项规则（如果尚不存在）添加到“JavaScript 源重写规则”一节的 default_gateway_ruleset 中。在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的 default_gateway_ruleset。

4. 重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<html>
<head>
<title>Dynamic JavaScript Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
var dJSVar="var dJSimgsrc=\q/tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\q../../../tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\qhttp://abc.sesta.com/tmp/tmp/jpg\q;"
//-->
</SCRIPT>
<br>
Testing Dynamic JavaScript Variables
<br>
<img src="images/logo.gif">
<br>
Image
</body>
</html>
```

规则

```
<Variable name="dJSVar" type="DJS"/>
<Variable name="dJSimgsrc" type="URL"/>
```

重写后的 HTML 页

```
<html>
<head>
<title>Dynamic JavaScript Variable Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
```

```
var dJSVar="var dJSimgsrc=\qgateway-URL
/portal-server-URL/tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\qgateway-URL
/portal-server-URL/rewriter/tmp/tmp/jpg\q;"
var dJSVar="var dJSimgsrc=\qgateway-URL
/http://abc.sesta.com/tmp/tmp/jpg\q;"
```

// 会用网关 URL 和 Portal Server URL 重写上面的所有语句。还会适当地在前面加上所需的路径。第一项规则将 `dJSVar` 的右侧部分确定为一个动态 JavaScript 变量。然后将其传递给第二项规则，后者将 `dJSimgsrc` 的右侧部分确定为 URL 类型的 JavaScript 变量。并且会相应地对其进行重写。

```
//-->
</SCRIPT>
<br>
Testing Dynamic JavaScript Variables
<br>
<img src="gateway-URL/portal-server-URL
/rewriter/JavaScript/variables/djs/images/logo.gif">
```

// 由于在 `default_gateway_ruleset` 中定义了规则 `<Attribute name="src"/>`，所以会重写该行。

```
<br>
Image
</body>
</html>
```

JavaScript SYSTEM 变量示例

▼ 使用 JavaScript 系统变量示例

1. 可从以下位置访问本示例：
   portal-server-URL /rewriter/JavaScript/variables/system/system.html
2. 将本示例中指定的规则（如果尚不存在）添加到“JavaScript 源重写规则”一节的 `default_gateway_ruleset` 中。
3. 在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的 `default_gateway_ruleset`。
4. 重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<html>
<head>
<title>JavaScript SYSTEM Variables Test Page</title>
</head>
<body>
<script LANGUAGE="Javascript">
<!--
//SYSTEM Var
alert(window.location.pathname);
//document.write
("<A HREF="+window.location.pathname+">SYSTEM</A><P>")
//-->
</SCRIPT>
Testing JavaScript SYSTEM Variables
<br>
This page displays the path where
the current page is located when loaded.
</body>
</html>
```

规则

```
<Variable name="window.location.pathname" type="SYSTEM"/>
```

重写后的 HTML

```
<html>
<head>
<title>JavaScript SYSTEM Variables Test Page</title>
</head>
<body>
<SCRIPT>
convertsystem function definition...
</SCRIPT>
<script LANGUAGE="Javascript">
<!--
//SYSTEM Var
alert(psSRAPRewriter_convert_system
(window.location, window.location.pathname, "window.location"));
```

// 重写器将 `window.location.pathname` 确定为 JavaScript SYSTEM 变量。无法在服务器端确定该变量的值。因此，重写器会在此变量前加上 `psSRAPRewriter_convert_pathname` 函数。这个包裹函数将在客户机端确定变量的值，并根据需要进行重写。

```
//-->
</SCRIPT>
Testing JavaScript SYSTEM Variables
<br>
This page displays the path where
the current page is located when loaded.
</body>
</html>
```

JavaScript URL 函数示例

▼ 使用 JavaScript URL 函数示例

1. 可从以下位置访问本示例：

   portal-server-URL /rewriter/JavaScript/functions/url/url.html

2. 将本示例中指定的规则（如果尚不存在）添加到“ **JavaScript** 源重写规则 ”一节的 `default_gateway_ruleset` 中。在 **Portal Server** 管理控制台中，编辑 **Portal Server** 配置下的重写器服务中的 `default_gateway_ruleset`。

3. 重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<html>
<body>
JavaScript URL Function Test Page
<br>
<script language="JavaScript">
<!--
function test(one,two,three)
{
alert(one + "##" + two + "##" +three);
}
test("/test.html","../test.html","123");
window.open("/index.html","gen",width=500,height=500);
//-->
</SCRIPT>
</body>
</html>
```

规则

```
<Function type="URL" name="test" paramPatterns="y,y"/>
<Function type="URL" name="window.open" paramPatterns="y"/>
```

重写后的 HTML 页

```
<html>
<body>
JavaScript URL Function Test Page
<br>
<script language="JavaScript">
<!--
function test(one,two,three)
{
alert(one + "##" + two + "##" +three);
}
test("/test.html","../test.html","123");
window.open("gateway-URL/portal-server-URL
/index.html","gen",width=500,height=500);
//-->
</SCRIPT>
</body>
</html>
```

JavaScript EXPRESSION 函数示例

▼ 使用 JavaScript 表达式函数示例

1. 可从以下位置访问本示例：

   ```
   <portal-install-location>/SUNWportal/samples/rewriter
   ```

2. 将本示例中指定的规则（如果尚不存在）添加到“ JavaScript 源重写规则 ”一节的 default_gateway_ruleset 中。

3. 使用 Portal Server 管理控制台编辑重写器服务中的 default_gateway_ruleset。

4. 重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<html>
<body>
JavaScript EXPRESSION Function Test Page
<br><br><br>
<script language="JavaScript">
<!--
function jstest2()
{
```

```
return ".html";
}
function jstest1(one)
{
return one;
}
var dir="/images/test"
var test1=jstest1(dir+"/test"+jstest2());
document.write("<a HREF="+test1+">Test</a>");
alert(test1);
//-->
</SCRIPT>
</body>
</html>
```

规则

```
<Function type="EXPRESSION" name="jstest1" paramPatterns="y"/>
```

重写后的 HTML 页

```
<html>
<body>
JavaScript EXPRESSION Function Test Page
<br><br><br>
<script>
<!--
// various functions including psSRAPRewriter_
convert_expression appear here.//-->
</SCRIPT>
<script language="JavaScript">
<!--
function jstest2()
{
return ".html";
}
function jstest1(one)
{
return one;
}
var dir="/images/test"
var test1=jstest1(psSRAPRewriter_convert_
expression(dir+"/test"+jstest2()));
```

// 此规则规定需要重写函数 `jstest1` 中类型为 EXPRESSION 的第一个参数。该表达式的值是 `/test/images/test.html`。会在该值前面加上 Portal Server URL 和“网关 URL”。

```
document.write("<a HREF="+test1+">Test</a>");
alert(test1);
//-->
</SCRIPT>
</body>
</html>
```

JavaScript DHTML 函数示例

▼ 使用 JavaScript DHTML 函数示例

1. 可从以下位置访问本示例：
   portal-server-URL /rewriter/JavaScript/functions/dhtml/dhtml.html
2. 将本示例中指定的规则（如果尚不存在）添加到“ **JavaScript** 源重写规则 ”一节的 `default_gateway_ruleset` 中。
3. 在 **Portal Server** 管理控制台中，编辑 **Portal Server** 配置下的重写器服务中的 `default_gateway_ruleset`。
4. 重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<html>
<head>
Testing JavaScript DHTML Functions
<br>
<br>
<script>
<!--
document.write(\q<a href="/index.html">write</a><BR>\q)
document.writeln(\q<a href="index.html">writeln</a><BR>\q)
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
//-->
</SCRIPT>
</head>
<body BGCOLOR=white>
<br><br>
Testing document.write and document.writeln
</body>
</html>
```

规则

```
<Function type="DHTML" name=" document.write" paramPatterns="y"/>
<Function type="DHTML" name=" document.writeln" paramPatterns="y"/>
```

重写后的 HTML 页

```
<html>
<head>
Testing JavaScript DHTML Functions
<br>
<br>
<script>
<!--
document.write(\q<a href="gateway-URL
/portal-server-URL/index.html">write</a><BR>\q)
```

// 第一项规则指定需要重写 DHTML JavaScript 函数 `document.write` 的第一个参数。重写器将第一个参数确定为一个简单 HTML 语句。`default_gateway_ruleset` 中的 HTML 规则部分具有规则 `<Attribute name="href" />`，该规则指示需要重写该语句。

```
document.writeln(\q<a href="gateway-URL
/portal-server-URL/rewriter/JavaScript/functions/dhtml/index.html">writeln</a><BR>\q)
```

// 第二项规则指定需要重写 DHTML JavaScript 函数 `document.writeln` 的第一个参数。重写器将第一个参数确定为一个简单 HTML 语句。`default_gateway_ruleset` 中的 HTML 规则部分具有规则 `<Attribute name="href" />`，该规则指示需要重写该语句。

```
document.write("http://abc.sesta.com/index.html<BR>")
document.writeln("http://abc.sesta.com/index.html<BR>")
```

// 虽然此 DHTML 规则确定出了 `document.write` 和 `document.writeln` 函数，但是不会重写上述语句。这是因为本例中的第一个参数不是简单 HTML。它可以是任意的字符串，因而重写器不知道该如何重写这个参数。

```
//-->
</SCRIPT>
</head>
<body BGCOLOR=white>
<br><br>
Testing document.write and document.writeln
</body>
</html>
```

JavaScript DJS 函数示例

▼ 使用 JavaScript DJS 函数示例

1. 可从以下位置访问本示例：
   portal-server-URL /rewriter/JavaScript/functions/djs/djs.html

2. 确保在网关服务的“域和子域的代理”列表中定义了 `abc.sesta.com`。
   如果没有定义该项，则假定采用直接连接，不会在其前面加网关 URL。

3. 将本示例中指定的规则（如果尚不存在）添加到“JavaScript 源重写规则”一节的 `default_gateway_ruleset` 中。在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的 `default_gateway_ruleset`。

4. 重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 HTML 页

```
<html>
Test for JavaScript DJS Functions
<br>
<script>
menu.addItem(new NavBarMenuItem("All Available
Information","JavaScript:top.location=\qhttp://abc.sesta.com\q"));
//menu.addItem(new NavBarMenuItem("All Available Information","http://abc.sesta.com"));
</script>
</html>
```

规则

```
<Function type="DJS" name="NavBarMenuItem" paramPatterns=",y"/>
<Variable type="URL" name="top.location"/>
```

重写后的 HTML 页

```
<html>
Testing JavaScript DJS Functions
<br>
<script>
menu.addItem(new NavBarMenuItem
("All Available Information","javaScript:top.location=
\qgateway-URL/http://abc.sesta.com\q"));
```

// abc.sesta.com 是网关服务的“域和子域的代理”列表中的一项。因此重写器需要重写这个 URL。但由于是一个绝对 URL，所以不需要在其前面加 Portal Server URL。此 DJS 规则规定需要重写 DJS 函数 NavBarMenuItem 的第二个参数。但第二个参数还是一个 JavaScript 变量。此时还需要第二项规则来重写该变量的值。第二项规则指定需要重写 JavaScript 变量 top.location 的值。由于满足上述所有条件，所以会重写此 URL。

```
//menu.addItem(new NavBarMenuItem("All Available Information","http://abc.sesta.com"));
```

// 虽然此 DJS 规则指定需要重写函数 NavBarMenuItem 的第二个参数，但在本语句中不会发生这种情况。这是因为重写器不会将第二个参数识别为简单 HTML。

```
</script>
</html>
```

XML 属性示例

▼ 使用 XML 属性示例

1. 可从以下位置访问本示例：

   portal-server-URL /rewriter/XML/attrib.html

2. 将本示例中指定的规则（如果尚不存在）添加到“ XML 源重写规则 ”一节的 default_gateway_ruleset 中。

3. 在 Portal Server 管理控制台中，编辑 Portal Server 配置下的重写器服务中的 default_gateway_ruleset。

4. 重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

重写前的 XML

```
<html>
RW_START
<body>
<xml>
<baseroot href="/root.html"/>
</xml>
<xml>
<img href="image.html"/>
</xml>
<xml>
<string href="1234|substring.html"/>
```

```
</xml>
<xml>
<check href="1234|string.html"/>
</xml>
</body>
RW_END
</html>
```

规则

```
<Attribute name="href" tag="check" valuePatterns="1234|"/>
```

重写后的 HTML

```
<html>
Rewriting starts
<br>
<br>
<body>
<xml><baseroot href="/root.html"/></xml>
<xml><img href="image.html"/></xml>
<xml><string href="1234|substring.html"/></xml>
<xml><check href="1234|gateway-URL/portal-server-URL
/rewriter/XML/string.html"/></xml>
```

// 由于本语句符合规则中指定的条件，所以会重写它。`Attribute name` 是 `href`， `tag` 是 `check` 而 `valuePatterns` 是 `1234`。将会重写 `valuePatterns` 后的字符串。有关 `valuePatterns` 的详细信息，参见第 71 页中的 “在规则中使用模式匹配”。

```
</body>
Rewriting ends
</html>
```

实例研究

本节包括一个邮件客户机示例的源 HTML 页。本实例研究并未涵盖所有可能的方案和规则。它只是一个规则集示例，目的是为了帮助您将自己内联网页的相应规则合在一起。

假设

本实例研究作了如下假设：

- 假定邮件客户机的基 URL 为 `abc.siroe.com`
- 假定网关 URL 为 `gateway.sesta.com`

- 假定网关服务的“域和子域代理”列表中有相关条目

示例页 1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0053)http://abc.siroe.com/mailclient/destin/?Cmd=navbar -->
<HTML XMLNS:WM><HEAD>
<META http-equiv=Content-Type content="text/html; CHARSET=utf-8">
<META http-equiv=Pragma content=no-cache>
<META http-equiv=Expires content=0><!--Copyright (c) 2000 Microsoft Corporation.
All rights reserved.--><!--CURRENT FILE== "IE5" "WIN32" navbar -->
<STYLE>WM\\:DROPMENU {
BEHAVIOR: url(http://abc.siroe.com/mailweb/controls/dropmenu.htc)
}
</STYLE>
<LINK href="destin_files/navbar.css" type=text/css rel=stylesheet>
<SCRIPT language=javascript>
var g_szUserBase= "http://abc.siroe.com/mailclient/destin"+"/";
var g_szFolder= ".";
var g_szVirtualRoot= "http://abc.siroe.com/mailweb";
var g_szImagePath= g_szVirtualRoot + "/img/";
</SCRIPT>
<SCRIPT src="/destin_files/navbar.js"></SCRIPT>
<META content="MSHTML 6.00.2600.0" name=GENERATOR></HEAD>
<BODY oncontextmenu=return(event.ctrlKey); onselectstart=return(false);
id=outbar_mainbody style="BACKGROUND-COLOR: appworkspace" leftMargin=0
topMargin=0 scroll=no>
<TABLE class=nbTableMain id=nbTableMain style="HEIGHT: 100%" cellSpacing=0
cols=1 cellPadding=0 rows="2">
<TBODY>
<TR>
<TD class=treeBrand>
<DIV class=treeOFLOW><IMG
style="PADDING-RIGHT: 0px; PADDING-LEFT: 0px; PADDING-BOTTOM: 0px; PADDING-TOP: 0px"
src="/destin_files/logo-ie5.gif" border=0></DIV></TD></TR>
<TR height="100%">
<TD>
<TABLE class=nbTable cellSpacing=0 cols=1 cellPadding=0 rows="4">
<TBODY>
<TR>
<TD class=nbFlybar id=show_navbar onkeydown=flybar_keydown()
onclick=ToggleTab(this.id) tabIndex=0 noWrap>
<DIV class=treeOFLOW>Shortcuts</DIV></TD></TR>
<TR style="HEIGHT: 100%">
<TD id=idOutbarpane style="TEXT-ALIGN: center" vAlign=top><A
id=inbox
href="http://abc.siroe.com/mailclient/destin/Inbox/?Cmd=contents&amp;
```

```
Page=1"
target=viewer alt="Go to inbox"><IMG class=nbImage alt="Go to inbox"
src="destin_files/navbar-inbox.gif"></A>
<DIV class=nbLabel>Inbox</DIV><BR><A id=calendar
href="http://abc.siroe.com/mailclient/destin/Calendar/?Cmd=contents"
target=viewer alt="Go to calendar"><IMG class=nbImage
alt="Go to calendar" src="destin_files/navbar-calendar.gif"></A>
<DIV class=nbLabel>Calendar</DIV><BR><A id=contacts
            href="http://abc.siroe.com/mailclient/destin/Contacts/?Cmd=contents"
target=viewer alt="Go to contacts"><IMG class=nbImage
alt="Go to contacts" src="destin_files/navbar-contacts.gif"></A>
<DIV class=nbLabel>Contacts</DIV><BR><A id=options
href="http://abc.siroe.com/mailclient/destin/?Cmd=options"
target=viewer alt="Go to options"><IMG class=nbImage
alt="Go to options" src="destin_files/navbar-options.gif"></A>
<DIV class=nbLabel>Options</DIV></TD></TR>
<TR style="HEIGHT: 1.5em">
<TD class=nbFlybar id=show_folders onkeydown=flybar_keydown()
onclick=ToggleTab(this.id) tabIndex=0 noWrap>
<DIV class=treeOFLOW>Folders</DIV></TD></TR>
<TR>
<TD class=nbTreeProgress id=treeProgress style="DISPLAY: none"
vAlign=top noWrap><SPAN id=idLoading
style="OVERFLOW: hidden">Loading...</SPAN>
</TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
</BODY></HTML>
```

描述

第 118 页中的 “描述” 展示了示例规则集与实例研究之间的映射。

表 4–3 示例规则集与实例研究之间的映射

页内容	应用的规则	重写器输出	描述
`var g_szVirtualRoot= "http://abc.siroe.com /mailweb";`	<Variable name="URL"> g_szVirtualRoot </Variable>	`var g_szVirtualRoot= "http://gateway. sesta.com /http://abc.siroe.com /mailweb";`	`g_szVirtualRoot` 是一个值为简单 URL 的变量。 该规则通知重写器搜索 URL 类型的变量 `g_szVirtualRoot`。如果网页中存在这样的变量，重写器会将其转换成绝对 URL，并在其前加上网关 URL。
`src="/destin_files/ logo-ie5.gif"`	<Attribute name="src" />	`src="http://gateway.sesta. com/ http://abc.siroe.com/ destin_files/logo-ie5.gif`	src 是属性的名称，它没有附带任何标记或 valuePattern。 该规则通知重写器搜索所有具有名称 `src` 的属性，并重写该属性的值。

表 4–3 示例规则集与实例研究之间的映射 （续）

页内容	应用的规则	重写器输出	描述
href="http://abc.siroe. com /mailclient/destin/Inbox/ ?Cmd=contents&Page=1"	<Attribute name="href"/>	href="http://gateway.sesta com/ http://abc.siroe.com /mailclient/destin/ Inbox/?Cmd=contents& Page=1"	href 是属性的名称，它没有附带任何标记或 valuePattern。 该规则通知重写器搜索所有具有名称 href 的属性，并重写该属性的值。

注 – 规则集应用优先顺序为 hostname-subdomain-domain。

例如，假定在基于域的规则集列表中有下列条目：

```
sesta.com|ruleset1
eng.sesta.com|ruleset2
host1.eng.sesta.com|ruleset3
```

ruleset3 将应用于 host1 上的所有页。

除了从 host1 中检索到的页之外，ruleset2 将应用于 eng 子域中的所有页。

除了从 eng 子域和 host1 中检索到的页之外，ruleset1 将应用于 sesta.com 域中的所有页。

1. 单击“保存”以完成修改。
2. 从终端窗口中重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

Outlook Web Access 规则集

Secure Remote Access 服务器支持 Sun Java System Web Server 及 IBM 应用服务器上 Outlook Web Access (OWA) 的 MS Exchange 2000 SP3 安装和 MS Exchange 2003 安装。

▼ 配置 OWA 规则集

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“ Secure Remote Access ”选项卡，然后选择想要为其设置属性的网关配置文件。

3 **在“将 URI 映射至规则集”字段中，输入安装了 Exchange 2000 的服务器名称，紧接着为 Exchange 2000 Service Pack 4 OWA 规则集。**

例如：

```
exchange.domain.com|exchange_2000sp3_owa_ruleset。
```

使用公共文件夹

在 Exchange 端，“公共文件夹”被配置为使用 NTLM 验证。在此，需要将其更改为使用 HTTP 基本验证。

要执行此操作，转到 Exchange 服务器并选择“控制面板”-->“管理工具”，然后打开“Internet 信息服务”。

在“默认 Web 站点”下，有一个名为“公共”的选项卡，用于设置公共文件夹。单击鼠标右键并选择“属性”。单击“目录安全”选项卡。在“匿名访问和验证”控制面板中选择“编辑..”。取消选中除“基本验证”之外的所有其他选项。

6.x 与 3.0 的规则集映射

下表列出了 Secure Remote Access 服务器重写器规则与 Portal Server 产品先前版本的映射。

表 4–4 与 SP3 的规则映射

Rewriter 6.0 DTD 元素	Rewriter 3.0 列表框名称
HTML 内容规则	
属性 - URL	重写 HTML 属性
属性 - DJS	重写包含 JavaScript 的 HTML 属性
表单	重写表输入标记列表
Applet	重写 Applet/Object 参数值列表
JavaScript 内容规则	
变量 - URL	重写 URL 中的 JavaScript 变量
变量 - EXPRESSION	重写 JavaScript 变量函数
变量 - DHTML	重写 HTML 中的 JavaScript 变量
变量 - DJS	重写 JavaScript 中的 JavaScript 变量
变量 - SYSTEM	重写 JavaScript 系统变量

表 4–4 与 SP3 的规则映射 （续）

Rewriter 6.0 DTD 元素	Rewriter 3.0 列表框名称
函数 - URL	重写 JavaScript 函数参数
函数 - EXPRESSION	重写 JavaScript 函数参数功能
函数 - DHTML	重写 HTML 中的 JavaScript 函数参数
函数 - DJS	重写 JavaScript 中的 JavaScript 函数参数
XML 内容规则	
属性 - URL	重写 XML 文档的属性值
TagText	重写XMl 文档的文本数据
CSS 内容规则	
无需任何规则。默认情况下，会转换所有 URL	
WML 内容规则	
未定义任何规则。WML 以 HTML 方式处理，并应用 HTML 规则。	
WMLScript 内容规则	
不支持 WML 脚本	

◆ ◆ ◆ 第 5 章

使用 NetFile

本章说明 NetFile 及其操作。若要配置 NetFile，参见第 14 章。

NetFile 简介

NetFile 是一个文件管理器应用程序，它允许用户对远程文件系统和目录进行访问和操作。

Secure Remote Access 的 NetFile 组件以 Java2 applet 形式提供。Java2 applet 具有更加理想的界面，同时使访问更轻松。

NetFile 提供了以下关键功能：

- 添加或删除共享或文件夹的工具
- 文件上载与下载
- 搜索文件与文件夹
- 使用 GZIP 和 ZIP 压缩文件
- NetFile 环境中的邮件工具
- 保存当前的 NetFile 会话信息
- 拖放文件

支持的文件访问协议

NetFile 允许您使用 FTP、NFS 和 jCIFS (Microsoft Windows) 协议访问远程系统。它包括以下文件访问协议功能：

- 如果用户指定使用 “AUTODETECT” 添加系统，NetFile 会按照以下顺序自动检测要使用的协议：

- 检查端口 21 上 FTP 服务器的主机。如果 FTP 响应包含字符串 "NetWare"，则将其视为 NETWARE 主机。
- 检查端口 2049 上 NFS 服务器的主机。
- 检查端口 139 上 Microsoft Windows 的主机。
- 如果上述操作均失败，将显示消息：无法确定主机类型。

  检测到的第一个文件系统类型将被用于连接所请求的主机。可在 Portal Server 管理控制台 (PSConsole) 中更改主机检测顺序。

注 – 如果服务器正在非标准端口上运行，则连接会失败。

- NetFile 允许用户自行选择文件服务器和协议。

  针对这些协议中的每一项，下面列出了所支持的平台。

表 5–1 文件系统和支持的协议

文件系统/协议	平台
FTP	运行于 Novell Netware 上的 Novell FTP 5.1 Server 运行于 Win NT 4.0 上的 MS FTP Server 4.0 运行于 Win NT 2000 上的 MS FTP Server 5.0 Solaris FTP Server WU_FTP 2.6.1 ProFTPD 1.2.8 vsFTPd 1.2.0
NFS	Solaris 2.6 及更高版本
jCIFS	Windows 95/98/NT/2000/ME/XP

注 – 要使用 NetFile 将文件上载到 ProFTPD 服务器，需要在运行 ProFTPD 服务器的主机中将 `proftpd.conf` 文件中的“AllowStoreRestart”设置为“on”。

只有通过 FTP 服务器才支持 Novell Netware，不能通过本机访问支持。

要访问 Microsoft Windows (SMB/CIFS) 文件系统，必须在 Portal Server 上安装 jCIFS。jCIFS 是实现 CIFS/SMB 联网协议的“开放源代码”客户机库。

▼ 创建 NetFile 策略

1 以管理员身份登录到 Portal 管理控制台。

2 选择“ Secure Remote Access ”选项卡，然后选择“ NetFile ”选项卡。

3 从“ 选择 DN ”下拉框选择“ 组织 ”/“ 角色 ”/“ 用户 ”。

4 将权限设置为访问/拒绝主机和服务。

5 单击“ 保存 ”。

6 重新启动网关。

◆ ◆ ◆ 第 6 章

使用 Netlet

本章说明如何使用 Netlet 在用户远程桌面与内联网中运行应用程序的服务器之间安全地运行应用程序。若要配置 Netlet，参见第 11 章。

本章包含以下各节：

Netlet 简介

Sun Java System Portal Server 软件用户可能希望在其远程桌面上以安全方式运行一些流行的或公司专用的应用程序。通过在您的平台上设置 Netlet，可提供对这些应用程序的安全访问。

Netlet 使得用户可以在不安全的网络（如 Internet）上安全地运行常用 TCP/IP 服务。您可以运行 TCP/IP 应用程序（如 Telnet 和 SMTP）、HTTP 应用程序和任何使用固定端口的应用程序。

如果应用程序基于 TCP/IP 或使用固定端口，则可以在 Netlet 上运行该应用程序。

注 – 仅当使用 FTP 时才支持动态端口。要使用 Microsoft Exchange，请使用 OWA (Outlook Web Access)。

请确保通知您的用户在使用 Netlet 时，必须禁用其浏览器中的弹出窗口阻止程序选项。

Netlet 组件

Netlet 使用的各种组件如 第 128 页中的 “Netlet 组件” 中所示。

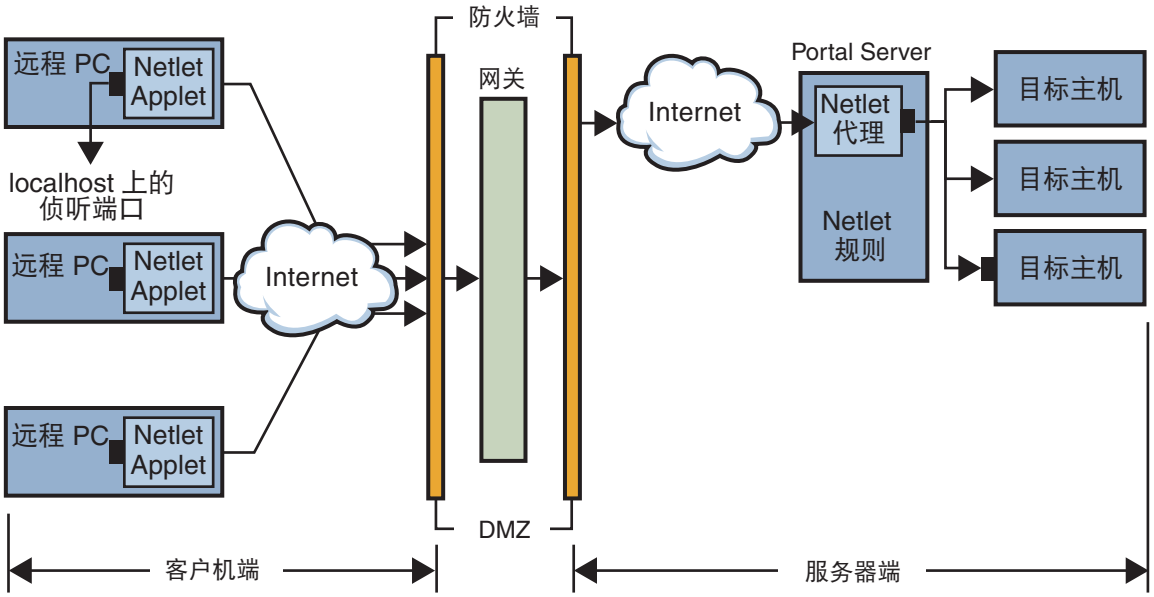


图 6–1 Netlet 组件

localhost 上的侦听端口

这是 Netlet applet 在其上进行侦听的客户机中的端口。客户机为 localhost。

Netlet Applet

Netlet applet 负责在远程客户机与内联网应用程序（如 Telnet、Graphon 或 Citrix）之间设置加密 TCP/IP 隧道。applet 将信息包加密，并将其发送至网关，然后解密来自网关的响应信息包，并将其发送至本地应用程序。

对于静态规则，当用户登录到门户时，将自动下载 Netlet applet。对于动态规则，当用户单击对应于动态规则的链接时，会下载 applet。有关静态和动态规则的详细信息，参见第 133 页中的 “规则类型” 。

要在 Sun Ray 环境下运行 Netlet，参见第 144 页中的 “在 Sun Ray 环境中运行 Netlet” 。

Netlet 规则

Netlet 规则会将需要在客户机上运行的应用程序映射到相应的目标主机中。这意味着 Netlet 仅适用于发送至在 Netlet 规则中所定义端口中的信息包。这将确保更高的安全性。

作为管理员，您需要为 Netlet 的运行配置特定规则。这些规则指定各种细节，如要使用的密码、要调用的 URL、要下载的 applet、目标端口以及目标主机。当客户机上的用户通过 Netlet 提出请求时，这些规则有助于确定建立连接必须采用的方式。有关详细信息，参见第 130 页中的 “定义 Netlet 规则”。

Netlet 提供者

这是 Netlet 的 UI 组件。提供者允许用户从 Portal Server 桌面配置所需的应用程序。提供者中创建了一个链接，用户单击此链接可运行所需的应用程序。用户也可在 Netlet 提供者的桌面上指定动态规则的目标主机。参见第 130 页中的 “定义 Netlet 规则”。

Netlet 代理（可选）

网关可确保远程客户机与网关之间的隧道安全。Netlet 代理是可选的，在安装期间可以选择不安装此代理。有关 Netlet 代理的信息，参见第 47 页中的 “使用 Netlet 代理”。

Netlet 使用方案

Netlet 的使用涉及下列一系列事件：

1. 远程用户登录至 Portal Server 桌面。
2. 如果已为用户、角色或组织定义了静态 Netlet 规则，则自动将 Netlet applet 下载到远程客户机中。

   如果已为用户、角色或组织定义了动态规则，则用户需要在 Netlet 提供者中配置所需的应用程序。当用户单击 Netlet 提供者中的应用程序链接时，将下载 Netlet applet。有关静态和动态规则的详细信息，参见第 130 页中的 “定义 Netlet 规则”。
3. Netlet 将侦听在 Netlet 规则中定义的本地端口。
4. Netlet 将在远程客户机与主机之间通过 Netlet 规则中指定的端口而设置一个频道。

使用 Netlet

为了使 Netlet 能够根据不同组织中各个用户的需要进行工作，您需要完成以下操作：

1. 根据用户要求，确定需要创建静态规则还是动态规则。参见第 133 页中的 “规则类型”。
2. 从 Portal Server 管理控制台配置 Netlet 服务的选项。有关配置 Netlet 的信息，参见第 11 章。

3. 确定规则应基于组织、角色还是用户，并按要求在每一级别进行修改。有关组织、角色和用户的详细信息，参见《Portal Server 管理指南》。

注 – 请勿本地化 `srapNetletServlet.properties` 文件中框架集参数的值。

从远程主机下载 Applet

有时，URL 会返回一个页面，该页面包含需要从远程机器获取的嵌入式 applet。但是，Java 安全性只允许 applet 与从自己中下载的主机进行通信。要允许 applet 通过本地网络端口与网关进行通信，您需要检查 Access Manager 管理控制台上的“下载 Applet”字段，并指定以下语法：

local-port:server-host:server-port

其中

local-port 是 Netlet 侦听来自 applet 的通信量的本地端口

server-host 是下载 applet 之处

server-port 是用于下载 applet 的端口

定义 Netlet 规则

Netlet 配置由 Netlet 规则定义，这些规则使用“Secure Remote Access”配置选项卡下的 Portal Server 管理控制台进行配置。可以为组织、角色或用户配置 Netlet 规则。如果为角色或用户配置 Netlet 规则，请在选择组织后选择所需的角色或用户。

注意 – Netlet 规则不支持多字节条目。请勿为 Netlet 规则中的任何字段指定多字节字符。

Netlet 规则不可包含任何高于 64000 的端口号。

第 130 页中的 “定义 Netlet 规则”列出了 Netlet 规则中的字段。

表 6–1 Netlet 规则中的字段

参数	描述	值
规则名称	指定 Netlet 规则的名称。需要为每一规则指定一个唯一的名称。这在定义用户对指定规则的访问权限时是很有用的。	
加密密码	定义加密密码，或指定用户可从中进行选择的密码列表。	您所选择的密码会以列表形式出现在 Netlet 提供者中。用户可从选定列表中选择所需的密码。 默认值 - 使用在 Netlet 管理控制台中指定的“默认 VM 本地密码”和“默认 Java 插件密码”。
远程应用程序 URL	指定当用户在 Netlet 提供者中单击相关链接时浏览器所打开的 URL。浏览器打开应用程序的窗口，并连接到本地端口号（稍后在规则中指定）处的 `localhost`。 您需要指定相关的 URL。	由 Netlet 规则所调用的应用程序的 URL。例如，`telnet://localhost:30000`。 如果应用程序使用 applet 调用应用程序，请指定一个 URL。 `null`– 如果应用程序不是由 URL 启动或由桌面控制，请设定此值。对于不基于网络的应用程序而言该项通常为 true。
启用下载 Applet	指示是否有必要为本规则下载 applet。	■ *Client Port* 表示客户机上的目标端口。该端口必须要不同于默认回环端口。为每一规则指定一个唯一的**本地端口**。 ■ *Server Host* 是从中下载 applet 的服务器名称。 ■ *Server Port* 代表服务器上用于下载 applet 的端口。 如果要下载 applet，且未指定服务器，则从 Portal Server 主机下载 applet。
启用扩展会话	当 Netlet 处于活动状态时，它控制 Portal Server 会话的空闲超时。	仅当 Netlet 处于活动状态而其余门户应用程序空闲时，才选中此复选框以保持门户会话处于活动状态。默认情况下，不会选择此选项。

表 6–1 Netlet 规则中的字段 （续）

参数	描述	值
将本地端口映射至目标服务器端口	本地端口	Netlet 进行侦听的客户机上的端口。 *local-port* 的值必须是唯一的。不能在一个以上的规则中指定特定端口号。 如果要为多个连接指定多台主机，需指定多个本地端口。有关语法，参见第 137 页中的“具有多个主机连接的静态规则”。 对于 FTP 规则，本地端口值必须是 30021。
	目标主机	Netlet 进行侦听的客户机上的端口。 Netlet 连接的收件人。 *host* - 接收 Netlet 连接的主机名。它用于静态规则中。使用简单主机名（如 `siroe`）或全限定 DNS 样式的主机名（如 `siroe.mycompany.com`）。由于以下原因，指定多个主机： *local-port* 的值必须是唯一的。不能在一个以上的规则中指定特定端口号。 如果要为多个连接指定多台主机，需指定多个本地端口。有关语法，参见第 137 页中的“具有多个主机连接的静态规则”。 对于 FTP 规则，本地端口值必须是 30021。 与指定的每一台主机建立连接。需要为指定的每一台主机指定相应的客户机与目标端口。有关语法，参见第 137 页中的“具有多个主机连接的静态规则”。 尝试连接到指定主机列表中的任何一台可用主机。有关语法，参见第 138 页中的“具有多主机选择的静态规则”。 `TARGET` - 在语法中指定 `TARGET` 的规则为动态规则。`TARGET` 表示最终用户可在桌面 Netlet 提供者中指定一台或多台所需的目标主机。 单个规则中不能同时具有静态主机和 TARGET。

表 6–1 Netlet 规则中的字段 （续）

参数	描述	值
目标端口	目标主机上的端口 除主机与目标主机外，还必须要指定一个目标端口。 在有多台目标主机的情况下，可以指定多个目标端口。以 `port1+port2+port3-port4+port5` 格式指定多个端口。 端口号之间的加号 (+) 表示某单一目标主机的备选端口。 端口号之间的减号 (-) 是不同目标主机端口号之间的分隔符。 在此，Netlet 按顺序依次使用 `port1`、`port2` 和 `port3`，以尝试连接到指定的第一台目标主机。如果失败，Netlet 使用 `port4` 和 `port5`（按此顺序）以尝试连接到第二台主机。 只可为静态规则配置多个端口。	

对于要从 Portal Server 获得会话通知的网关，将以下内容：

```
com.iplanet.am.jassproxy.trustAllServerCerts=true
```

添加到 Portal Server 上的以下属性文件中：

Portal Server 上的 `/etc/opt/SUNWam/config/AMConfig.`*instance-name* `.properties`

规则类型

根据规则中目标主机的指定方式，将 Netlet 规则分为两种类型。

静态规则

静态规则将目标主机指定为规则的一部分。如果创建静态规则，用户就无权指定所需的目标主机。在下面的示例中，`sesta` 是目标主机。

规则名称	加密密码	URL	启用下载 Applet	启用扩展会话	将本地端口映射至目标服务器端口
ftpstatic	SSL_RSA_WITH_RC 4_128_MD5	null	false	true	■ 本地端口：30021 ■ 目标主机：sesta ■ 目标端口：21

可为静态规则配置多台目标主机和多个端口。有关示例，参见第 137 页中的 “具有多个主机连接的静态规则”。

动态规则

在动态规则中，不会将目标主机指定为规则的一部分。用户可以在 Netlet 提供者中指定所需的目标主机。在下面的示例中，TARGET 是目标主机的占位符。

规则名称	加密密码	远程应用程序 URL	启用下载 Applet	启用扩展会话	将本地端口映射至目标服务器端口
ftpdynamic	SSL_RSA_WIT H_RC4_128_MD5	null	选中复选框	选中复选框	■ 本地端口：30021 ■ 目标主机 ：TARGET ■ 目标端口：21

加密密码

根据加密密码，可将 Netlet 规则进一步分类如下：

- **用户可配置密码规则** - 在此规则中，您可以指定一个用户可从中进行选择的密码列表。这些可选密码以列表形式出现在 Netlet 提供者中。用户可从该列表中选择所需的密码。在下面的示例中，用户可从多个密码中进行选择。

规则名称	加密密码	远程应用程序 URL	启用下载 Applet	启用扩展会话	将本地端口映射至目标服务器端口
Telnet	SSL_RSA_WITH_RC4 _ 128_SHA	null	选中复选框	选中复选框	■ 本地端口：30000 ■ 目标主机 ：TARGET ■ 目标端口：23
	SSL_RSA_WITH_RC4 _ 128_MD5				

注 – 尽管 Portal Server 主机可能启用了多个不同的密码，但用户可以只从作为 Netlet 规则一部分而配置的密码列表中进行选择。

有关 Netlet 所支持的密码列表，参见第 135 页中的 “支持的密码”。

- **管理员配置的密码规则** - 在此规则中，密码被 定义为 Netlet 规则的一部分。用户无权选择所需的密码。在下面的示例中，密码被配置为 SSL_RSA_WITH_RC4_128_MD5。

规则名称	加密密码	远程应用程序 URL	启用下载 Applet	启用扩展会话	将本地端口映射至目标服务器端口
Telnet	SSL_RSA_WITH_RC4_128_MD5	null	选中复选框	选中复选框	■ 本地端口：30000 ■ 目标主机：TARGET ■ 目标端口：23

有关 Netlet 所支持的密码列表，参见第 135 页中的 “支持的密码”。

支持的密码

第 135 页中的 “支持的密码”列出了 Netlet 支持的密码。

表 6–2 支持密码的列表

密码
本机 VM 密码
KSSL_SSL3_RSA_WITH_3DES_EDE_CBC_SHA
KSSL_SSL3_RSA_WITH_RC4_128_MD5
KSSL_SSL3_RSA_WITH_RC4_128_SHA
KSSL_SSL3_RSA_EXPORT_WITH_RC4_40_MD5
KSSL_SSL3_RSA_WITH_DES_CBC_SHA
Java 插件密码
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5
SSL_RSA_WITH_DES_CBC_SHA
SSL_RSA_WITH_NULL_MD5
TLS_RSA_WITH_AES_128_CBC_SHA

表 6–2 支持密码的列表 （续）

密码
TLS_RSA_WITH_AES_256_CBC_SHA

向后兼容性

Portal Server 的早期版本不支持将密码作为 Netlet 规则的一部分。考虑到不带密码的现有规则的向后兼容性，这些规则将使用默认密码。不带密码的现有规则，如：

规则名称	加密密码	远程应用程序 URL	启用下载 Applet	启用扩展会话	将本地端口映射至目标服务器端口
Telnet		`telnet://localhost:30000`	请勿选中复选框	选中复选框	■ 本地端口：30000 ■ 目标主机：TARGET ■ 目标端口：23

会被解释为：

规则名称	加密密码	远程应用程序 URL	启用下载 Applet	启用扩展会话	将本地端口映射至目标服务器端口
Telnet	默认密码	`telnet://localhost:30000`	请勿选中复选框	选中复选框	■ 本地端口：30000 ■ 目标主机：TARGET ■ 目标端口：23

这与将“加密密码”字段选为“默认值”的“管理员配置规则”相似。

注 – Netlet 规则无法包含任何大于 64000 的端口号。

Netlet 规则示例

本部分包含 Netlet 规则的几个示例以说明 Netlet 语法的作用机理。

- 第 136 页中的 “基本静态规则”
- 第 137 页中的 “具有多个主机连接的静态规则”
- 第 139 页中的 “调用 URL 的动态规则”
- 第 140 页中的 “下载 Applet 的动态规则”

基本静态规则

该规则支持客户机与 `sesta` 机器之间的 Telnet 连接。

规则名称	加密密码	远程应用程序 URL	下载 Applet	扩展会话	将本地端口映射至目标服务器端口
myrule	SSL_RSA_WITH_RC4_128_MD5	null	请勿选中复选框	true	■ 本地端口：1111 ■ 目标主机：sesta ■ 目标端口：23

其中

myrule 是规则名称。

SSL_RSA_WITH_RC4_128_MD5 表示要使用的密码。

null 表示该应用程序并非由 URL 调用或通过桌面运行。

false 表示客户机并不会下载 applet 以运行该应用程序。

true 表示当 Netlet 连接处于活动状态时 Portal Server 不应超时。

1111 是客户机上的端口，Netlet 在此侦听来自目标主机的连接请求。

sesta 是 Telnet 连接中的收件人主机名称。

23 是目标主机上用于连接的端口号，在本例中为众所周知的 Telnet 端口。

桌面 Netlet 提供者不显示链接，但 Netlet 会自动启动，并侦听指定端口 (1111)。指示用户启动客户机软件 - 此情况下为连接至端口 1111 上的 localhost 的 Telnet 会话。

例如，要启动 Telnet 会话，客户需要在终端的 UNIX 命令行中键入以下内容：

```
telnet localhost 1111
```

具有多个主机连接的静态规则

该规则支持从客户机到两台机器 sesta 和 siroe 的 Telnet 连接。

规则名称	加密密码	远程应用程序 URL	启用下载 Applet	启用扩展会话	将本地端口映射至目标服务器端口
myrule	SSL_RSA_WITH_RC4_128_MD5	null	请选中复选框	选中复选框	■ 本地端口：1111–1234 ■ 目标主机：sesta-siroe ■ 目标端口：23

其中

23 是目标主机上用于连接的端口号 – Telnet 的专用端口。

1111 是客户机上的端口，Netlet 在此侦听来自第一台目标主机 sesta 的连接请求。

1234 是客户机上的端口，Netlet 在此侦听来自第二台目标主机 siroe 的连接请求。

该规则中的前六个字段与第 136 页中的 “基本静态规则”中相同。区别在于标识第二台目标主机的另外三个字段。

在向规则中添加附加目标时，必须为每一台新目标主机添加三个字段，即**本地端口**、**目标主机**和**目标端口**。

注 – 您可以用多组三字段来描述与每一目标主机的连接。如果远程客户机是基于 UNIX 的，切不可使用小于 2048 的侦听端口号，因为编号较低的端口将受限制，而且您必须是根用户才能启动侦听器。

该规则与前一规则作用相同。Netlet 提供者不显示任何链接，但 Netlet 会自动启动，并侦听指定的两个端口（1111 和 1234）。用户需要启动客户机软件（在本例中为 Telnet 会话，它连接至端口 1111 上的 localhost 或端口 1234 上的 localhost）以连接到第二个示例中的主机。

具有多主机选择的静态规则

使用该规则可指定多台备选主机。如果与规则中第一台主机的连接失败，Netlet 会尝试连接指定的第二台主机，依此类推。

规则名称	加密密码	远程应用程序 URL	启用下载 Applet	启用扩展会话	将本地端口映射至目标服务器端口
gojoe	SSL_RSA_WITH_RC4_128_MD5	/gojoe.html	■ 客户机端口：8000 ■ 服务器主机：gojoeserver ■ 服务器端口：8080	选中复选框	■ 本地端口：10491 ■ 目标主机：siroe+sesta ■ 目标端口：35+26+491-35+491

其中

10491 是客户机上的端口，Netlet 在此侦听来自目标主机的连接请求。

依据可用的端口，Netlet 将首先尝试与端口 35、端口 26 和端口 491 上的 siroe 建立连接（均按相同顺序）。

如果无法连接到 siroe，则 Netlet 会尝试连接到端口 35 和 491 上的 sesta（按相同顺序）。

主机之间的加号 (+) 表示备选主机。

端口号之间的加号 (+) 表示某单一目标主机的备选端口。

端口号之间的减号 (-) 是不同目标主机端口号之间的分隔符。

注 – 会尝试顺次连接到链条中提供的主机。例如，如果规则为 `siroe+ sesta`，则会首先尝试连接到 `siroe`。如果连接失败，则会尝试连接到 `sesta`。如果规则中首先列出的主机无法在活动的网络中实际使用，则连接到下一个可用主机所需花费的时间将会随着规则中不可用主机数的增加而增加。

调用 URL 的动态规则

该规则允许用户配置所需的目标主机，从而使用户可通过 Netlet 远程登录到各个主机上。

规则名称	加密密码	远程应用程序 URL	启用下载 Applet	启用扩展会话	将本地端口映射至目标服务器端口
myrule	`SSL_RSA_WITH_RC4_128_MD5`	telnet://localhost:30000	请勿选中复选框	选中复选框	■ 本地端口：30000 ■ 目标主机：TARGET ■ 目标端口：23

其中

`myrule` 是规则名称。

SSL_RSA_WITH_RC4_128_MD5 表示要使用的密码。

`telnet://localhost:30000` 是由规则所调用的 URL。

`false` 表示将不会下载任何 applet。

`Extend Session (true)` 表示当 Netlet 连接处于活动状态时 Portal Server 不应超时。

`30000` 是客户机上的端口，Netlet 在此侦听针对此规则的连接请求。

`TARGET` 表示使用 Netlet 提供者的用户需要配置的目标主机。

`23` 是由 Netlet 打开的目标主机上的端口，在本例中为众所周知的 Telnet 端口。

▼ 在添加规则之后运行 Netlet

添加该规则之后，用户必须要完成一些步骤才能使 Netlet 按照预期方式运行。用户需要在客户机端执行以下操作：

1. **在标准 Portal Server 桌面的 Netlet 提供者部分单击“ 编辑 ”。**
   在“添加新目标”部分的“规则名称”下面会列出新的 Netlet 规则。
2. **选择规则名称，并键入目标主机的名称。**

3 **保存更改。**
用户返回桌面，同时在 Netlet 提供者部分会看到新的链接。

4 **单击新链接。**
将会启动一个新的浏览器，转至 Netlet 规则中给出的 URL。

注 – 通过重复以上步骤可以为同一规则添加多台目标主机。只有选定的最后一个链接是活动的。

下载 Applet 的动态规则

此规则定义了从客户机到动态分配主机的连接。它会从 applet 所处的服务器上将 GO-Joe applet 下载到客户机中。

规则名称	加密密码	远程应用程序 URL	启用下载 Applet	扩展会话	将本地端口映射至目标服务器端口
gojoe	SSL_RSA_WITH_RC4_128_MD5	/gojoe.html	■ 客户机端口：8000 ■ 服务器主机：gojoeserver ■ 服务器端口：8080	选中复选框	■ 本地端口：3399 ■ 目标主机：TARGET ■ 目标端口：58

其中

`gojoe` 是规则名称。

SSL_RSA_WITH_RC4_128_MD5 表示要使用的密码。

例如，`/gojoe.html` 是包含 applet 的 HTML 页的路径，该路径应相对于部署门户的 Web 容器文档根目录。

`8000:server:8080` 指明端口 8000 是客户机上用于接收 applet 的目标端口，`gojoeserve` 是提供 applet 的服务器的名称，`8080` 则是自其下载 applet 的服务器上的端口。

`Extended Session (true)` 表示当 Netlet 连接处于活动状态时 Portal Server 不应超时。

`3399` 是客户机上的端口，Netlet 在此侦听此类型的连接请求。

`TARGET` 表示使用 Netlet 提供者的用户需要配置的目标主机。

`58` 是由 Netlet 打开的目标主机上的端口，在本例中为 GoJoe 的端口。端口 58 是目标主机侦听其自身通信量的端口。Netlet 将信息从新 applet 传递至该端口。

Netlet 规则示例

第 141 页中的 “Netlet 规则示例”列出了一些常见应用程序的 Netlet 规则示例。

该表由 7 列组成，分别对应于 Netlet 规则中的以下字段：规则名称、URL、下载 Applet、本地端口、目标主机、目标端口。最后一列中包括对规则的描述。

注 – 第 141 页中的 “Netlet 规则示例”未列出 Netlet 规则的“密码”和“扩展会话”字段。对于所提供的示例，假定这两个字段分别为 "SSL_RSA_WITH_RC4_128_MD5" 和 "true"。

表 6–3 Netlet 规则示例

规则名称	远程应用程序 URL	启用下载 Applet	将本地端口映射至目标服务器端口	描述
IMAP	null	请勿选中复选框	■ 本地端口：10143 ■ 目标主机：imapserver ■ 目标端口：143	客户机端的 Netlet **本地端口**不必与服务器端的**目标端口**相同。如果您使用的不是标准的 IMAP 和 SMTP 端口，请确保配置客户机以将其连接在不同于标准端口的端口上。 Solaris 客户机用户不能连接到小于 1024 的端口号，除非他们以超级用户身份运行。
SMTP	null	请勿选中复选框	■ 本地端口：10025 ■ 目标主机：smtpserver ■ 目标端口：25	
Lotus Web 客户机	null	请勿选中复选框	■ 本地端口：80 ■ 目标主机：lotus-server ■ 目标端口：80	该规则将指示 Netlet 侦听端口 80 上的客户机，并连接到端口 80 上的 lotus-server 服务器。“Lotus Web 客户机”的一项要求就是客户机侦听端口必须与服务器端口匹配。

表 6–3 Netlet 规则示例 （续）

规则名称	远程应用程序 URL	启用下载 Applet	将本地端口映射至目标服务器端口	描述
Lotus Notes 非 Web 客户机	null	请勿选中复选框	■ 本地端口：1352 ■ 目标主机：lotus-domino ■ 目标端口：1352	利用此项规则，Lotus Notes 客户机可以通过 Netlet 连接到 Lotus Domino 服务器。需确保在客户机尝试连接到服务器时，它切不可指向 `localhost` 作为服务器名称。它必须指向 Lotus Domino 服务器的实际服务器名称。服务器名称必须与服务器的系统名称相同。当使用 Netlet 时，客户机必须将该名称解析为 `127.0.0.1`。实现此目的的两种方式为： ■ 设置客户机名称以使其指向客户机主机表中的 `127.0.0.1`。 ■ 导出指向 `127.0.0.1` 的服务器名称的 DNS 条目。 服务器名称必须是安装期间用于配置 Domino 服务器的同一服务器名称。

表 6–3 Netlet 规则示例 （续）

规则名称	远程应用程序 URL	启用下载 Applet	将本地端口映射至目标服务器端口	描述

Microsoft Outlook 和 Exchange Server
 	
它不适用于 Windows NT、2000 和 XP。对于 Windows NT、2000 和 XP，请借助于重写器使用 Outlook Web Access
	null	请勿选中复选框	■ 本地端口：135
			■ 目标主机：exchange
			■ 目标端口：135
				该规则指示 Netlet 侦听客户机上的端口 135，并连接到端口 135 上的服务器 `exchange`。Outlook 客户机使用此端口进行首次尝试以联络 Exchange 服务器，并确定和该服务器通话时要使用的后续端口。
				
				在客户机上：
				■ 用户必须将 Outlook 客户机中所配置的 Exchange 服务器的主机名更改为 `localhost`。该选项的位置随 Outlook 版本的不同而不同。
				■ 用户必须使用主机文件将 Exchange 服务器的主机名（单一且全限定）映射到 IP 地址 `127.0.0.1`。
				■ 在 Windows 95 或 98 中，该文件位于 `\\Windows\\Hosts` 下
				■ 在 Windows NT4 中，该文件位于 `\\WinNT\\System32\\drivers\\etc\\Hosts` 下。
				该项形式如下：
				`127.0.0.1 exchange`
				`exchange.company.com`
				Exchange 服务器将其本身的名称发送回 Outlook 客户机。该映射可确保 Outlook 客户机使用 Netlet 客户机连接回服务器。

FTP	null	请勿选中复选框	■ 本地端口：30021 ■ **目标主机**：*your-ftp_server.your-domain* ■ 目标端口：21	您可以利用所控制的最终用户帐户为“FTP 服务器”提供 FTP 服务。这可确保在最终用户系统和单一位置之间进行安全的远程 FTP 传输。若无用户名，则将 FTP URL 解释为一个匿名的 FTP 连接。 您**必须**将端口 30021 定义为 Netlet FTP 规则的本地端口。 使用 Netlet 连接支持动态 FTP。

表 6–3 Netlet 规则示例 （续）

规则名称	远程应用程序 URL	启用下载 Applet	将本地端口映射至目标服务器端口	描述
Netscape 4.7 邮件客户端	null	请勿选中复选框	■ 本地端口：30143、30025。 ■ 目标主机：TARGET ■ 目标端口：10143	在 Netscape 客户机中，用户需要指定： 用于 IMAP 或接收邮件的 `localhost:30143` 用于 SMTP 或发送邮件的 `localhost:30025`
Graphon	third_party/xsession_start.html	选中复选框	■ 本地端口：10491 ■ 目标主机：TARGET ■ 目标端口：491	这是用于通过 Netlet 访问 Graphon 的规则。`xsession_start.html` 与 Graphon 捆绑在一起。
Citrix	third_party/citrix_start.html	选中复选框	■ 本地端口：1494 ■ 目标主机：TARGET ■ 目标端口：1494	这是用于通过 Netlet 访问 Citrix 的规则。`citrix_start.html` 与 Citrix 捆绑在一起。
RemoteControl	third_party/pca_start.html	选中复选框	■ 本地端口：5631 5632 ■ 目标主机：TARGET TARGET ■ 目标端口：5631 5632	这是用于通过 Netlet 访问“远程控制”的规则。`pca_start.html` 与“远程控制”捆绑在一起。

Netlet 日志信息

netlet applet 或 jws 的客户端日志会在客户端的 java 控制台上显示。

netlet 的服务器端日志在 `/var/opt/SUNWportal/portals/<portal_ID>/logs/<INSTANCE_ID>` 目录下的 `portal.0.0.log` 文件中显示。

在 Sun Ray 环境中运行 Netlet

如果您要在 Sun Ray 环境中运行需要将 applet 下载到客户机上的应用程序，就需要更改 HTML 文件。这里是一个示例文件，向您显示需要完成的必要修改。

新 HTML 文件

```
<!-- @(#)citrix_start.html 2.1
98/08/17 Copyright (c) 1998 i-Planet, Inc., All rights reserved.-->
```

```
<html>
<script language="JavaScript">
var KEY_VALUES;  // KEY_VALUES[\qkey\q] = \qvalue\q;
function retrieveKeyValues() {
      KEY_VALUES = new Object();
      var queryString  = \q\q + this.location;
      queryString = unescape(queryString);
      queryString = queryString.substring((queryString.indexOf(\q?\q)) + 1);
      if (queryString.length < 1) {
          return false; }
      var keypairs = new Object();
      var numKP = 0;
      while (queryString.indexOf(\q&\q) > -1) {
        keypairs[numKP] = queryString.substring(0,queryString.indexOf(\q&\q));
        queryString = queryString.substring((queryString.indexOf(\q&\q)) + 1);
        numKP++;
      }
      // Store what\qs left in the query string as the final keypairs[] data.
      keypairs[numKP++] = queryString;
      var keyName;
      var keyValue;
      for (var i=0; i < numKP; ++i) {
        keyName = keypairs[i].substring(0,keypairs[i].indexOf(\q=\q));
        keyValue = keypairs[i].substring((keypairs[i].indexOf(\q=\q)) + 1);
        while (keyValue.indexOf(\q+\q) > -1) {
          keyValue = keyValue.substring(0,keyValue.indexOf(\q+\q)) + \q \q
           + keyValue.substring(keyValue.indexOf(\q+\q) + 1);

        }
        keyValue = unescape(keyValue);
          // Unescape non-alphanumerics
        KEY_VALUES[keyName] = keyValue;
      }
}
function getClientPort(serverPort) {
    var keyName = "clientPort[\q" + serverPort +"\q]";
    return KEY_VALUES[keyName];
}
function generateContent() {
    retrieveKeyValues();
    var newContent =
        "<html>\\n"
          + "<head></head>\\n"
          + "<body>\\n"
          + "<applet code=\\"com.citrix.JICA.class\\" archive=\\
               "JICAEngN.jar\\" width=800 height=600>\\n"
          + "<param name=\\"cabbase\\" value=\\"JICAEngM.cab\\">\\n"
          + "<param name=\\"address\\" value=\\"localhost\\">\\n"
```

```
          + "<param name=ICAPortNumber value="
          + getClientPort(\q1494\q)
          + ">\\n"
          + "</applet>\\n"
          + "</body>\\n"
          + "</html>\\n";
    document.write(newContent);
}
</script>
<body onLoad="generateContent();">
</body>
</html>
```

弃用的 HTML 文件

```
<html>
<body>
<applet code="com.citrix.JICA.class" archive=
   "JICAEngN.jar" width=800 height=600>
<param name="cabbase" value="JICAEngM.cab">
<param name="address" value="localhost">
<param name=ICAPortNumber value=1494>
</applet>
</body></html>
```

第 2 部分

配置 Secure Remote Access 服务器

可使用 Portal Server 管理控制台中“Secure Remote Access”选项卡下的可用选项来设置大部分属性。默认情况下，所创建的任何组织或用户均会继承这些值。

您可在组织级别、角色级别和用户级别配置与 Secure Remote Access 有关的属性，以下情况除外：

- 不能在用户级别设置“冲突解决方案级别”。参见第 29 页中的 “设置冲突解决方案”。
- 只能在组织级别设置“MIME 类型配置文件位置”属性。

在组织级别设置的值会由该组织下的所有角色和用户继承。在用户级别设置的值会覆盖在组织级别或角色级别设置的相应值。

您可在“服务配置”级别对属性值进行更改。只有添加了新的组织后，才反映这些新的属性值。

本节包括以下章节：

◆ ◆ ◆ 第 7 章

7 配置 Secure Remote Access 服务器访问控制

本章说明如何从 Sun Java System Portal Server 管理控制台中允许或拒绝用户的访问。

配置访问控制

您可以使用此字段指定最终用户无法通过网关访问的 URL 列表。网关会在检查“允许的 URL”列表之前先检查“拒绝的 URL”列表。

可指定最终用户能够通过网关访问的所有 URL。默认情况下，此列表有一通配符项 (*)，表示可以访问所有 URL。如果要允许访问所有 URL，仅限制访问特定 URL，可将受限 URL 添加到“拒绝的 URL”列表中。用同样的方法，如果希望仅允许访问特定 URL，则将“拒绝的 URL”字段留为空白，在“允许的 URL”字段中指定所需的 URL。

SRA 软件中的“访问控制”服务允许您控制各个主机的单点登录功能。为使单点登录功能可用，必须在“网关”服务中启用“启用 HTTP 基本验证”选项。

通过“访问控制”服务，可禁用某些主机的单点登录功能。这意味着最终用户在每次连接到需要 HTTP 基本验证的主机时都需要进行验证，除非启用“每个会话的单点登录”功能。

如果某台主机的单点登录功能已被禁用，则用户可在单个 Portal Server 会话中重新连接到该主机。例如，假定已禁用至 `abc.sesta.com` 的单点登录。用户第一次连接到此站点时，需要进行验证。用户可以浏览其他页面，并稍后返回此页面，如果该页面在同一 Portal Server 会话中，则无需验证。

▼ 配置访问控制

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡。**
3. **选择“ 访问控制 ”选项卡。**
4. **修改以下属性：**

属性名称	描述
COS 优先	指定用于确定属性值继承性的值。有关此属性的详细信息，参见 Sun Java System Directory Server 管理指南。
每个会话的单点登录	选中“启用”复选框以启用单点登录会话。
禁用单点登录主机	以 `abc.siroe.com` 格式输入主机名。
允许的验证级别	输入允许的验证级别。用星号可表示允许所有级别。默认值为星号。
允许/拒绝访问的 URL	在 URL 字段中输入允许或拒绝通过网关访问的 URL。输入 URL 的格式为：`http://abc.siroe.com`。在“操作”下拉列表下，单击相应的“允许”或“拒绝”选项。 也可以使用正则表达式，如 `http://*.siroe.com`。在这种情况下，用户不能访问 `siroe.com` 域中的所有主机。 在检查允许的 URL 列表之前，网关会首先检查已拒绝访问的 URL。 **注 –** 默认情况下，“允许的 URL”字段具有一个 `*`，它表示可通过网关访问所有的 URL。

注 – 安装 SRA 时，默认情况下“访问控制”服务并非对所有用户均可用。该服务仅适用于安装期间默认创建的 `amadmin` 用户。没有此服务，其他用户便无法通过网关访问桌面。以 `amadmin` 身份登录，并将此项服务指定给所有用户。

5. **单击“ 保存 ”以完成修改。**

◆ ◆ ◆ 第 8 章

配置 Secure Remote Access 网关

本章说明如何从 Sun Java System Portal Server 管理控制台配置网关属性。

本章包含以下各节：

开始之前

- 要创建网关配置文件，参见第 32 页中的 “创建网关配置文件”

配置配置文件核心选项

本节说明以下任务：

配置启动模式

如果安装时选择在 HTTPS 模式下运行网关，安装完成后，网关将以 HTTPS 模式运行。在 HTTPS 模式中，网关接受来自浏览器的 SSL 连接，而拒绝非 SSL 连接。不过，您也可以将网关配置为在 HTTP 模式下运行。这样将提高网关性能，因为管理 SSL 会话以及加密、解码 SSL 通信的开销均未涉及到。

▼ 配置启动模式

1 **以管理员身份登录到 Portal Server 管理控制台。**

2 **选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。**

3 **选择“ 核心 ”选项卡。**

4 **修改以下属性：**

HTTP 连接	选中“HTTP 连接”复选框以使网关能够接受非 SSL 连接。
HTTP 端口	输入 HTTP 端口号。默认值为 80。
HTTPS 连接	选中“HTTPS 连接”复选框以使网关能够接受 SSL 连接。默认情况下，此选项已选中。
HTTPS 端口	输入 HTTPS 端口号。默认值为 443。

注 – 可使用《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”（参见《Sun Java System Portal Server 7.2 Command-Line Reference》）修改以下属性

/space/PS/portal/bin/`psadmin set-attribute -u amadmin -f` */space/PS/portal/bin/ps_password -p portal1* `-m gateway --gateway-profile` *profileID* `-a` *sunPortalGatewayDomainsAndRulesets -A $entry*

- sunPortalGatewayDefaultDomainAndSubdomains=Default Domains
- sunPortalGatewayLoggingEnabled=Enable Logging
- sunPortalGatewayEProxyPerSessionLogging=Enable per Session Logging
- sunPortalGatewayEProxyDetailedPerSessionLogging=Enable Detailed per Session Logging
- sunPortalGatewayNetletLoggingEnabled=Enable Netlet Logging
- sunPortalGatewayEnableMIMEGuessing=Enable MIME Guessing
- sunPortalGatewayParserToURIMap=Parser to URI Mappings
- sunPortalGatewayEnableObfuscation=Enable Masking
- sunPortalGatewayObfuscationSecretKey=Seed String for Masking
- sunPortalGatewayNotToObscureURIList=URIs not to Mask
- sunPortalGatewayUseConsistentProtocolForGateway=Make Gateway protocol Same as Original URI Protocol
- sunPortalGatewayEnableCookieManager=Store External Server Cookies
- sunPortalGatewayMarkCookiesSecure=Mark Cookies as secure

---

5 从终端窗口中重新启动网关：

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

配置核心组件

Netlet 使用户可以在不安全的网络上（如 Internet）安全地运行常用 TCP/IP 服务。您可以运行 TCP/IP 应用程序（如 Telnet 和 SMTP）、HTTP 应用程序和任何使用固定端口的应用程序。如果启用了 Netlet，网关就需要判断接收的通信是 Netlet 通信还是 Portal Server 通信。由于网关假定所有接收的通信都是 HTTP 通信或 HTTPS 通信，所以禁用 Netlet 可以减少此类开销。只有在确信不需要与 Portal Server 一同使用任何应用程序时，才可以禁用 Netlet。

▼ 配置组件

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。

3 选择“ 核心 ”选项卡。

4 修改以下属性：

属性名称	描述
Netlet	选中“启用”复选框以启动 Netlet 服务。默认情况下，此选项已选中。
Proxylet	选中“启用”复选框以启动 Proxylet 服务。默认情况下，此选项已选中。

5 使用以下命令选项从终端窗口中重新启动网关：

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

配置基本选项

关于 Cookie 管理属性

许多网站使用 cookie 对用户会话进行跟踪和管理。当网关向网站发送在 HTTP 报头中设置 cookie 的请求时，网关以下述方式丢弃或传送这些 cookie：

- 如果未在网关服务中选中“启用 Cookie 管理”属性，则不会重写 cookie。因此，来自浏览器的 cookie 可能不会到达内联网主机，反之亦然。

- 如果选择了“启用 Cookie 管理”属性，网关将重写 cookie。网关会确保来自浏览器的 cookie 到达内联网的目标主机，反之亦然。

此设置不会应用于 Portal Server 用来跟踪 Portal Server 用户会话的 cookie。该设置由“用户会话 Cookie 被转发到的 URL”URL 选项配置控制。

此设置适用于用户可以访问的所有网站（即不能选择丢弃某些网站的 cookie，而保留其他网站的 cookie）。

注 – 即使在无 cookie 的网关中，也不要从“Cookie 域”列表中删除 URL。有关“Cookie 域”列表的信息，参见 Access Manager 管理指南。

关于 HTTP 基本验证属性

HTTP 基本验证可在网关服务中设置。

网站可以使用“HTTP 基本验证”，要求访问者在浏览网站之前输入用户名和密码（HTTP 响应代码为 401 和 WWW-authenticate：BASIC），从而获得保护。Portal Server 可以保存用户名和密码，这样用户再次访问受 BASIC 保护的 Web 站点时，就不需要重新输入他们的凭证。这些凭证存储在 Directory Server 的用户配置文件中。

此设置不决定用户能否访问受 BASIC 保护的站点，而只确定是否将用户输入的凭证保存在该用户的配置文件中。

此设置适用于用户可以访问的所有网站（即 HTTP 基本验证高速缓冲功能不能对某些网站可用，而对其他网站不可用）。

注 – 如果 Microsoft Internet Information Server (IIS) 是通过 Windows NT 质询/响应（HTTP 响应代码为 401，WWW-Authenticate：NTLM）而不是 BASIC 验证进行保护，则不支持浏览由其提供的 URL。

您也可使用管理控制台中的“访问控制”服务来启用单点登录。

关于 Portal Server 属性

可以为网关配置多个 Portal Server 以为请求提供服务。安装网关时，可能已经指定需要和网关协同工作的 Portal Server。默认情况下，此 Portal Server 会在 Portal Server 字段中列出。可向列表中添加更多的 Portal Server，格式为 `http://`*portal- server-name:port number*。网关会以循环方式尝试联系每个 Portal Server 来为请求提供服务。

关于“用户会话 Cookie 转发到的 URL”属性

portal server 利用 cookie 跟踪用户会话。当网关向服务器提出 HTTP 请求时（例如，当调用桌面 servlet 以生成用户桌面页时），此 cookie 将被转发到服务器。服务器上的应用程序使用该 cookie 来确认并标识用户。

Portal Server 的 cookie 不会被转发到向该服务器以外的其他机器发出的 HTTP 请求，除非在“用户会话 Cookie 转发到的 URL”列表中指定了那些机器上的 URL。因此向此列表中添加 URL 可使 servlet 和 CGI 接收 Portal Server 的 cookie，并使用 API 来标识用户。

URL 使用隐含的后缀通配符进行匹配。例如，列表的默认条目：

```
http://server:8080
```

会使 cookie 被转发到所有以 `http://server:8080` 开始的 URL。

添加：

```
http://newmachine.eng.siroe.com/subdir
```

会使 cookie 被转发到所有开头与该字符串完全相同的 URL。

在此例中，cookie 不会转发到任何以“`http://newmachine.eng/subdir`”开始的 URL，因为该字符串的开头部分与转发列表中的字符串不完全相同。要将 cookie 转发到以这个改变的机器名开始的 URL，必须向转发列表添加新的条目。

同样，cookie 也不会转发到以“`https://newmachine.eng.siroe.com/subdir`”开始的 URL，除非向列表中添加相应的条目。

关于“从 URL 获取会话”属性

选择“从 URL 获取会话”选项后，不管是否支持 cookie，会话信息都会作为 URL 的一部分进行编码。这意味着网关使用在 URL 中找到的会话信息进行验证，而不使用客户机浏览器发出的会话 cookie。

▼ 配置基本选项

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“Secure Remote Access”选项卡，然后单击配置文件名称以修改其属性。**
3. **选择“核心”选项卡。**
4. **修改以下属性：**

属性名称	描述
Cookie 管理	选中“启用”复选框以启用 cookie 管理。 默认情况下，此选项已选中。

属性名称	描述
HTTP 基本验证	选中“启用 HTTP 基本验证”复选框，以启用 HTTP 基本验证。
Portal Server	在字段中以 `http://portal-server-name:port-number` 格式输入 Portal Server，然后单击“添加”。 重复此步骤以将更多 Portal Server 添加到 Portal Server 列表中。
用户会话 Cookie 转发到的 URL	输入“用户会话 Cookie 转发到的 URL”，然后单击“添加”。 重复此步骤以将更多 URL 添加到“用户会话 Cookie 转发到的 URL”列表中。
网关最低验证级别	输入验证级别。 默认情况下，会添加星号以允许在所有级别下进行验证。
从 URL 获取会话	选择“是”以从 URL 检索有关会话的信息。 默认情况下，“否”选项已选中。

配置部署选项

配置代理设置

▼ 配置代理设置

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。**
3. **选择“ 部署 ”选项卡。**
4. **修改以下属性：**

属性名称	描述
使用代理	选中“使用代理服务器”复选框以启用 Web 代理。

属性名称	描述	
Web 代理 URL	在“使用 Web 代理 URL”编辑框中，使用 `http://host name.subdomain.com` 格式输入所需的 URL，然后单击“添加”。 该 URL 添加到“使用 Webproxy URL”列表中。	可以指定即使在禁用“使用代理”选项时，网关也只能通过在“域和子域代理”列表中列出的 Web 代理与某些 URL 联络。您需要在“使用 Webproxy URL”字段中指定这些 URL。有关该值如何影响代理使用的详细信息，参见第 33 页中的 “指定与 Access Manager 联络的代理”。
域和子域代理	即会将该条目添加到“域和子域代理”列表框中。 输入代理信息的格式如下： `domainname proxy1:port1|subdomain1 proxy2:port2|subdomain2 proxy3:port3|* proxy4:port4` * 表示在 * 后定义的代理需要用于所有域和子域，特别指出的除外。 如果未指定代理端口，默认使用端口 8080。	有关如何将代理信息应用到各主机的详细信息，参见第 33 页中的 “指定与 Access Manager 联络的代理”。
代理密码列表	在“代理密码列表”字段中，输入每个代理服务器的相应信息，然后单击“添加”。 输入代理信息的格式如下： `proxyserver|username|password` `proxyserver` 与在“域和子域的代理”列表中定义的代理服务器相对应。	如果代理服务器访问某些或所有站点需要验证，则您需要指定所需的用户名和密码，以便网关验证至特定的代理服务器。
自动代理配置支持	选中“启用自动代理配置支持”复选框以启用 PAC 支持。	如果您选择了“启用自动代理配置”选项，“域和子域代理”字段中的信息将被忽略。网关仅对内联网配置使用“代理自动配置”(Proxy Automatic Configuration, PAC) 文件。有关 PAC 文件的信息，参见第 44 页中的 “使用自动代理配置”。
自动代理配置文件位置	在“位置”字段中，输入 PAC 文件的名称和位置。	

配置重写器代理和 Netlet 代理

关于 NetLet 代理

Netlet 代理通过将安全隧道从客户机，经网关扩展到内联网中的 Netlet 代理，提高了网关和内联网之间 Netlet 通信的安全性。如果已启用 Netlet 代理，Netlet 信息包将由 Netlet 代理解码，然后发送到目标服务器。这将减少需要在防火墙中打开的端口数量。

关于重写器代理

重写器代理可以使网关和内联网之间安全地进行 HTTP 通信。如果未指定重写器代理，那么当用户试图访问内联网中的机器时，网关组件将会直接连接到内联网。重写器代理在安装后不会自动运行。您需要执行以下步骤启用“重定器”代理。

▼ 配置重写器代理和 Netlet 代理

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。**

注 – 确保重写器代理和网关使用相同的网关配置文件。

---

3. **选择“ 部署 ”选项卡。**
4. **修改以下属性：**

属性名称	描述
重写器代理	选中“重写器代理”复选框以启用重写器代理服务。
重写器代理列表	a. 在“重写器代理”编辑框中，使用 `hostname:port` 格式输入主机和端口。 **提示 –** 要确定所需端口是否可用或未使用，在命令行中输入： **`netstat -a | grep`** *port-number* **`| wc -l`** *port-number* 是所需的端口。 b. 单击“添加”。
Netlet 代理	选中“启用 Netlet 代理”复选框以启用 Netlet 代理服务。
Netlet 代理主机	a. 在“Netlet 代理主机”字段中，使用 `hostname:port` 格式输入 Netlet 代理主机和端口。 **提示 –** 要确定所需端口是否可用或未使用，在命令行中输入： **`netstat -a | grep`** *port-number* **`| wc -l`** *port-number* 是所需的端口。 b. 单击“添加”。
通过 Web 代理服务器的 Netlet 隧道	选中“启用通过 Web 代理服务器的 Netlet 隧道”复选框以启用隧道。

5 **在服务器上运行 *portal-server-install-root*/SUNWportal/bin/certadmin 创建重写器代理证书。**

只有当您在安装重写器代理过程中未选择创建证书时，才需要执行此步骤。

6 **以根用户身份登录至安装重写器代理的机器，并启动重写器代理：**

rewriter-proxy-install-root/SUNWportal/bin/rwproxyd -n *gateway-profile-name* start

7 **以根用户的身份登录至安装网关的机器，并重新启动网关：**

./psadmin start-sra-instance -u amadmin -f *passwordfile* -N *profilename* -t gateway

配置安全性选项

配置 PDC 和非验证 URL

▼ 配置 PDC 和非验证 URL

1 **以管理员身份登录到 Portal Server 管理控制台。**

2 **选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。**

3 **选择“ 安全 ”选项卡。**

4 **修改以下属性：**

属性名称	描述
已启用证书的网关主机	a. 将网关名添加到“已启用证书的网关”主机。 以 host1.sesta.com 格式添加网关。 b. 单击“添加”。
免验证 URL	可以指定某些 URL 不需要验证。它们通常是包含图像的目录。 在“免验证 URL”字段中，使用 folder/subfolder 格式输入所需的文件夹路径。 未全限定的 URL（例如，/images）被视为门户 URL。 要添加非门户 URL，请完全限定该 URL，并单击“添加”将此条目添加到“免验证 URL”列表中。
信任的 SSL 域	在“信任的 SSL 域”字段中，输入域名，然后单击“添加”。

配置 TLS 和 SSL 选项

▼ 配置 TLS 和 SSL 选项

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。

3 选择“ 安全 ”选项卡。

4 修改以下属性：

属性名称	描述
40 位加密	如果要允许 40 位（弱）“加密套接字层”(SSL) 连接，请选择此选项。如果没有选择此选项，则只支持 128 位连接。 如果禁用此选项，用户需要确保浏览器的配置支持所需的连接类型。 **注 –** 对于 Netscape Navigator 4.7x，用户需要执行以下操作： a. 在“通信器”菜单中，选择“工具”下的“安全性信息”。 b. 在左侧窗格中单击“导航器”链接。 c. 在“高级安全性 (SSL) 配置”下单击“配置 SSL v2”或“配置 SSL v3”。 d. 启用所需的密码。
Null 密码	选中“启用空密码”复选框以启用空密码。
SSL 密码选择	Secure Remote Access 支持许多标准密码。您可选择支持所有预封装的密码或单独选择所需的密码。您可以为每个网关实例指定特定的 SSL 密码。只要客户机站点中存在任一选定的密码，SSL 信号交换即可成功。
SSL 2.0 版本	选中“启用 SSL 2.0 版本”复选框以启用 2.0 版本。默认情况下，此选项已启用。 您可以启用或禁用 SSL 2.0 版本。禁用 SSL 2.0 意味着仅支持 SSL 2.0 之前版本的浏览器会无法验证到 Secure Remote Access。这将确保更高级别的安全性。
SSL2 密码	选中“启用 SSL 密码选择”复选框选项。 您可以从 SSL 密码列表中选择所需的密码。
SSL 3.0 版本	您可以启用或禁用 SSL 3.0 版本。禁用 SSL 3.0 意味着仅支持 SSL 3.0 的浏览器会无法验证到 SRA 软件。这将确保更高级别的安全性。 选中“启用 SSL 3.0 版本”复选框以启用 3.0 版本。
SSL3 密码	选中“启用 SSL 密码选择”复选框选项。 您可以从 SSL3 密码列表中选择所需的密码。

属性名称	描述
TLS 密码	选中“启用 SSL 密码选择”复选框选项。 您可以从 TLS 密码列表中选择所需的密码。

配置性能选项

配置超时和重试

▼ 配置超时和重试

1. 以管理员身份登录到 Portal Server 管理控制台。
2. 选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。
3. 选择“ 性能 ”选项卡。
4. 修改以下属性：

属性名称	描述
服务器重试时间间隔（秒）	指定在无法使用 Portal Server、重写器代理或 Netlet 代理时（如崩溃或关机），尝试启动它们的各个请求之间的时间间隔（以秒为单位）。
网关超时时间（秒）	指定一个以秒为单位的时间间隔，如超过此值，网关与浏览器的连接就会超时。 在“网关超时”字段中，指定所需的时间间隔（以秒为单位）。
高速缓存套接字超时时间（秒）	指定一个以秒为单位的时间间隔，如超过此值，网关与 Portal Server 的连接就会超时。

配置 HTTP 选项

▼ 配置 HTTP 选项

1. 以管理员身份登录到 Portal Server 管理控制台。
2. 选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。
3. 选择“ 性能 ”选项卡。

4 修改以下属性：

属性名称	描述
线程组合容量最大值	指定所需的线程数。 可以指定网关线程池中可以预创建的最大线程数。
持久性 HTTP 连接	选中“启用持久性 HTTP 连接”复选框，以启用 HTTP 连接。 可在网关处启用 HTTP 持久性连接，以防止为 Web 页面中的所有对象（如图像和样式表）打开套接字。
每个持久连接的最大请求数目	输入最大请求数目。
持久套接字连接超时时间（秒）	输入所需的超时时间（以秒为单位）。
反映周转时间的宽限超时时间（秒）	输入所需的宽限超时时间（以秒为单位）。 这是客户机（浏览器）与网关之间的网络通信的往返时间。 ■ 浏览器发送请求后，该请求到达网关所花费的时间 ■ 网关发送响应与浏览器实际收到响应之间的时间 这取决于网络状况和客户机的连接速度等因素。
最大连接队列长度	指定网关可接受的最大并发连接数量。 指定所需的连接数。

监视 Secure Remote Access 性能

监视可让管理员评估 Secure Remote Access 各种组件的性能。

▼ 监视 Secure Remote Access 性能

1 登录到 Portal Server 管理控制台。

2 选择“ Secure Remote Access ”选项卡，然后单击子菜单中的“ 监视 ”。

3 在“ 监视 ”页内，从下拉式菜单中选择一个代理实例。

4 在“ MBean ”表中选择属性以查看性能值。

配置重写器选项

配置基本选项

▼ 配置基本选项

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。**
3. **选择“ 重写器 ”选项卡。**
4. **修改以下属性：**

属性名称	描述
重写全部 URI	选中“启用所有 URI 的重写”复选框使网关重写所有 URL。 如果您启用了网关服务中的“启用全部 URL 重写”选项，重写器会重写任何 URL，而不检查“域和子域代理”列表中的条目。“域和子域代理”列表中的条目将被忽略。
禁止重写的 URI	在编辑框中添加 URI。 **注 –** 向该列表中添加 #* 可重写 URI（即使规则集中包含 href 规则）。

配置 URI 到规则集的映射

规则集是在 Portal Server 管理控制台中，在Portal Server 配置下的重写器服务中创建的。有关详细信息，参见 Portal Server 管理指南。

创建了规则集之后，可使用“将 URI 映射至规则集”字段将某个域与此规则集相关联。默认情况下，会将以下两个条目添加到“将 URI 映射至规则集”字段中：

- *://*.Sun.COM/portal/*|default_gateway_ruleset
  其中，sun.com 是门户的安装域，/portal 是门户的安装环境
- *|generic_ruleset

这表示默认域的所有页都应用默认网关规则集。对于其他所有页，将会应用一般规则集。默认网关规则集和一般规则集都是预封装的规则集。

注 – 对于所有在桌面上显示的内容，不管内容取自何处，均使用默认域规则集。

例如，假定将桌面配置为凑集来自 URL `yahoo.com` 的内容。Portal Server 位于 `sesta.com` 中。`sesta.com` 的规则集将会应用至获取的内容。

注 – 为其指定规则集的域必须在“域和子域代理”列表中列出。

▼ 配置 URI 到规则集的映射

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。**
3. **选择“ 重写器 ”选项卡。**
4. **修改以下属性：**

属性名称	描述

URI	在“将 URI 映射至规则集”字段中，输入所需的域或主机名和规则集，然后单击“添加”。
	会将此条目添加到“将 URI 映射至规则集”字段中。
	指定域或主机名以及规则集时采用的格式如下：
	`domain-name|ruleset-name`
	例如：
	`eng.sesta.com|default`
	注 – 规则集应用优先顺序为 `hostname-subdomain-domain`。
	例如，基于域的规则集列表中有下列条目：
	`sesta.com|ruleset1`
	`eng.sesta.com|ruleset2`
	`host1.eng.sesta.com|ruleset3`
	■ `ruleset3` 将应用于 `host1` 上的所有页。
	■ 除了从 `host1` 中检索到的页之外，`ruleset2` 将应用于 `eng` 子域中的所有页。
	■ 除了从 `eng` 子域和 `host1` 中检索到的页之外，`ruleset1` 将应用于 `sesta.com` 域中的所有页。

配置解析器至 MIME 类型的映射

重写器有 4 个不同的解析器，可用来根据内容类型（HTML、JAVASCRIPT、CSS 和 XML）对网页进行解析。默认情况下，这些解析器与常见的 MIME 类型相关联。您可以在网关服务的“将解析器映射至 MIME 类型”字段中，将新的 MIME 类型与这些解析器相关联。这会将重写器功能扩展到其他 MIME 类型。

使用分号或逗号分隔多个条目（ “;” 或 “,” ）

例如：

HTML=text/html;text/htm;text/x-component;text/wml; text/vnl/wap.wml

它表示会将含有上述 MIME 的任何内容发送到“HTML 重写器”，并且会应用“HTML 规则”来重写这些 URL。

提示 – 从 MIME 映射列表中删除不必要的解析器可以提高操作速度。例如，如果您确信来自某个内联网的内容不含有任何 JavaScript，则可从 MIME 映射列表中删除 JAVASCRIPT 条目。

▼ 配置解析器至 MIME 类型的映射

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后单击配置文件名称以修改其属性。**
3. **选择“ 重写器 ”选项卡。**
4. **修改以下属性：**

属性名称	描述
解析器	a. 在“将解析器映射至 MIME 类型”字段中，将所需的 MIME 类型添加到编辑框中。可使用分号或逗号分隔多个条目。 以 `HTML=text/html;text/htm` 格式指定该条目 b. 单击“添加”，将所需条目添加到列表中。

配置个人数字证书验证

PDC 由“认证机构”(CA) 发放，并使用该认证机构的私有密钥签名。颁发证书之前，CA 将对请求主体的身份进行验证。因此，PDC 的出现提供了一种功能强大的验证机制。

PDC 含有所有者的公共密钥、所有者名称、到期日期、发放“数字证书”的认证机构名称、序列号以及可能包含的其他信息。

用户可以将 PDC 和编码设备（如 Smart 卡和 Java 卡）用于 Portal Server 中的验证。编码设备与卡中存储的 PDC 具有等效的电子效力。如果用户使用上述任一机制登录，将不会显示登录屏幕和验证屏幕。

PDC 验证过程涉及以下步骤：

1. 用户在浏览器中键入一个连接请求，例如 `https://my.sesta.com`。

   对此请求的响应取决于到 `my.sesta.com` 的网关是否已经配置为接受证书。

   注 – 当网关被配置成接受证书时，它仅接受以证书方式进行的登录，而拒绝其他所有类型的登录。

   网关检查证书是否由已知的“认证机构”发放，是否尚未过期，以及是否未经篡改。如果证书有效，网关将允许用户进入验证过程的下一步。
2. 网关将证书传递给服务器中的 PDC 验证模块。

▼ 配置 PDC 和编码设备

1 **将以下行添加到 Portal Server 机器的 `/etc/opt/SUNWam/config/AMConfig.properties` 文件中：`com.iplanet.authentication.modules.cert.gwAuthEnable=yes`。**

2 **将“必要的证书”导入希望启用 PDC 的网关的证书数据库。要配置证书，参见第 168 页中的“在网关机器上导入根 CA 证书”**

3 **以管理员身份登录到 Access Manager 管理控制台，然后执行以下操作：**

   a. **选择“身份管理”选项卡，然后选择一个“组织”。**

   b. **从“视图”下拉式菜单中单击组织的“服务”。**

   c. **单击“添加”注册证书。**

4 从 Access Manager 管理控制台，执行以下操作：

a. 选择所需的组织，然后单击“证书”旁边的箭头。

b. 在“信任的远程主机”列表框中，突出显示“无”，然后单击“删除”。

c. 在文本字段中输入任何内容，然后单击“添加”。

d. 单击“保存”。

5 从 Access Manager 管理控制台，执行以下操作：

a. 选择所需的组织，然后从“视图”下拉式菜单中选择“服务”。
将显示服务列表。

b. 单击“验证配置”核心服务旁边的箭头，然后单击“新建”。
将显示“新建服务实例”页面。

c. 输入服务实例名称 `gatewaypdc`。

d. 单击“提交”。
将显示 gatewaypdc“服务实例列表”。

e. 单击 gatewaypdc 编辑服务。
将显示 gatewaypdc 显示属性页。

f. 单击“验证配置”旁边的“编辑”链接，然后单击“添加”。
将显示“添加模块”页面。

g. 从“模块名称”字段中选择“证书”并选择“必需”作为执行标准，然后单击“确定”。

h. 单击“确定”完成操作。

6 从 Access Manager 管理控制台，执行以下操作：

a. 单击“核心”旁的箭头。

b. 在“组织验证”模块列表框中，选择 gatewaypdc。

c. 从“用户配置文件”下拉式菜单中选择“动态”。

d. 单击“保存”以完成修改。

7. 以管理员身份登录到 Portal Server 管理控制台，然后执行以下操作：

   a. 选择“ Secure Remote Access ”选项卡，然后选择相应的网关配置文件。

   b. 选择“ 安全 ”选项卡。

   c. 在“ 已启用证书的网关主机 ”列表框中，添加网关名称。

   d. 单击“ 保存 ”。

8. 从终端窗口中重新启动网关配置文件：

   ```
   ./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t
   gateway
   ```

9. 将 CA 签发的客户机证书安装到必须访问启用了 PDC 网关的浏览器。

10. 将客户机证书安装到 JVM 密钥库中。可通过如下所示的方式：从 Windows 机器的“ 开始 ”>“ 设置 ”>“ 控制面板 ”>“ Java ”来访问 JVM 控制面板。

    将以下内容添加到 Applet 运行时参数：

    - `Djavax.net.ssl.keyStore=Path to Keystore`
    - `Djavax.net.ssl.keyStorePassword=password`
    - `Djavax.net.ssl.keyStoreType=type`

11. 访问您的网关配置文件和组织：

    https://gateway:instance-port/YourOrganization

    如果没有提示您输入用户名和密码，则应该使用证书名称登录。

▼ 在网关机器上导入根 CA 证书

1. 在网关机器上导入根 CA 证书。

   a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`

      此时会列出 Certadmin 菜单。

   b. 选择选项 3。输入证书的路径。

   有关详细信息，参见第 10 章。

2. 生成一个“证书签名请求”以提交到 CA。

   a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`
   此时会列出 Certadmin 菜单。

   b. 选择选项 2。输入相应的信息。

   c. 保存该文件。

3. 将“证书签名请求”提交到 CA 并使其获得批准。在 CA 签名后保存证书响应。

4. 在获得 CA 批准后导入“服务器证书”。

   a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`
   此时会列出 Certadmin 菜单。

   b. 选择选项 4。

   c. 指定包含“服务器证书”的文件的位置。

5. 在 Portal Server 机器上导入根 CA 证书。

使用命令行选项配置网关属性

本节为以下任务提供命令行选项，以从终端窗口中配置网关属性：

▼ 管理外部服务器 Cookie 的存储

当启用“存储外部服务器 Cookie”选项时，网关存储并管理通过网关访问的任何第三方应用程序或服务器的 cookie。尽管应用程序或服务器不能服务于 cookieless 设备或依赖于 cookie 进行状态管理，网关也会透明地屏蔽应用程序或服务器，使其不知道网关正服务于 cookieless 设备。

有关 cookieless 设备和客户机检测的信息，参见《*Access Manager* 自定义和 *API* 指南》

- **键入以下命令并按 Enter 键以管理外部服务器 cookie 的存储。**

  - **启用：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a CookieManagement true`

  - **禁用：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a CookieManagement false`

  - **获取属性值：**

    PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a CookieManagement`

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

▼ 启用将 Cookie 标记为安全

将 cookie 标记为安全 cookie 时，浏览器以额外的安全对待该 cookie。安全的实现方式取决于浏览器。如果要使用此功能，必须启用“启用 Cookie 管理”属性。

- **键入以下命令并按 Enter 键以将 cookie 标记为安全。**

  - **启用：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a MarkCookiesSecure true`

  - **禁用：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a MarkCookiesSecure false`

- **获取属性值：**

  PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a MarkCookiesSecure`

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

▼ 创建不使用的代理 URL 列表

网关会尝试直接连接到在“不可使用 Webproxy URL”列表中列出的 URL。Webproxy 将不用于连接到这些 URL。

- **键入以下命令并按 Enter 键以管理不使用的代理 URL。**

  注 – 如果有多个 URL，请使用空格分隔每个 URL。

  - **指定不使用的 URL：**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DontUseWebProxyURL -A` *"LIST_OF_URLS"*

  - **添加到现有 URL 列表：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DontUseWebProxyURL -A` *"LIST_OF_URLS"*

  - **从现有 URL 列表中删除：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DontUseWebProxyURL -E` *"LIST_OF_URLS"*

  - **获取现有 URL 列表：**

    PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DontUseWebProxyURL`

更多信息

另请参见

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

▼ 管理 URI 映射的规则集

Secure Remote Access 支持 Microsoft Exchange 2000 SP3 安装和 Outlook Web Access (OWA) 的 MS Exchange 2003。

1. **添加 URI 到现有列表：**

   PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile default -a DomainsAndRulesets -A "`*URI*`|`*RULE_SET_NAME* *URI*`|`*RULE_SET_NAME*`"`

2. **从现有列表中删除 URI：**

   PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile default -a DomainsAndRulesets -E "`*URI*`|`*RULE_SET_NAME* *URI*`|`*RULE_SET_NAME*`"`

3. **获取现有列表：**

   PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DomainsAndRulesets`

4. **键入以下命令并按 Enter 键以管理 Outlook Web Access 的规则集。**

   - **添加规则集：**

     PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile default -a DomainsAndRulesets -A "`*EXCHANGE2000_SERVER_NAME*`exchange_2000sp3_owa_ruleset"`

   - **删除规则集：**

     PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile default -a DomainsAndRulesets -E "`*EXCHANGE2000_SERVER_NAME* `exchange_2000sp3_owa_ruleset"`

   - **设置 URI 至规则集的映射的列表：**

     PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DomainsAndRulesets "`*URI*`|` *RULE_SET_NAME* *URI*`|`*RULE_SET_NAME* `"`

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

▼ 指定默认域

当 URL 仅包含不带域和子域的主机名时，才会用到默认域。在这种情况下，网关假定主机名在默认域列表中，并进行相应处理。

例如，如果 URL 中的主机名为 `host1`，并且将默认的域和子域指定为 `red.sesta.com`，则主机名会被解析为 `host1.red.sesta.com`。

- **键入以下命令并按 Enter 键以指定默认域。**
  - **设置默认域：**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DefaultDomainsAndSubdomains "`*DOMAIN_NAME*`"`
  - **获取默认域：**

    PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a DefaultDomainsAndSubdomains`

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

▼ 管理 MIME 推测

重写器根据页面的 MIME 类型选择解析器。某些 Web 服务器（如 WebLogic 和 Oracle）不发送 MIME 类型。要回避这个问题，可通过在“将解析器映射至 URI”列表框中添加数据来启用 MIME 推测功能。

- **键入以下命令并按 Enter 键以管理 MIME 推测。**
  - **启用 MIME 推测：**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableMIMEGuessing true`

- **禁用 MIME 推测：**

  PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableMIMEGuessing false`

- **获取值：**

  PS_INSTALL_DIR`/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableMIMEGuessing`

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

▼ 创建要解析的 URI 映射列表

如果启用了“MIME 推测”复选框，并且服务器尚未发送 MIME 类型，可使用该列表框将解析器映射到 URI。

多个 URI 以分号进行分隔。

例如，HTML=*.html; *.htm;*Servlet。这意味着会使用“HTML 重写器”来重写具有 html、htm 或 Servlet 扩展名的任何页的内容。

- **键入以下命令并按 Enter 键以创建要解析的 URI 映射列表。**

  - **设置要解析的 URI 映射列表：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a MIMEMap`

  - **添加到现有列表：**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a MIMEMap -A` *LIST*

  - **从现有列表中删除：**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a MIMEMap -E` *LIST*

  - **获取现有列表：**

    PS_INSTALL_DIR `/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME*`-a MIMEMap`

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”

▼ 管理屏蔽

屏蔽功能允许重写器重写 URI 以便使人们看不到页的内联网 URL。

- **键入以下命令并按 Enter 键以管理屏蔽。**
  - **启用屏蔽：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableObfuscation true`
  - **禁用屏蔽：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableObfuscation false`
  - **获取值：**

    PS_INSTALL_DIR`/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a EnableObfuscation`

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

▼ 指定屏蔽种子字符串

种子字符串用于屏蔽 URI。该字符串由屏蔽算法生成。

注 – 如果该种子字符串已更改或是重启了网关，则可能无法为屏蔽后的 URI 加书签。

- **键入以下命令并按 Enter 键以指定屏蔽种子字符串。**
  - **设置屏蔽种子字符串：**

    PS_INSTALL_DIR `/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a ObfuscationSecretKey` *SECRET_KEY*

- **获取值：**

  PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a ObfuscationSecretKey

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

▼ 创建禁止屏蔽的 URI 列表

一些应用程序（如 applet）需要一个 Internet URI，而且不能对其进行屏蔽。要指定这些应用程序，可将 URI 添加到列表框中。

例如，当您将 */Applet/Param** 添加到列表框中后，如果内容 URI http://abc.com/Applet/Param1.html 与规则集中的规则匹配，则不会屏蔽该 URL。

注 – 如果有多个 URI，请使用空格分隔每个 URI。

- **键入以下命令并按 Enter 键以创建禁止屏蔽的 URI 列表。**

  - **设置禁止屏蔽的 URI 列表：**

    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a NotToObscureURIList *LIST_OF_URI*

  - **添加到现有列表：**

    PS_INSTALL_DIR /bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a NotToObscureURIList -A *LIST_OF_URI*

  - **从现有列表中删除：**

    PS_INSTALL_DIR /bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a NotToObscureURIList -E *LIST_OF_URI*

  - **获取现有值：**

    PS_INSTALL_DIR /bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a NotToObscureURIList

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

▼ 使网关协议与原始 URI 协议相同

当网关同时运行于 HTTP 和 HTTPS 模式下时，您可以允许重写器使用一致的协议来访问 HTML 内容中引用的资源。

例如，如果原始 URL 是 `http://intranet.com/Public.html`，则添加 http 网关。如果原始 URL 是 `https://intranet.com/Public.html`，则添加 https 网关。

注 – 这只适用于静态 URI，不适用于 Javascript 中生成的动态 URI。

- **键入以下命令并按 Enter 键以使网关协议与原始 URI 协议一致。**

  - **启用：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a UseConsistentProtocolForGateway true`

  - **禁用：**

    PS_INSTALL_DIR`/bin/psadmin set-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a UseConsistentProtocolForGateway false`

  - **获取值：**

    PS_INSTALL_DIR`/bin/psadmin get-attribute -u amadmin -f` *PASSWORD_FILE* `-m gateway --gateway-profile` *PROFILE_NAME* `-a UseConsistentProtocolForGateway`

更多信息 **另请参见**

《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin set-attribute”和《Sun Java System Portal Server 7.2 Command-Line Reference》中的“psadmin get-attribute”

◆ ◆ ◆ 第 9 章

9 在网关服务中配置重写器

突出显示此处的内容。

本章包括以下各节：

有关重写器规则的详细信息，参见第 66 页中的 “定义基于语言的规则”

有关重写器问题的详细信息，参见第 88 页中的 “使用调试日志排除故障”。

有关重写器示例，参见第 91 页中的 “工作示例”。

创建 URI 到规则集的映射列表

创建了规则集之后，可使用 “将 URI 映射至规则集” 字段将某个域与此规则集相关联。默认情况下，会将以下两个条目添加到 “将 URI 映射至规则集” 字段中：

- `*://*.Sun.COM/portal/*|default_gateway_ruleset`

  其中，`sun.com` 是门户的安装域，/portal 是门户的安装环境
- `*|generic_ruleset`

此条目表示，对于域为 `sun.com` 的门户目录中的所有页，都会应用 `default_gateway_ruleset`。对于其他所有页，将会应用一般规则集。`default_gateway_ruleset` 和 `generic_ruleset` 是预先封装的规则集。

注 – 对于标准 Portal 桌面上出现的所有内容，不管内容取自何处，均会使用 `default_gateway_ruleset` 的规则集。

例如，假定将标准 Portal 桌面配置为凑集来自 URL `yahoo.com` 的内容。Portal Server 位于 `sesta.com` 中。`sesta.com` 的规则集将会应用至获取的内容。

注 – 为其指定规则集的域必须在“域和子域代理”列表中列出。

在语法中使用通配符

可以在规则集中使用星号来映射全限定 URI 或部分 URI。

例如，可以将 `java_index_page_ruleset` 应用于 `index.html` 页，如下所示：

```
www.sun.com/java/index.html/java_index_page_ruleset
```

也可以将 java 目录中的所有页应用于 `java_directory_ruleset`，如下所示：

```
www.sun.com/java/* /java_directory_ruleset
```

在网关服务中配置重写器

使用“重写器”选项卡下的网关服务，可以在两类（“基本”和“高级“）范围内执行下列任务：

基本任务

- 第 180 页中的 “允许网关重写所有 URL”
- 第 181 页中的 “指定禁止重写的 URI”
- 第 181 页中的 “将 URI 映射至规则集”
- 第 182 页中的 “指定 MIME 映射”
- 第 183 页中的 “指定默认域”

▼ 允许网关重写所有 URL

如果您启用了网关服务中的“启用全部 URL 重写”选项，重写器会重写任何 URL，而不检查“域和子域代理”列表中的条目。“域和子域代理”列表中的条目将被忽略。

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后选择您要修改其属性的网关配置文件。**

3 选择“ 重写器 ”选项卡。

4 在“ 基本选项 ”下，选中“ 启用所有 URI 的重写 ”复选框以使网关能够重写所有 URL。

5 单击“ 保存 ”以完成修改。

6 从终端窗口中重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ 指定禁止重写的 URI

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“ Secure Remote Access ”选项卡，然后选择您要设置其属性的网关配置文件。

3 选择“ 重写器 ”选项卡。

4 在“ 基本选项 ”下，于“ 添加 ”文本字段中输入 URI，然后单击“ 添加 ”。
URI 值会显示在“禁止重写的 URI”框中。

注 – 向该列表中添加 #* 可重写 URI（即使规则集中包含 href 规则）。

---

5 单击“ 保存 ”以完成修改。

6 从终端窗口中重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ 将 URI 映射至规则集

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“ Secure Remote Access ”选项卡，然后选择您要设置其属性的网关配置文件。

3 选择“ 重写器 ”选项卡。

4 在“ 重写器选项 ”下，单击“ 将 URI 映射至规则集 ”，然后单击“ 添加行 ”。

5 **在 URI 字段中输入所需的域或主机名，然后在“规则集”字段中输入该域相应的规则集。**

会将此条目添加到“将 URI 映射至规则集”列表中。指定域或主机名以及规则集时采用的格式如下：

```
domain name|ruleset name
```

例如：

```
eng.sesta.com|default
```

6 **单击“保存”以完成操作。**

7 **从终端窗口中重新启动网关：**

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ 指定 MIME 映射

重写器有四个不同的解析器，用于根据内容类型对 Web 页进行解析：HTML、JAVASCRIPT、CSS 和 XML。默认情况下，这些解析器与常见的 MIME 类型相关联。您可以在网关服务的“将解析器映射至 MIME 类型”字段中，将新的 MIME 类型与这些解析器相关联。这会将重写器功能扩展到其他 MIME 类型。

使用分号或逗号分隔多个条目（“;”或“,”）。例如：

```
HTML=text/html;text/htm;text/x-component;text/wml; text/vnl/wap.wml
```

它表示会将含有上述 MIME 的任何内容发送到 HTML 重写器，并且会应用 HTML 规则来重写这些 URL。

提示 – 从 MIME 映射列表中删除不必要的解析器可以提高操作速度。例如，如果您确信来自某个内联网的内容不会含有任何 JavaScript，便可从 MIME 映射列表中删除 JAVASCRIPT 条目。

---

1 **以管理员身份登录到 Portal Server 管理控制台。**

2 **选择“Secure Remote Access”选项卡，然后选择您要设置其属性的网关配置文件。**

3 **选择“重写器”选项卡。**

4 **在“重写器选项”下，单击“将解析器映射至 MIME 类型”。**

以 `HTML=text/html;text/htm` 格式指定该条目

5 **单击“添加行”以将该条目添加到列表中。在“MIME 类型”字段中，输入解析器值以及要映射到的相应 MIME 值。**

6 单击“保存”以完成修改。

7 从终端窗口中重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

▼ 指定默认域

当 URL 仅包含没有域和子域的主机名时，默认的域和子域会非常有用。在这种情况下，网关将假定主机名在默认的域和子域中，并进行相应处理。

例如，如果 URL 中的主机名为 `host1`，并且将默认的域和子域指定为 `red.sesta.com`，则主机名会被解析为 `host1.red.sesta.com`。

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“Secure Remote Access”选项卡，然后选择您要设置其属性的网关配置文件。

3 选择“部署”选项卡。

4 在“域和子域的代理”字段中，键入所需的域名（不包括代理）。

5 单击“保存”以完成修改。

6 从终端窗口中重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f  <password file> –N <profile name>– t  <gateway>
```

◆ ◆ ◆ 第 10 章

10 使用证书

本章介绍证书管理并说明如何安装自签名证书和来自“证书授权机构”的证书。

本章说明以下主题：

SSL 证书简介

Sun Java System Portal Server Secure Remote Access 软件为远程用户提供基于证书的验证。SRA 使用“安全套接字层”(SSL) 来实现安全通信。SSL 协议可在两台机器之间实现安全通信。

SSL 证书使用公共和私有密钥对提供加密和解密功能。

证书的两种类型为：

- 自签名证书（也称为根 CA 证书）
- 由证书授权机构 (CA) 签发的证书

默认情况下，当安装网关时，将会生成并安装自签名证书。

在安装后，您可随时生成、获取或替换证书。

SRA 还支持用“个人数字证书”(Personal Digital Certificates, PDC) 进行客户机验证。PDC 是一个通过 SSL 客户机验证来验证用户的机制。进行 SSL 客户机验证时，SSL 信号交换将在网关结束。网关提取用户的 PDC 并将其传递给已验证的服务器。该服务器使用 PDC 验证用户。要将 PDC 与验证链一起配置，参见第 53 页中的 “使用验证链”。

SRA 提供了一个名为 `certadmin` 的工具，可使用它来管理 SSL 证书。参见 第 191 页中的 “certadmin 脚本”。

注 – 在 SSL 应用程序中，证书弹出式窗口很常见。建议用户接受警告并继续进行。

证书文件

证书相关文件位于 `/etc/opt/SUNWportal/cert/` *gateway-profile-name*。默认情况下，此目录包含 5 个文件。

第 186 页中的 “证书文件”列出了这些文件及其说明。

表 10–1 证书文件

文件名	类型	描述
`cert8.db`、`key3.db`、`secmod.db`	二进制	包含证书、密钥和加密模块的数据。 可使用 `certadmin` 脚本处理。 如有必要，可在 Portal Server 主机和网关组件或网关之间共享这些文件。
`.jsspass`	隐藏文本文件	包含 SRA 密钥数据库的加密密码。
`.nickname`	隐藏文本文件	以 *token-name:certificate-name* 格式存储网关需要使用的令牌和证书的名称。 如果您使用的是默认令牌（默认内部软件加密模块上的令牌），则省略令牌名。多数情况下，`.nickname` 文件仅存储证书名。 作为管理员，您可在此文件中修改证书名。您指定的证书此时由网关使用。

证书的信任属性

证书的信任属性表明以下信息：

- 该证书（如客户机或服务器证书）是否由“委托 CA”发放。
- 证书（如根证书）是否可受委托作为服务器或客户机证书的签发人。

按如下顺序表示每一证书的三种可用委托类别：“SSL、电子邮件、对象签名”。对于网关，仅第一种类别有用。在每一类别位置中，将使用多个或不使用任何信任属性代码。

类别的属性代码以逗号分隔，并且整个属性组由引号括在其中。例如，在网关安装期间生成并安装的自签名证书被标记为 "u,u,u"，它表示该证书是服务器证书（用户证书），而不是根 CA 证书。

第 187 页中的 “证书的信任属性”列出了可能的属性值及每个值的含义。

表 10–2 证书的信任属性

属性	描述
p	有效同级
P	委托同级（暗示 p）
c	有效 CA
T	签发客户机证书的委托 CA（暗示 c）
C	签发服务器证书的委托 CA（仅限 SSL）（暗示 c）
u	证书可用于验证或签名
w	发送警告（当证书在该环境中使用时，与其他属性一起使用以包括警告）

CA 信任属性

证书数据库中包括大多数众所周知的公共 CA。有关修改公共 CA 的信任属性的信息，参见第 197 页中的 “修改证书的信任属性”。

第 187 页中的 “CA 信任属性”列出了最常见“证书授权机构”及其信任属性。

表 10–3 公共证书授权机构

证书授权机构名	信任属性
Verisign/RSA Secure Server CA	CPp,CPp,CPp

表 10–3 公共证书授权机构 （续）

VeriSign Class 4 Primary CA	CPp,CPp,CPp
GTE CyberTrust Root CA	CPp,CPp,CPp
GTE CyberTrust Global Root	CPp,CPp,CPp
GTE CyberTrust Root 5	CPp,CPp,CPp
GTE CyberTrust Japan Root CA	CPp,CPp,CPp
GTE CyberTrust Japan Secure Server CA	CPp,CPp,CPp
Thawte Personal Basic CA	CPp,CPp,CPp
Thawte Personal Premium CA	CPp,CPp,CPp
Thawte Personal Freemail CA	CPp,CPp,CPp
Thawte Server CA	CPp,CPp,CPp
Thawte Premium Server CA	CPp,CPp,CPp
American Express CA	CPp,CPp,CPp
American Express Global CA	CPp,CPp,CPp
Equifax Premium CA	CPp,CPp,CPp
Equifax Secure CA	CPp,CPp,CPp
BelSign Object Publishing CA	CPp,CPp,CPp
BelSign Secure Server CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 0 CA	CPp,CPp,CPp
C TrustCenter, Germany, Class 1 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 2 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 3 CA	CPp,CPp,CPp
TC TrustCenter, Germany, Class 4 CA	CPp,CPp,CPp
ABAecom (sub., Am. Bankers Assn.) Root CA	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 1	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 3	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 2	CPp,CPp,CPp
Digital Signature Trust Co. Global CA 4	CPp,CPp,CPp
Deutsche Telekom AG Root CA	CPp,CPp,CPp
Verisign Class 1 Public Primary Certification Authority	CPp,CPp,CPp

表 10–3 公共证书授权机构 （续）

Verisign Class 2 Public Primary Certification Authority	CPp,CPp,CPp
Verisign Class 3 Public Primary Certification Authority	CPp,CPp,CPp
Verisign Class 1 Public Primary Certification Authority - G2	CPp,CPp,CPp
Verisign Class 2 Public Primary Certification Authority - G2	CPp,CPp,CPp
Verisign Class 3 Public Primary Certification Authority - G2	CPp,CPp,CPp
Verisign Class 4 Public Primary Certification Authority - G2	CPp,CPp,CPp
GlobalSign Root CA	CPp,CPp,CPp
GlobalSign Partners CA	CPp,CPp,CPp
GlobalSign Primary Class 1 CA	CPp,CPp,CPp
GlobalSign Primary Class 2 CA	CPp,CPp,CPp
GlobalSign Primary Class 3 CA	CPp,CPp,CPp
ValiCert Class 1 VA	CPp,CPp,CPp
ValiCert Class 2 VA	CPp,CPp,CPp
ValiCert Class 3 VA	CPp,CPp,CPp
Thawte Universal CA Root	CPp,CPp,CPp
Verisign Class 1 Public Primary Certification Authority - G3	CPp,CPp,CPp
Verisign Class 2 Public Primary Certification Authority - G3	CPp,CPp,CPp
Verisign Class 3 Public Primary Certification Authority - G3	CPp,CPp,CPp
Verisign Class 4 Public Primary Certification Authority - G3	CPp,CPp,CPp
Entrust.net Secure Server CA	CPp,CPp,CPp
Entrust.net Secure Personal CA	CPp,CPp,CPp
Entrust.net Premium 2048 Secure Server CA	CPp,CPp,CPp
ValiCert OCSP Responder	CPp,CPp,CPp
Baltimore CyberTrust Code Signing Root	CPp,CPp,CPp
Baltimore CyberTrust Root	CPp,CPp,CPp
Baltimore CyberTrust Mobile Commerce Root	CPp,CPp,CPp
Equifax Secure Global eBusiness CA	CPp,CPp,CPp
Equifax Secure eBusiness CA 1	CPp,CPp,CPp
Equifax Secure eBusiness CA 2	CPp,CPp,CPp

表 10–3 公共证书授权机构 （续）

Visa International Global Root 1	CPp,CPp,CPp
Visa International Global Root 2	CPp,CPp,CPp
Visa International Global Root 3	CPp,CPp,CPp
Visa International Global Root 4	CPp,CPp,CPp
Visa International Global Root 5	CPp,CPp,CPp
beTRUSTed Root CA	CPp,CPp,CPp
Xcert Root CA	CPp,CPp,CPp
Xcert Root CA 1024	CPp,CPp,CPp
Xcert Root CA v1	CPp,CPp,CPp
Xcert Root CA v1 1024	CPp,CPp,CPp
Xcert EZ	CPp,CPp,CPp
CertEngine CA	CPp,CPp,CPp
BankEngine CA	CPp,CPp,CPp
FortEngine CA	CPp,CPp,CPp
MailEngine CA	CPp,CPp,CPp
TraderEngine CA	CPp,CPp,CPp
USPS Root	CPp,CPp,CPp
USPS Production 1	CPp,CPp,CPp
AddTrust Non-Validated Services Root	CPp,CPp,CPp
AddTrust External Root	CPp,CPp,CPp
AddTrust Public Services Root	CPp,CPp,CPp
AddTrust Qualified Certificates Root	CPp,CPp,CPp
Verisign Class 1 Public Primary OCSP Responder	CPp,CPp,CPp
Verisign Class 2 Public Primary OCSP Responder	CPp,CPp,CPp
Verisign Class 3 Public Primary OCSP Responder	CPp,CPp,CPp
Verisign Secure Server OCSP Responder	CPp,CPp,CPp
Verisign Time Stamping Authority CA	CPp,CPp,CPp
Thawte Time Stamping CA	CPp,CPp,CPp
E-Certify CA	CPp,CPp,CPp

表 10–3 公共证书授权机构 （续）

E-Certify RA	CPp,CPp,CPp
Entrust.net Global Secure Server CA	CPp,CPp,CPp
Entrust.net Global Secure Personal CA	CPp,CPp,CPp

certadmin 脚本

您可使用 certadmin 脚本完成以下证书管理任务：

生成自签名证书

您需要为每个服务器与网关之间的 SSL 通信生成证书。

▼ 在安装后生成自签名证书

1 **以超级用户身份，在您要为其生成证书的网关机器上运行 certadmin 脚本：**

portal-server-install-root/SUNWportal/bin/certadmin -n *gateway-profile-name*

显示证书管理菜单。

```
1) Generate Self-Signed Certificate
2) Generate Certificate Signing Request (CSR)
3) Add Root CA Certificate
4) Install Certificate From Certificate Authority (CA)
5) Delete Certificate
6) Modify Trust Attributes of Certificate (e.g., for PDC)
7) List Root CA Certificates
8) List All Certificates
9) Print Certificate Content
10) Quit
choice: [10]
1
```

2 **选择证书管理菜单上的选项 1。**

证书管理脚本询问您是否要保留现有数据库文件。

3 **输入组织特定信息、令牌名和证书名。**

注 – 对于通配符证书，在主机的全限定 DNS 名中指定一个 `*`。例如，如果主机的全限定 DNS 名为 `abc.sesta.com`，则将其指定为 `*.sesta.com`。现在，生成的证书对 `sesta.com` 域中的所有主机名均有效。

```
What is the fully-qualified DNS name of this host? [host_name.domain_name]
What is the name of your organization (ex: Company)? []
What is the name of your organizational unit (ex: division)? []
What is the name of your City or Locality? []
What is the name (no abbreviation please) of your State or Province? []
What is the two-letter country code for this unit? []
Token name is needed only if you are not using the default internal
(software) cryptographic module, for example, if you want to use a crypto card
(Token names could be listed using:
modutil -dbdir /etc/opt/SUNWportal/cert/gateway-profile-name -list);
Otherwise, just hit Return below.
Please enter the token name. []
Enter the name you like for this certificate?
Enter the validity period for the certificate (months) [6]
A self-signed certificate is generated and the prompt returns.
```

令牌名（默认值为空）和证书名存储在 `/etc/opt/SUNWportal/cert/`*gateway-profile-name* 下面的 `.nickname` 文件中。

4 **重新启动网关以使证书生效：**

`./psadmin start-sra-instance -u amadmin -f` *passwordfile* `-N` *profilename* `-t gateway`

生成证书签名请求 (CSR)

需先生成包含 CA 所需信息的证书签名请求，方可从 CA 定购证书。

▼ 生成 CSR

1 **以超级用户身份，运行 `certadmin` 脚本：**

portal-server-install-root`/SUNWportal/bin/certadmin -n` *gateway-profile-name*

显示证书管理菜单。

```
1) Generate Self-Signed Certificate
2) Generate Certificate Signing Request (CSR)
```

```
3) Add Root CA Certificate
4) Install Certificate From Certificate Authority (CA)
5) Delete Certificate
6) Modify Trust Attributes of Certificate (e.g., for PDC)
7) List Root CA Certificates
8) List All Certificates
9) Print Certificate Content
10) Quit
choice: [10]
2
```

2 选择证书管理菜单上的选项 2。

脚本会提示您输入组织特定信息、令牌名以及 Web 站点管理员的电子邮件和电话号码。

确保指定主机的全限定 DNS 名。

```
What is the fully-qualified DNS name of this host? [snape.sesta.com]
What is the name of your organization (ex: Company)? []
What is the name of your organizational unit (ex: division)? []
What is the name of your City or Locality? []
What is the name (no abbreviation please) of your State or Province? []
What is the two-letter country code for this unit? []
Token name is needed only if you are not using the default internal
(software) cryptographic module,
for example, if you want to use a crypto card
(Token names could be listed using:
modutil -dbdir /etc/opt/SUNWportal/cert -list);
Otherwise, just hit Return below.
Please enter the token name []
Now input some contact information for
the webmaster of the machine that the certificate
is to be generated for.
What is the email address of the admin/webmaster for this server [] ?
What is the phone number of the admin/webmaster for this server [] ?
```

3 键入全部所需信息。

注 – 请勿将 Web 站点管理员的电子邮件和电话号码留为空白。此信息对于获取有效 CSR 是必需的。

会产生 CSR 并将其存储在 *portal-server-install-root* `/SUNWportal/bin/csr.hostname.datetimestamp` 文件中。CSR 也会打印在屏幕上。当您从 CA 定购证书时，可直接复制并粘贴 CSR。

添加根 CA 证书

如果客户机站点所提交的经 CA 签名的证书不为网关证书数据库所知，则 SSL 握手将会失败。

为防止出现此情况，您需要向证书数据库中添加根 CA 证书。这将确保使 CA 为网关所知。

浏览到 CA 的 Web 站点并获取该 CA 的根证书。当您使用 `certadmin` 脚本时，要指定根 CA 证书的文件名和路径。

▼ 添加根 CA 证书

1 **以超级用户身份，运行 `certadmin` 脚本。**

portal-server-install-root`/SUNWportal/bin/certadmin -n` *gateway-profile-name*

显示证书管理菜单。

```
1) Generate Self-Signed Certificate
2) Generate Certificate Signing Request (CSR)
3) Add Root CA Certificate
4) Install Certificate From Certificate Authority (CA)
5) Delete Certificate
6) Modify Trust Attributes of Certificate (e.g., for PDC)
7) List Root CA Certificates
8) List All Certificates
9) Print Certificate Content
10) Quit
choice: [10]
3
```

2 **选择证书管理菜单上的选项 3。**

3 **输入包含根证书的文件名，并输入证书名。**

根 CA 证书会被添加到证书数据库。

安装来自证书授权机构的 SSL 证书

在安装网关过程中，默认情况下将创建并安装自签名证书。在安装之后，您可随时安装由供应商（提供官方证书授权机构 (CA) 服务）或您公司的 CA 签名的 SSL 证书。

此项任务涉及三个步骤：

- 第 192 页中的 “生成证书签名请求 (CSR)”
- 第 195 页中的 “从 CA 定购证书”

- 第 195 页中的 “安装来自 CA 的证书”

从 CA 定购证书

在生成证书签名请求 (CSR) 后，您需要使用 CSR 从 CA 定购证书。

▼ 从 CA 定购证书

1. **转到证书授权机构的 Web 站点，并定购您的证书。**
2. **提供 CA 所需的 CSR。提供 CA 所需的其他信息（如果需要）。**

   您会收到来自 CA 的证书。将其保存到文件中。在文件中需包括连同证书在内的 "BEGIN CERTIFICATE" 和 "END CERTIFICATE" 行。

   以下示例省略了实际的证书数据。

```
-----BEGIN CERTIFICATE-----
The certificate contents...
----END CERTIFICATE-----
```

安装来自 CA 的证书

使用 `certadmin` 脚本，在 `/etc/opt/SUNWportal/cert/` *gateway-profile-name* 下的本地数据库文件中安装从 CA 获取的证书。

▼ 安装来自 CA 的证书

1. **以超级用户身份，运行 `certadmin` 脚本。**

   portal-server-install-root`/SUNWportal/bin/certadmin -n` *gateway-profile-name*

   显示证书管理菜单。

```
1) Generate Self-Signed Certificate
2) Generate Certificate Signing Request (CSR)
3) Add Root CA Certificate
4) Install Certificate From Certificate Authority (CA)
5) Delete Certificate
6) Modify Trust Attributes of Certificate (e.g., for PDC)
7) List Root CA Certificates
8) List All Certificates
9) Print Certificate Content
10)Quit
choice: [10]
4
```

2 **选择证书管理菜单上的选项 4。**

脚本要求您输入证书文件名、证书名和令牌名。

```
What is the name (including path) of file that contains the certificate?
Please enter the token name you used when creating CSR for this certificate. []
```

3 **提供全部所需的信息。**

证书被安装在 `/etc/opt/SUNWportal/cert/`*gateway-profile-name* 中，并且返回屏幕提示。

4 **重新启动网关以使证书生效：**

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

删除证书

通过使用证书管理脚本可删除证书。

▼ 删除证书

1 **以超级用户身份，运行 `certadmin` 脚本。**

portal-server-install-root`/SUNWportal/bin/certadmin -n`

其中，*gateway-profile-name* 是网关实例名。

显示证书管理菜单。

```
1) Generate Self-Signed Certificate
2) Generate Certificate Signing Request (CSR)
3) Add Root CA Certificate
4) Install Certificate From Certificate Authority (CA)
5) Delete Certificate
6) Modify Trust Attributes of Certificate (e.g., for PDC)
7) List Root CA Certificates
8) List All Certificates
9) Print Certificate Content
10)Quit
choice: [10]
5
```

2 **选择证书管理菜单上的选项 5。**

3 **输入要删除证书的名称。**

修改证书的信任属性

证书的信任属性需要修改的一种情况便是，同时使用客户机验证和网关。PDC（个人数字证书）是客户机验证的一个示例。签发 PDC 的 CA 必须要受网关委托，并且必须将 CA 证书标记为 "T" 以满足 SSL。

如果安装网关与 HTTPS 站点进行通信，则 HTTPS 站点服务器证书的 CA 必须要受网关委托，并且必须将 CA 证书标记为 "C" 以满足 SSL。

▼ 修改证书的信任属性

1 以超级用户身份，运行 certadmin 脚本。

```
gateway-install-root/SUNWportal/bin/certadmin -n
gateway-profile-name
```

其中，*gateway-profile-name* 是网关实例名。

显示证书管理菜单。

```
1) Generate Self-Signed Certificate
2) Generate Certificate Signing Request (CSR)
3) Add Root CA Certificate
4) Install Certificate From Certificate Authority (CA)
5) Delete Certificate
6) Modify Trust Attributes of Certificate (e.g., for PDC)
7) List Root CA Certificates
8) List All Certificates
9) Print Certificate Content
10)Quit
choice: [10]
6
```

2 选择证书管理菜单上的选项 6。

3 输入证书名。例如，Thawte Personal Freemail CA。

```
Please enter the name of the certificate?
Thawte Personal Freemail CA
```

4 输入证书的信任属性。

```
Please enter the trust attribute you want the
certificate to have [CT,CT,CT]
```

证书信任属性将被更改。

列出根 CA 证书

通过使用证书管理脚本可查看所有根 CA 证书。

▼ 查看根 CA 的列表

1 **以超级用户身份，运行 certadmin 脚本。**

portal-server-install-root/SUNWportal/bin/certadmin -n
gateway-profile-name

其中，*gateway-profile-name* 是网关实例名。

显示证书管理菜单。

```
1) Generate Self-Signed Certificate
2) Generate Certificate Signing Request (CSR)
3) Add Root CA Certificate
4) Install Certificate From Certificate Authority (CA)
5) Delete Certificate
6) Modify Trust Attributes of Certificate (e.g., for PDC)
7) List Root CA Certificates
8) List All Certificates
9) Print Certificate Content
10)Quit
choice: [10]
7
```

2 **选择证书管理菜单上的选项 7。**

显示所有根 CA 证书。

列出所有证书

通过使用证书管理脚本可查看所有证书及其相应的信任属性。

▼ 列出所有证书

1 **以超级用户身份，运行 certadmin 脚本。**

portal-server-install-root
/SUNWportal/bin/certadmin -n
gateway-profile-name

其中，*gateway-profile-name* 是网关实例名。

显示证书管理菜单。

```
1) Generate Self-Signed Certificate
2) Generate Certificate Signing Request (CSR)
3) Add Root CA Certificate
4) Install Certificate From Certificate Authority (CA)
5) Delete Certificate
6) Modify Trust Attributes of Certificate (e.g., for PDC)
7) List Root CA Certificates
8) List All Certificates
9) Print Certificate Content
10)Quit
choice: [10]
8
```

2 **选择证书管理菜单上的选项 8。**

显示所有 CA 证书。

打印证书

通过使用证书管理脚本可打印证书。

▼ 打印证书

1 **以超级用户身份，运行 certadmin 脚本。**

portal-server-install-root/SUNWportal/bin/certadmin -n *gateway-profile-name*

其中，*gateway-profile-name* 是网关实例名。

显示证书管理菜单。

```
1) Generate Self-Signed Certificate
2) Generate Certificate Signing Request (CSR)
3) Add Root CA Certificate
4) Install Certificate From Certificate Authority (CA)
5) Delete Certificate
6) Modify Trust Attributes of Certificate (e.g., for PDC)
7) List Root CA Certificates
8) List All Certificates
9) Print Certificate Content
10)Quit
choice: [10]
9
```

2 **选择证书管理菜单上的选项 9。**

3 输入证书名。

◆ ◆ ◆ 第 11 章

11 配置 Netlet

本章说明如何从 Sun Java System Portal Server 管理控制台配置 Netlet 属性。可在组织级别配置的所有属性也可以在用户级别配置。有关组织、角色和用户级别属性的详细信息，参见《Access Manager 管理指南》。

本章包括以下各节：

配置 Netlet 属性

您可以执行以下任务配置 Netlet：

▼ 配置基本属性

1. 以管理员身份登录到 Portal Server 管理控制台。
2. 选择 “Secure Remote Access” 选项卡，然后选择 “Netlet” 选项卡。
3. 从 “选择 DN” 列表中为用户或组织选择一个 DN 或添加一个 DN。
4. 修改以下属性：

属性名称	描述
COS 优先	指定用于确定属性值继承性的值。有关此属性的详细信息，参见《Sun Java System Directory Server 管理指南》。
启动 Netlet 使用	选择 Java Webstart 或 Applet 选项模式来启动 Netlet 服务。
默认回环端口	指定通过 Netlet 下载 applet 时使用的本机上的端口。使用的默认值为 58000，除非在 Netlet 规则中取代了该值。 输入所需的端口号。
保活间隔（秒）	如果客户机是通过 Web 代理连接到网关的，则会因代理超时而断开空闲的 Netlet 连接。要阻止出现该情况，请输入小于代理超时的值。

5 **单击“保存”以完成修改。**

▼ 配置高级属性

1 **以管理员身份登录到 Portal Server 管理控制台。**

2 **选择“Secure Remote Access”选项卡，然后选择“Netlet”选项卡。**

3 **从“选择 DN”列表中为用户或组织选择一个 DN 或添加一个 DN。**

4 **修改以下属性：**

属性名称	描述
门户注销时中止 Netlet	选择“是”以确保在用户从 Portal Server 中注销时终止所有连接。这将确保更高的安全性。默认情况下，此选项已选中。 选择“否”以确保即使在用户已从 Portal Server 桌面中注销后，活动的 Netlet 连接仍会起作用。 **注 –** 当选择“否”选项后，将不允许用户从 Portal Server 中注销后再建立新的 Netlet 连接。而是只保持现有的连接。
连接时重新验证	选择“是”以指定通过 Netlet 下载 Applet 时使用的本机上的端口。默认值为 58000，除非在 Netlet 规则中取代了该值。默认情况下，“否”选项已选中。
显示连接时弹出警告	如果选择“是”，则当其他用户试图通过侦听端口连接到 Netlet 并且用户正使用 Netlet 运行应用程序时，会在用户桌面上显示弹出式警告对话框。默认情况下，“是”选项已选中。
在端口警告对话框中显示复选框	如果选择“是”，则当 Netlet 试图通过本机上可用的端口连接到目标主机时（如果已在管理控制台中启用），会在用户桌面上显示弹出式警告对话框。默认情况下，“是”选项已选中。

属性名称	描述
Netlet 规则	创建全局级别的 Netlet 规则。您新创建的任何组织都会继承这些规则。有关创建、修改和删除 Netlet 规则的详细信息，参见第 203 页中的 “创建、修改或删除 Netlet 规则”
默认本地 VM 密码	从下拉框中选择 Netlet 规则的默认密码。在使用不包含密码的现有规则时，此功能非常有用。有关详细信息，参见第 136 页中的 “向后兼容性” 一节。
默认 Java 插件密码	从下拉框中选择默认的 Java 插件密码。有关支持的密码列表，参见第 135 页中的 “支持的密码” 。
允许/拒绝的主机	选择主机地址复选框，然后根据用户或组织类型选择允许访问的主机，并从下拉框中选择 “允许” 或 “拒绝” 选项。添加新主机： a. 单击 “添加行” 。 b. 输入指定的全限定主机地址，例如：`abc`，请键入 `abc.sesta.com`。 **注 –** 删除现有的主机：从 “主机” 列表中，选择该主机并单击 “删除” 。 您可以定义允许或拒绝特定组织、角色或用户对特定主机的访问。例如，您可以建立一个含有五个主机的 “允许” 列表，使用户可以远程登录到这些主机。可拒绝访问组织内的特定主机。为每一规则指定一个唯一的**本地端口**。 **注 –** 该字段中的星号 (*) 表示指定域中的所有主机都是可以访问的。例如，如果指定 `*.sesta.com`，则用户可以执行 `sesta.com` 域中的所有 Netlet 目标。也可以指定一个含有通配符的 IP 地址，例如 `xxx.xxx.xxx.*`。
访问/拒绝 Netlet 规则	选择 Nelet 规则，然后从下拉框中选择 “允许” 或 “拒绝” 选项。 可为某些组织、角色或用户定义对特定 Netlet 规则的访问。 可拒绝某些组织、角色或用户定义访问特定的 Netlet 规则。 **注 –** 该字段中的星号 (*) 表示所有已定义的 Netlet 规则都可用于所选组织。

5 单击 “ 保存 ” 以完成修改。

▼ 创建、修改或删除 Netlet 规则

还可以在组织、角色或用户级别创建新规则或修改现有规则。您新创建的任何组织都会继承这些规则。

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择 “ Secure Remote Access ” 选项卡，然后选择 “ Netlet ” 选项卡。

3 从 “ 选择 DN ” 列表中为用户或组织选择一个 DN 或添加一个 DN。

4 **在“ 高级 ”>“ Netlet 规则 ”下，单击“ 新建规则 ”。**

- **要删除规则，请选择一项规则，然后单击“ 删除 ”。**
- **要修改规则，请单击规则名称。**
  在 Netlet 页中，按照如下所述步骤修改参数。

5 **在“ 规则名称 ”字段中输入规则名称。**

6 **从可用密码列表中选择“ 其他 ”，然后在“ 加密密码 ”列表下，选择一个或多个加密密码，或选择“ 默认 ”以保留默认的加密密码。**
在使用不包含密码的现有规则时，此功能非常有用。有关信息，参见“向后兼容性”一节。有关密码的更详细信息，参见“指定默认加密密码”。

7 **在“ 远程应用程序 URL ”字段中，输入指向要调用的应用程序的 URL。**

8 **如果需要下载某个 applet，请选中“ 客户机端口 ”复选框。在“ 客户机端口 ”、“ 服务器主机 ”和“ 服务器端口 ”字段中，输入客户机端口号、服务器主机地址和服务器端口号。为每一规则指定一个唯一的本地端口。**
默认情况下，“启用下载 Applet”框已禁用。仅当需要从 Portal Server 主机以外的某个主机下载 applet 时，才会指定 applet 详细信息。有关详细信息，参见第 130 页中的 “从远程主机下载 Applet”。

9 **选中“ 启用扩展会话 ”复选框，确保在此规则相应的 Netlet 会话运行期间延长 Portal Server 会话时间。**

10 **在“ 映射本地端口到目标服务器端口 ”下，执行以下操作：**

a. **在“ 本地端口 ”字段中输入 Netlet 侦听的本地端口。**
对于 FTP 规则，本地端口值必须是 30021。

b. **在“ 目标主机 ”字段中输入条目。**
对于静态规则，输入 Netlet 要连接的目标机器的主机名。对于动态连接，输入“TARGET”。

c. **在“ 目标端口 ”字段中输入目标主机上的端口。**

11 **单击“ 保存 ”以完成修改。**
规则名称会显示在 Netlet 主页中。

Netlet 的代理配置

可以在用户级别配置以下属性：

- 浏览器代理类型
- 浏览器代理主机
- 浏览器代理端口
- 浏览器代理忽略列表

如果您没有在管理控制台中指定这些值以及 Netlet 无法确定浏览器代理的设置，则在首次通过 Netlet 建立连接时，将会请求用户提供该信息。该信息被存储起来并由用户在将来建立连接时使用。

Netlet 无法在下列情况下确定浏览器代理设置：

- 用户使用的是装有 Java 插件（1.4.0 版本以下）的 Internet Explorer 4.x、5.x 或 6.x，已在“Java 插件控制面板”的“代理”选项卡中启用了“使用浏览器设置”选项，并且已在 Internet Explorer 的“局域网设置”对话框的“使用自动配置脚本”字段中指定了一个附加产品或 INS 文件。
- 用户使用的是装有 Java 插件（1.3.1_01 版或更高版本）的 Netscape 6.2，并且已在“Java 插件控制面板”的“代理”选项卡中启用了“使用浏览器设置”选项。

在这两种情况下，Netlet 有可能无法确定浏览器设置，因而会请求用户提供以下信息：

- 浏览器代理类型

  该属性可取的值为“直接”或“手动”。如果用户从下拉列表中选择“直接”，Netlet 会直接连接到网关主机。
- 浏览器代理主机

  指定所需的代理主机，Netlet 需要通过该主机进行连接。
- 浏览器代理端口

  指定代理主机上的端口，Netlet 需要通过该端口进行连接。
- 浏览器代理忽略列表（以逗号分隔）

  指定不想让 Netlet 通过代理进行连接的主机。该列表可以包含多个以逗号分隔的主机名。

◆ ◆ ◆ 第 12 章

12 配置具有私有域证书的 Netlet

本章说明如何配置客户机浏览器的 Java 插件，以使 Netlet 能够与 PDC 一起使用。

注 – 仅具有 JSSE 的“虚拟机”(VM) 才支持具有 PDC 的 Netlet。

为 PDC 配置 Netlet

此处应为简介文本。

▼ 为 PDC 配置 Netlet

1. **在 Portal Server 机器中 `/ect/opt/SUNWam/config/AMConfig.properties` 文件的任意位置添加 `com.iplanet.authentication.modules.cert.gwAuthEnable=yes`。**
2. **将需要的证书导入希望启用 PDC 的网关的证书数据库。**
3. **在网关机器上导入根 CA 证书。**
4. **将 CA 证书添加到您的网关配置文件。**

提示 – 创建您自己的网关配置文件以测试 PDC。

执行以下步骤以将证书添加到网关配置文件中。

a. *Gateway Install Directory*`/SUNWportal/bin/certadmin -n gateway profile name`
此时将列出 Certadmin 菜单。

b. **选择选项 3。**

c. 提供证书路径。
此时将显示已添加证书消息。

5 生成一个“证书签名请求”以提交到 CA。
执行以下步骤以生成“证书签名请求”：

a. *Gateway Install Directory*`/SUNWportal/bin/certadmin -n gateway profile name`
此时将列出 Certadmin 菜单。

b. 选择选项 2。

c. 提供问题的相应答案。

d. 在一个文件中保存该请求。

6 将“证书签名请求”提交到 CA 并使其获得批准。

提示 – 在 CA 签名后保存证书签名响应。

---

7 导入 CA 批准的“服务器证书”。
执行以下步骤以导入“服务器证书”：

a. *Gateway Install Directory*`/SUNWportal/bin/certadmin -n gateway profile name`
此时将列出 Certadmin 菜单。

b. 选择选项 4。

c. 提供包含“服务器证书”的文件的位置。

8 将“根 CA 证书”导入到 Portal Server 机器。

- **对于 Application Server，请使用以下命令添加** `root-ca`。

```
./certutil -A -n rootca -t "TCu,TCu,TCuw" -d
/var/opt/SUNWappserver/domains/domain1/config -a -i path to root-ca
```

◆ ◆ ◆ 第 13 章

13 配置 Proxylet

本章说明如何从 Sun Java System Portal Server 管理控制台配置 Proxylet。

本章包含以下各节：

配置 Proxylet 属性

选中“部署”选项下的“自动下载 Proxylet Applet”复选框，可将 Proxylet 配置成在用户登录时自动启动。如果未选中“自动下载 Proxylet”复选框，用户可单击标准 Portal 桌面的 Proxylet 频道中的“启动 Proxylet”链接，以在需要时获取 Proxylet。

▼ 配置 Proxylet 属性

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后选择“ Proxylet ”选项卡。**
3. **从“ 选择 DN ”列表框中选择相应的 DN，或为特定用户或组织添加现有的 DN。**
4. **在 Proxylet 页下，执行以下操作：**

属性名称	描述
COS 优先	从选项列表中选择 Proxylet 通信服务的等级。

属性名称	描述
自动下载 Proxylet Applet	单击“是”以自动将 Proxylet applet 下载到客户机。以下是下载 Proxylet applet 的基本要求： 客户机可以运行服务器应用程序 客户机 Java 版本为 1.4 和更高版本 浏览器为 IE 6.0 sp2 或 Firefox 2.0 正确的浏览器权限
通过 Proxylet 刷新门户	如果想要在启动 Proxylet 后刷新 Portal 桌面，并使通信通过 Proxylet，则单击“是”。如果启用了“在 Proxylet 启动后刷新门户”和“自动下载 Proxylet Applet”，则“应用程序 Url”不会起作用。
启动模式	选择 Java Web Start 或 Applet.
默认 Proxylet Applet 绑定 IP	键入 Proxylet 绑定和用于侦听浏览器请求的 IP 地址。
默认 Proxylet Applet 端口	键入 Proxylet 用于侦听浏览器请求的端口号。
自动代理配置文件位置	键入包含代理设置的配置文件的位置，此代理设置可来自代理自动配置 (Proxy Auto Configuration, PAC) 文件或代理配置列表。

5 **在“ Proxylet 规则 ”选项中，请执行以下操作：**

   a. **指定要通过 Proxylet 服务启动的应用程序的规则。**

   b. **单击“ 添加 ”。**

   c. **在“ 域 ”字段中输入域名，例如 `www.google.com`。**

   d. **输入 Proxylet 要处理的域主机和相应端口号。这可确保 Proxylet 解析 HTTP 请求，并且该请求不通过网关路由。**

6 **单击“ 保存 ”以完成修改。**

将应用程序配置到 Portal 桌面

请求（如 HTTP、FTP 等）会通过 Proxylet 服务。Proxylet 规则使管理员能够根据协议、主机或端口指定到域的映射。使用 Proxylet 规则，您可以在代理自动配置 (Proxy Auto Configuration, PAC) 文件中指定域设置和代理设置。例如，您可以创建规则，使所有 FTP 通信量通过 Netlet 发送，所有 HTTP 通信量通过 Proxylet 发送。您可以配置需要通过 Proxylet 服务提交的预定义应用程序。这可以根据用户或组织首选项完成。为 Proxylet 添加要处理的应用程序后，用户桌面便会变得易于管理并提供更好的性能。

▼ 将应用程序配置到 Portal 桌面

开始之前

- 确保已启用 Proxylet 选项。有关启用 Proxylet 的详细信息，参见“网关配置文件”一章。

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Portal ”选项卡，然后选择要修改的门户实例。**
   将会显示“桌面”页。
3. **从“ 选择 DN ”列表框中选择相应的 DN，或为特定用户或组织添加现有的 DN。**
4. **单击“ 管理容器和频道 ”链接。**
   将会显示“管理容器和频道”页。
5. **从左侧窗格中选择 Proxylet。**
6. **从右侧窗格中选择 `Appurls` 链接。**
7. **在“ 属性 ”向导中，输入应用程序名称和值。根据需要修改应用程序的属性。例如，输入应用程序的相应名称和 http://www.example.com。**
8. **单击“ 关闭 ”以完成操作。**
   现在，用户或在组织级别可以查看 Portal 桌面上的应用程序链接。

在 Java Web Start 或 Applet 模式下启动 Proxylet

您可以在 Java Web Start 或 Applet 模式下从 Portal 桌面启动 Proxylet。

▼ 在 Java Web Start 或 Applet 模式下启动 Proxylet

1. **以 proxylet 用户身份登录到 Portal 桌面。**
2. **在“ 扉页 ”中，转至 Proxylet 频道并单击“ 编辑 ”图标。**
3. **从“ 启动模式 ”列表框中选择“ Java Web Start ”或“ Applet ”选项。**
4. **单击“ 已完成 ”。**
   要调用 Proxylet，请从“Proxylet 频道”中选择应用程序。这会在 Java Web Start 或 Applet 模式下启动应用程序。

- 如果选择了“自动下载”，请单击 Proxylet 频道下的应用程序。
- 在用户首选项的基础上，会根据 Java Web Start 或 Applet 模式的选择来显示 Proxylet 控制台。接受所有证书并继续使用应用程序。

◆ ◆ ◆ 第 14 章

配置 NetFile

本章说明如何从 Sun Java System Portal Server 管理控制台配置 NetFile。

本章包含以下各节：

NetFile 配置任务

本节包含以下任务：

▼ 配置基本选项

1. 以管理员身份登录到 Portal Server 管理控制台。
2. 选择 “Secure Remote Access” 选项卡，然后选择 “Netfile” 选项卡。
3. 从 “选择 DN” 列表中为用户或组织选择一个 DN 或添加一个 DN。
4. 修改以下属性：

属性名称	描述
COS 优先	指定用于确定属性值继承性的值。有关此属性的详细信息，参见《Sun Java System Directory Server 管理指南》。
域/主机首选项	输入 NetFile 联络允许主机时所需的默认域。 仅当用户在使用 NetFile 添加主机时指定的不是全限定主机名时，该默认域值才适用。 **注 –** 确保“默认域”字段不为空，并且其中包含有效的域名。
默认 WINS/DNS 服务器	输入 NetFile 用于访问 Microsoft Windows 主机的 WINS/DNS 服务器主机地址。 **注 –** 用户可以在添加机器时指定不同的值来覆盖该值。
主机侦测顺序	使用“上移”和“下移”按钮指定主机侦测顺序。
通用主机	输入主机名或全限定名称，然后单击“添加”。 如果您提供的主机名与用户配置的主机名匹配，则会合并这两组信息，并且用户指定的值会覆盖您指定的值。 配置一个主机列表，使所有远程 NetFile 用户都可以通过 NetFile 访问该列表中的主机。

注 – 例如，假定您已经配置了 4 个通用主机 - `sesta`、`siroe`、`florizon` 和 `abc`。某位用户配置了 3 个主机，其中两个为 `sesta` 和 `siroe`。在这种冲突情况下，用户指定的值会覆盖管理员指定的值。在用户的 NetFile 中也会列出 `florizon` 和 `abc`，而且用户可对这些主机执行各种操作。即使您已将 `florizon` 列入“拒绝的主机列表”， `florizon` 仍会在用户的 NetFile 中列出，只是不能对 `florizon` 执行任何操作。

主机类型— 如果用户已添加了一个“通用主机”列表中列出的主机，则用户设置优先。如果存在类型冲突，不会为该用户添加管理员所添加的共享。如果用户与管理员所添加的共享相同，则会添加此共享，但用户设置的密码优先。

5 **单击“保存”以完成修改。**

▼ 配置访问权限

1 **以管理员身份登录到 Portal Server 管理控制台。**

2 **选择“Secure Remote Access”选项卡，然后选择“Netfile”选项卡。**

3 **从“选择 DN”列表中为用户或组织选择一个 DN 或添加一个 DN。**

4 单击“ 访问权限 ”并修改以下属性：

属性名称	描述
访问 Windows 主机	选中“允许”复选框以确保用户有访问 Windows 主机的权限。 默认情况下，“允许”复选框已选中。
访问 FTP 主机	选中“允许”复选框以确保用户有访问 FTP 主机的权限。
访问 NFS 主机	选中“允许”复选框以确保用户有访问 NFS 主机的权限。
访问 Netware 主机	选中“允许”复选框以确保用户有访问 Netware 主机的权限。

5 单击“ 保存 ”以完成修改。

▼ 配置主机首选项

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“ Secure Remote Access ”选项卡，然后选择“ Netfile ”选项卡。

3 从“ 选择 DN ”列表中为用户或组织选择一个 DN 或添加一个 DN。

4 默认情况下，由于允许/拒绝主机列表中有 * 条目，所以允许用户通过 NetFile 访问所有主机。如果您想要改变这种情况，可在列表中删除 * 条目，然后只在其中指定用户需要通过 NetFile 来进行访问的那些主机。另一种做法是，保留此处的 * 条目，而在“ 拒绝的主机 ”列表中指定想要拒绝访问的主机。在这种情况下，除了“ 拒绝的主机 ”列表中指定的主机以外，允许访问所有主机。

注 – 如果您拒绝访问某一主机而某位用户已将此主机添加到 NetFile 窗口中，被拒绝的主机仍会在该用户的 NetFile 窗口中显示。但该用户不能对该主机执行任何操作。在 NetFile Java2 中，被拒绝的主机在应用程序中显示时标有红色的叉，指示它们是不可访问的。如果“允许的主机”列表和“拒绝的主机”列表均为空，则不允许访问任何主机。

---

5 单击“ 保存 ”以完成修改。

▼ 配置操作首选项

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后选择“ Netfile ”选项卡。**
3. **从“ 选择 DN ”列表中为用户或组织选择一个 DN 或添加一个 DN。**
4. **修改以下属性：**

属性名称	描述
默认压缩类型	从下拉框中选择 ZIP 或 GZ 作为默认文件压缩格式。
默认压缩级别	从下拉框中选择默认的压缩级别。默认值为 6。
临时目录位置	输入临时文件的位置。如果所指定的临时目录在服务器上不存在，则会创建它。 有些文件操作（例如邮寄文件）需要一个临时目录。默认的临时目录是 `/tmp`。在执行完所需操作后，会删除这些临时文件。 **注 –** 确保运行 Web 服务器的 ID（如 `nobody` 或 `noaccess`）对所指定的目录具有 `rwx` 权限。还要确保此 ID 对于到所需临时目录的完整路径具有 `rx` 权限。 **提示 –** 最好为 NetFile 创建一个单独的临时目录。如果所指定的临时目录对于 Portal Server 的所有模块来说是公用目录，磁盘空间可能很快就会用完。如果临时目录没有空间，NetFile 中的一少部分操作（如邮寄文件）将无法正常进行。
文件上载限制 (MB)	在该字段中输入可上载文件大小的最大值。默认值为 5MB。 当正在上载的文件大小超出此处指定的限制时，将显示一条错误消息并且不会上载该文件。如果输入无效值，NetFile 会将其重置为默认值。可为不同用户指定不同的文件上载大小限制。

属性名称	描述
搜索目录限制	输入单个搜索操作中可搜索的最大目录数。在大量用户同时登录的情况下，此限制有助于减少网络阻塞，提高访问速度。默认值为 100。 假设用户有一个名为 *A* 的目录。同时，假定 *A* 有 100 个子目录。如果指定要搜索的最大目录数为 100，搜索操作在搜遍目录 A 后才会停止。由于在目录 *A* 中已达到 100 这一限制，所以不会继续在用户机器的其他目录中进行搜索。截至达到搜索限制为止所积累的搜索结果会显示给用户，并显示一条错误消息，表明已超出搜索限制。要继续搜索，用户必须在下一个目录处手动重新启动搜索。搜索操作以深度优先方式进行。这就意味着，搜索操作在转移到下一目录之前，会先在用户所选目录的所有子目录中进行。

5 单击“ 保存 ”以完成修改。

▼ 配置操作权限

可允许或拒绝用户从远程主机上执行以下任务。

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“ Secure Remote Access ”选项卡，然后选择“ Netfile ”选项卡。

3 从“ 选择 DN ”列表中为用户或组织选择一个 DN 或添加一个 DN。

4 修改以下属性：

属性名称	描述
文件重命名	选中“允许”复选框以允许用户重命名文件。默认情况下，此选项已选中。
文件/文件夹删除	选择“允许”复选框以允许用户删除文件和文件夹。默认情况下，此选项已选中。
文件上载	选中“允许”复选框以允许用户上载文件。默认情况下，此选项已选中。
文件/文件夹下载	选择“允许”复选框以允许用户下载文件或目录。默认情况下，此选项已选中。
文件搜索	选中“允许”复选框以允许用户执行文件搜索操作。默认情况下，此选项已选中。

属性名称	描述
文件邮寄	选中“允许”复选框以允许用户访问邮件。默认情况下，此选项已选中。
文件压缩	选中“允许”复选框以允许用户选择压缩类型。默认情况下，此选项已选中。
更改用户 ID	选中“允许”复选框以允许用户更改其用户 ID。用户可以使用不同的 ID 连接到使用 NetFile 的主机。 在大型组织中，用户可能会有多个用户 ID。您可能想要限制用户使用单个用户 ID。在这种情况下，您可以禁用“允许更改用户 ID”选项。这将防止特定组织中的所有用户更改其用户 ID，并限制他们只能通过 NetFile 使用单个 ID（桌面登录 ID）连接到主机。在另一种情况下，用户可能在不同的机器上有不同的登录 ID，此时，您或许希望允许用户根据需要更改 ID。
更改 Microsoft Windows 域	选中“允许”复选框以允许用户更改默认的 Microsoft Windows 域主机。默认情况下，此选项已选中。 当用户指定域名后，还需要为该域指定用户名和密码。如果需要使用主机用户名和密码，用户需从“用户域名”字段中删除此域。

注 – 当未选中上述任何选项时，仅当用户再次登录到 Portal Server 桌面后更改才会生效。

5 单击“保存”以完成修改。

◆ ◆ ◆ 第 15 章

15 配置安全套接字层加速器

本章说明如何配置 Sun Java System Portal Server Secure Remote Access 的各种加速器。

本章包含以下各节：

加速器简介

外部加速器是专用的硬件协处理器，用于从服务器的 CPU 卸载安全套接字层 (SSL) 功能，借此释放 CPU 空间以使其执行其他任务，同时提高对 SSL 事务的处理速度。

Sun Crypto Accelerator 1000

Sun™ Crypto Accelerator 1000 (Sun CA1000) 板是一种短 PCI 板，用作加密协处理器以加速公共密钥和对称加密。本产品无外部接口。该板通过内部 PCI 总线接口与主机通信。采用此板卡的目的是针对电子商务应用程序中的安全协议，加速各种在计算上较为密集的加密算法。

许多关键的加密功能，如 RSA [7] 和 Triple-DES (3DES) [8]，都可从应用程序中卸载到 Sun CA1000 并以并行方式执行。这样便可释放中央处理器空间以执行其他任务，同时提高对 SSL 事务的处理速度。

有关步骤，参见第 220 页中的 “配置 Crypto Accelerator 1000”。

启用 Crypto Accelerator 1000

确保已安装了 Portal Server Secure Remote Access，并安装了网关服务器证书（自签名或由任意 CA 所签发）。有关详细信息，参见第 10 章。

第 220 页中的 “启用 Crypto Accelerator 1000” 是一个清单，有助于您在安装 “SSL 加速器” 之前熟悉所需信息，并列出了 Crypto Accelerator 1000 的参数和相应值。

表 15–1 Crypto Accelerator 1000 安装清单

参数	值
SRA 安装基目录	/opt
SRA 证书数据库路径	/etc/opt/SUNWportal/cert/default
SRA 服务器证书昵称	server-cert
领域	sra-keystore
领域用户	crypta

▼ 配置 Crypto Accelerator 1000

1. **按照用户指南中的说明安装硬件。参见：**

   `http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-2450-11.pdf`

2. **从光盘安装以下软件包。**

   `SUNWcrypm`、`SUNWcrypu`、`SUNWcrysu`、`SUNWdcar`、`SUNWcrypr`、`SUNWcrysl`、`SUNWdcamn`、`SUNWdcav`

3. **安装以下修补程序。（您可从 `http://sunsolve.sun.com` 获取这些程序）**

   110383-01、108528-05、112438-01

4. **确保您拥有 `pk12util` 和 `modutil` 工具。**

   这些工具安装在 `/usr/sfw/bin` 下面。如果在 `/usr/sfw/bin` 目录中没有这些工具，则需要从 Sun Java System 分发介质手动添加 SUNWtlsu 软件包：

   ```
   Solaris_[sparc/x86]/Product/shared_components/
   ```

5. **创建插槽文件：**

   ```
   vi /etc/opt/SUNWconn/crypto/slots
   ```

   然后将 "crypta@sra" 作为第一行而且是唯一一行放在文件中。

6 **创建和设置区域。**

a. **以根用户身份登录。**

b. **键入以下命令：**

```
cd /opt/SUNWconn/bin/secadm
```

```
secadm> create realm=sra
```

已成功创建区域 sra。

7 **创建一个用户：**

a. **键入并回应以下命令：**

secadm> `set realm=sra`

secadm{srap}> `su`

secadm{root@sra}> `create user=crypta`

Initial password:

Confirm password:

已成功创建用户 crypta。

8 **以您创建的用户身份登录。**

secadm{root@sra}> `login user=crypta`

Password:

secadm{crypta@sra}> `show key`

不存在此用户的密钥。

9 **载入 Sun Crypto 模块。**

环境变量 `LD_LIBRARY_PATH` 必须指向 `/usr/lib/mps/secv1/`

键入：

```
modutil -dbdir /etc/opt/SUNWportal/cert/default -add "Sun Crypto Module"
-libfile /opt/SUNWconn/crypto/lib/libpkcs11.so
```

使用以下命令验证是否已载入此模块：

```
modutil -list -dbdir /etc/opt/SUNWportal/cert /default
```

10 **将网关证书和密钥导出到“Sun Crypto 模块”中。**

环境变量 `LD_LIBRARY_PATH` 必须指向 `/usr/lib/mps/secv1/`

键入：

```
pk12util -o servercert.p12 -d /etc/opt/SUNWportal/cert/default -n server-cert
pk12util -i servercert.p12 -d /etc/opt/SUNWportal/cert/default -h "crypta@sra"
```

现在，运行显示密钥命令：

```
secadm{crypta@sra}> show key
```

应该可以看到此用户的两个密钥。

11 更改 /etc/opt/SUNWportal/cert/default/.nickname 文件中的昵称。

```
vi /etc/opt/SUNWportal/cert/default/.nickname
```

用 `crypta@sra:server-cert` 替换 `server-cert`

12 启用加速密码。

SUN CA1000 可加速 RSA 功能，但只支持对 DES 和 3DES 密码的加速。

13 修改 `/etc/opt/SUNWportal/platform.conf.`*gateway-profile-nam* 以启用加速器：

```
gateway.enable.accelerator=true
```

14 从终端窗口重新启动网关：

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

注 – 网关会绑定到端口（在配置文件中被称为 https 端口）上的一个普通 ServerSocket（非 SSL）。

不对收到的客户机通信进行任何 SSL 加密或解密操作。此操作会由加速器来执行。

PDC 在此模式下不起作用。

Sun Crypto Accelerator 4000

Sun™ Crypto Accelerator 4000 板是一种基于千兆以太网的网络接口卡，支持 Sun 服务器上的 IPsec 和 SSL（对称和非对称）加密硬件加速。

除了作为用于未加密网络通信的标准千兆以太网网卡之外，该板还包含加密硬件以支持加密 IPsec 通信实现更高的通过量。

Crypto Accelerator 4000 板可同时在硬件和软件上加速加密算法。它也支持密码 DES 和 3DES 的整体加密。

有关步骤，参见第 223 页中的 “配置 Crypto Accelerator 4000” 。

启用 Crypto Accelerator 4000

确保已安装了 SRA，并安装了网关服务器证书（自签名或由任意 CA 所签发）。以下核对表将帮助您在安装 SSL 加速器之前熟悉所需的信息。

第 220 页中的 “启用 Crypto Accelerator 1000” 列出了 Crypto Accelerator 4000 的参数和相应值。

表 15–2 Crypto Accelerator 4000 安装清单

参数	值
Portal Server Secure Remote Access 安装基目录	/opt
SRA 实例	default
SRA 证书数据库路径	/etc/opt/SUNWportal/cert/default
SRA 服务器证书昵称	server-cert
CA4000 密钥库	srap
CA4000 密钥库用户	crypta

▼ 配置 Crypto Accelerator 4000

1. **按用户指南中的说明安装硬件和软件包。参见：**
   http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-2450-11.pdf

2. **安装以下修补程序。（您可从 http://sunsolve.sun.com 处获取这些程序）：** `114795`

3. **确保您拥有 `certutil`、`pk12util` 和 `modutil` 工具。**
   这些工具安装在 `/usr/sfw/bin` 下面
   如果工具不位于 `/usf/sfw/bin` 目录中，则需要
   从 Sun Java System 分发介质手动添加 SUNWtlsu 软件包：
   `Solaris_[sparc/x86]/Product/shared_components/`

4. **初始化该板。**
   运行 `/opt/SUNWconn/bin/vcadm` 工具初始化密码板并设置以下值：
   初始安全主管名：`sec_officer`
   密钥库名称：`sra-keystore`
   以 FIPS 140-2 模式运行：**否**

5 **创建一个用户。**

vcaadm{vca0@localhost, sec_officer}> `create user`

新用户名：`crypta`

输入新的用户密码：

Confirm password:

已成功创建用户 crypta。

6 **将令牌映射到密钥库。**

```
vi /opt/SUNWconn/cryptov2/tokens
```

然后，将 `sra-keystore` 追加到文件中。

7 **启用整体加密。**

```
touch /opt/SUNWconn/cryptov2/sslreg
```

8 **载入 Sun Crypto 模块。**

环境变量 `LD_LIBRARY_PATH` 必须指向 `/usr/lib/mps/secv1/`

键入：

```
modutil -dbdir /etc/opt/SUNWportal/cert/default -add "Sun Crypto Module"
-libfile /opt/SUNWconn/cryptov2/lib/libvpkcs11.so
```

您可使用以下命令检验是否已载入此模块：

```
modutil -list -dbdir /etc/opt/SUNWportal/cert/default
```

9 **将网关证书和密钥导出到“`Sun Crypto` 模块”中。**

环境变量 `LD_LIBRARY_PATH` 必须指向 `/usr/lib/mps/secv1/`

```
pk12util -o servercert.p12 -d /etc/opt/SUNWportal/cert/default -n server-cert
```

```
pk12util -i servercert.p12 -d /etc/opt/SUNWportal/cert/default -h "sra-keystore"
```

您可使用以下命令检验是否已经导出密钥：

```
certutil -K -h "sra-keystore" -d /etc/opt/SUNWportal/cert/default
```

10 **更改 `/etc/opt/SUNWportal/cert/default/.nickname` 文件中的昵称：**

```
vi /etc/opt/SUNWportal/cert/default/.nickname
```

用 `sra-keystore:server-cert` 替换 `server-cert`

11 **启用加速密码。**

12 **从终端窗口重新启动网关：**

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

网关会提示您输入密钥库密码。

为 "sra-keystore":crypta:crytpa-password 输入密码或 Pin

注 – 网关会绑定到端口（在配置文件中被称为 https 端口）上的一个普通 ServerSocket（非 SSL）。

不对收到的客户机通信进行任何 SSL 加密或解密操作。此操作会由加速器来执行。

PDC 在此模式下不起作用。

外部 SSL 设备和代理加速器

外部 SSL 设备可在开放模式下于 Portal Server Secure Remote Access (SRA) 前端运行。它提供了客户机与 SRA 之间的 SSL 链路。

可执行以下任务：

- 第 225 页中的 “启用外部 SSL 设备加速器”
- 第 226 页中的 “配置外部 SSL 设备加速器”

▼ 启用外部 SSL 设备加速器

1 **确保已安装了 SRA，并且网关在开放模式（HTTP 模式）下运行。**

2 **启用 HTTP 连接。**

表格列出外部 SSL 设备和代理加速器的参数和值。

参数	值
SRA 实例	default
网关模式	http
网关端口	880
外部设备/代理端口	443

▼ 配置外部 SSL 设备加速器

1 按用户指南中的说明安装硬件和软件包。

2 安装必需的修补程序（如果有）。

3 配置网关实例以使用 HTTP。

4 在 platform.conf 文件中输入以下值：

```
gateway.enable.customurl=true

gateway.enable.accelerator=true

gateway.httpurl=https:// external-device-URL:port-number
```

5 可以两种方式配置网关通知:

- 当 Access Manager 可在端口 880 联络网关机器时（会话通知采用 HTTP 方式），在 platform.conf 文件中输入值。

```
vi /etc/opt/SUNWportal/platform.conf.default

gateway.protocol=http

gateway.port=880
```

  - 当 Access Manager 可在端口 443 联络外部设备/代理时（会话通知采用 HTTPS 方式），在 platform.conf 文件中输入值。

```
vi /etc/opt/SUNWportal/platform.conf.default

gateway.host=External Device/Proxy Host Name

gateway.protocol=https

gateway.port=443
```

6 确保 SSL 设备/代理就绪并处于运行状态，而且经过配置以便将通信引向网关端口。

7 从终端窗口重新启动网关：

```
./psadmin start-sra-instance -u amadmin -f passwordfile -N profilename -t gateway
```

第 3 部分

管理 Secure Remote Access 服务器

Secure Remote Access 服务器有两个管理界面 :

- Portal Server 管理控制台
- 《Sun Java System Portal Server 7.2 Command-Line Reference》中的第 1 章 “psadmin Utility”中介绍的命令行实用程序

大多数管理任务可通过基于 Web 的 Portal Server 管理控制台执行，用户可通过使用 Web 浏览器从本地或远程访问该控制台。有关详细信息，参见《Sun Java System Portal Server 7.2 管理指南》中的“使用 Portal Server 管理控制台”。

但是，某些任务（如文件修改）必须通过 UNIX 命令行界面进行管理。

◆ ◆ ◆ 第 16 章

管理网关

此处介绍重要内容

管理网关的任务

本节说明管理门户服务器网关的以下任务：

▼ 创建网关配置文件

1 **以管理员身份登录到 Portal Server 管理控制台。**

2 **单击“ Secure Remote Access ”选项卡并单击“ 新建配置文件 ”。**

将显示“新建配置文件”页。

3 **输入新网关配置文件的名称。**

4 **从下拉列表中选择用于创建新配置文件的配置文件。**

默认情况下，您创建的任何新配置文件都基于预封装的默认配置文件。如果已创建一个自定义配置文件，则可以从下拉列表中选择该配置文件。新配置文件会继承所选配置文件的全部属性。

用现有配置文件复制而成的新配置文件沿袭同一端口。更改新配置文件的端口，使其不与现有配置文件冲突。

5 **单击“ 确定 ”。**

新配置文件将被创建并在“配置文件”页中列出。

注意 – 确保更改该实例的端口以使其不与任何已占用的现有端口冲突。

6 **使用 Telnet 连接到需要创建实例的机器。该机器上已启动并运行默认网关实例。**

7 **在立即配置模式下安装 AM-SDK。**

8 **在立即配置模式下或稍后选择配置模式下，使用 UI 安装程序安装网关。**

9 **将 `/opt/SUNWportal/template/sra/GWConfig.properties.template` 文件复制到临时位置。例如，`/tmp`。**

10 **根据需要修改值。**

注 – 该值应与新配置文件的网关实例中的端口号相匹配。

11 **完成后，运行以下命令：**

```
./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file
location>.template -t gateway
```

12 **使用新的网关配置文件名重新启动网关以确保使更改生效：**

`./psadmin start-sra-instance –u amadmin – f` *<password file>* `–N` *<profile name>*`– t` *<gateway>*

有关启动和停止网关的详细信息，参见第 231 页中的 “启动网关实例”。要配置网关，参见第 8 章

▼ 使用同一 LDAP 创建网关实例

1 **使用与第一个网关相同的字符串替换用于加密和解密密码的密钥。**

```
am.encryption.pwd= string_key_specified_in gateway-install
```

2 **替换作为应用程序验证模块共享密钥的密码：**

```
com.iplanet.am.service.secret= string_key_specified_in gateway-install
```

3 **在 `/etc/opt/SUNWam/config/ums` 中，修改 `serverconfig.xml` 中的以下区域，使其与第一个安装的 Portal Server 实例一致：**

```
<DirDN> cn=puser,ou=DSAME Users,dc=sun,dc=net</DirDN>

<DirPassword> string_key_specified_in gateway-install</DirPassword>
```

```
<DirDN>cn=dsameuser,ou=DSAME Users,dc=sun,dc=net</DirDN>

<DirPassword>string_key_specified_in gateway-install </DirPassword>
```

4 **重新启动 Access Manager 服务。**

▼ 启动网关实例

默认情况下，网关是以用户 noaccess 启动的。

1 **安装网关并创建所需的配置文件之后，请运行下面的命令启动网关：**

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t <gateway>`

`default` — 安装期间所创建的默认网关配置文件。可在以后创建自己的配置文件，然后用新的配置文件重新启动网关。参见第 32 页中的 “创建网关配置文件” 。

注 – 以相应的配置文件名更换 <*profile name*> 以启动其他网关实例。

重新启动服务器（在其中配置网关实例的机器）会重新启动所有网关实例。

确保 `/etc/opt/SUNWportal` 目录中没有备份的配置文件。

---

2 **运行以下命令，检查网关是否在指定的端口上运行：**

`netstat -an | grep` *port-number*

默认网关端口为 443。

▼ 停止网关

1 **请使用以下命令停止网关：**

`./psadmin stop-sra-instance –u amadmin – f <password file> –N <profile name> – t <gateway>`

注 – 以相应的配置文件名更换 <*profile name*> 以启动其他网关实例。

---

2 **运行以下命令以验证是否有网关进程仍在运行：**

```
/usr/bin/ps -ef | grep entsys
```

▼ 使用管理控制台启动和停止网关

1. 登录 Portal Server 管理控制台。
2. 选择“ Secure Remote Access ”选项卡。
3. 单击“ 管理实例 ”子菜单。
4. 在“ SRA 代理实例 ”下，选择一个实例。
   - 单击“ 启动 ”启动实例。
   - 单击“ 停止 ”停止实例。

▼ 用不同的配置文件重新启动网关

- **重新启动网关：**

  `./psadmin start-sra-instance –u amadmin – f <`*password file*`> –N <`*profile name*`> – t <`*gateway*`>`

▼ 重新启动网关

- **在终端窗口中，以超级用户身份连接并执行以下步骤之一：**
  - 启动监视程序进程：

    ```
    ./psadmin sra-watchdog -u uid -f password-filename -t instance-type on
    ```

`[--adminuser | -u] uid`	指定管理员的标识名 (DN) 或用户 ID。
`[-passwordfile | -f] password-filename`	指定密码文件中的管理员密码。
`[--type | -t] instance-type`	指定 Secure Remote Access 实例的类型。输入： gateway、nlproxy 或 rwproxy。

    有关监视程序命令的信息，参见 Sun Java System Portal Server Command Line Reference Guide。

    此操作在 crontab 实用程序中创建一个条目，并且监视程序进程当前处于活动状态。监视程序监控正在特定机器上运行的所有网关实例和网关端口，并在网关停机时重新启动它。

▼ 指定虚拟主机

1 **以超级用户身份登录并编辑所需网关实例的** `platform.conf` **文件：**

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

2 **添加下列条目：**

`gateway.virtualhost=`*fully-qualified-gateway-host gateway-ip-address fully-qualified-reverse-proxyhost*

`gateway.enable.customurl=true`（该值默认设置为 false。）

3 **重新启动网关：**

`./psadmin start-sra-instance –u amadmin – f <`*password file*`> –N <`*profile name*`>– t <`*gateway*`>`

如果未指定这些值，则网关以默认值执行正常操作。

▼ 指定代理

1 **在命令行中编辑以下文件：**

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

2 **添加下列条目：**

`http.proxyHost=`*proxy-host*
`http.proxyPort=`*proxy-port*
`http.proxySet=true`

3 **重新启动网关以使用指定的代理处理发往服务器的请求：**

`./psadmin start-sra-instance –u amadmin – f <`*password file*`> –N <`*profile name*`>– t <`*gateway*`>`

▼ 创建 Netlet 代理实例

1 **使用 Telnet 连接到需要创建实例的机器。该机器上已启动并运行默认网关实例。**

2 **将** `/opt/SUNWportal/template/sra/NLPConfig.properties.template` **文件复制到临时位置。例如，**`/tmp`**。**

3 **在新配置文件的文件中根据需要修改相应值。**

4 完成后，运行以下命令：

```
./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file location>.template -t nlproxy
```

5 以所需网关配置文件名启动 Netlet 代理的新实例以确保使更改生效：

```
./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t nlproxy
```

▼ 重新启动 Netlet 代理

- 在终端窗口中，以超级用户身份连接并执行以下步骤之一：

  - 启动监视程序进程：

    ```
    psadmin sra-watchdog -u uid -f password-filename -t instance-type on
    ```

    输入 `nlproxy` 替换 *instance-type*。有关此命令的详细信息，参见 Sun Java Portal Server Command Line Reference Guide。

    此操作在 crontab 实用程序中创建一个条目，并且监视程序进程当前处于活动状态。监视程序监控 Netlet 代理端口，只要它停止运行就重新启动。

  - 手动启动 Netlet 代理：

    ```
    psadmin start-sra-instance -u uid -f password-filename -N sra-instance-name -t instance-type
    ```

    输入 `nlproxy` 替换 *instance-type*。此配置文件名与所需的 Netlet 代理实例相对应。有关此命令的详细信息，参见 Sun Java Portal Server Command Line Reference Guide。

▼ 创建重写器代理实例

1 使用 Telnet 连接到需要创建实例的机器。该机器上已启动并运行默认网关实例。

2 将 `/opt/SUNWportal/template/sra/GWConfig.properties.template` 文件复制到临时位置。例如，`/tmp`。

3 在新配置文件的文件中根据需要修改相应值。

4 完成后，运行以下命令：

```
./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file location>.template -t rwproxy
```

5 **以所需网关配置文件名启动重写器代理的新实例以确保使更改生效：**

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t rwproxy`

▼ 重新启动重写器代理

- **在终端窗口中，以超级用户身份连接并执行以下步骤之一：**

  - **启动监视程序进程：**

    `psadmin sra-watchdog -u uid -f password-filename -t instance-type on`

    输入 `rwproxy` 替换 *instance-type*。有关此命令的详细信息，参见 Sun Java Portal Server Command Line Reference Guide。

    此操作在 crontab 实用程序中创建一个条目，并且监视程序进程当前处于活动状态。监视程序监控重写器代理端口，只要它停止运行就重新启动。

  - **手动启动重写器代理：**

    `start-sra-instance -u uid -f password-filename -N sra-instance-name -t instance-type`

    输入 `rwproxy` 替换 *instance-type*。此配置文件名与所需的重写器代理实例相对应。有关此命令的详细信息，参见 Sun Java Portal Server Command Line Reference Guide。

▼ 启用反向代理

1 **以超级用户身份登录并编辑所需网关实例的 `platform.conf` 文件：**

`/etc/opt/SUNWportal/platform.conf.` *gateway-profile-name*

2 **添加下列条目：**

`gateway.virtualhost=` *fully-qualified-gateway-host gateway-ip-address fully-qualified-reverse-proxyhost*

`gateway.enable.customurl=true`（该值默认设置为 false。）

`gateway.httpurl=` *http reverse-proxy-URL*

`gateway.httpsurl=`*https reverse-proxy-URL*

`gateway.httpurl` 用于重写对端口处所收到请求的响应，该端口在网关配置文件中列为 HTTP 端口。

`gateway.httpsurl` 用于重写对端口处所收到请求的响应，该端口在网关配置文件中列为 HTTPS 端口。

3 **重新启动网关：**

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name> – t <gateway>`

如果未指定这些值，则网关以默认值执行正常操作。

▼ 向现有 PDC 实例添加验证模块

1 **以管理员身份登录到 Access Manager 管理控制台。**

2 **选择所需的组织。**

3 **在“ 查看 ”下拉框中选择“ 服务 ”。**
将显示这些服务。

4 **单击“ 验证配置 ”。**
将显示“服务实例列表”。

5 **单击 Gatewaypdc。**
将显示 Gatewaypdc 属性页。

6 **单击“ 编辑 ”。**
将显示“添加模块”页面。

7 **选择“ 模块名称 ”并将“ 标志 ”设置为“ 必填 ”。**

8 **单击“ 确定 ”。**

9 **添加一个或多个模块之后，单击“ 保存 ”。**

10 **单击 `gatewaypdc` 属性页中的“ 保存 ”。**

11 **重新启动网关以使更改生效：**

gateway-install-location`/SUNWportal/bin/psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>– t <gateway>`

▼ 禁用浏览器高速缓存

1 **以超级用户身份登录并编辑所需网关实例的 `platform.conf` 文件：**

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

2 **编辑以下行：**

```
gateway.allow.client.caching=true
```

该值默认设置为 `true`。将该值更改为 `false` 可禁用客户机的浏览器高速缓存。

3 **重新启动网关：**

`./psadmin start-sra-instance –u amadmin – f` <*password file*> `–N` <*profile name*> `– t` <*gateway*>

▼ 共享 LDAP 目录

1 **修改 `AMConfig.properties` 中的以下区域，使其与第一个安装的 Portal Server 和 Access Manager 服务器实例同步：**

将用于加密和解密密码的密钥。
`am.encryption.pwd=t/vnY9Uqjf12NbFywKuAaaHibwlDFNLO <==` REPLACE THIS STRING WITH THE ONE FROM FIRST PORTAL INSTALL

/* The following key is the shared secret for application auth module */
`com.iplanet.am.service.secret=AQICxIPLNc0WWQRVlYZN0PnKgyvq3gTU8JA9 <==` REPLACE THIS STRING WITH THE ONE FROM FIRST PORTAL INSTALL

2 **在 `/etc/opt/SUNWam/config/ums` 中，修改 `serverconfig.xml` 中的以下区域，以便与第一个安装的 Portal Server 和 Access Manager 服务器实例不同步：**

```
<DirDN>
    cn=puser,ou=DSAME Users,dc=sun,dc=net
</DirDN>
    <DirPassword>
         AQICxIPLNc0WWQT22gQnGgnCp9rUf+FuaqpY
         <==  REPLACE THIS STRING WITH THE ONE FROM FIRST PORTAL INSTALL
  </DirPassword>

<DirDN>
   cn=dsameuser,ou=DSAME Users,dc=sun,dc=net
</DirDN>
     <DirPassword>
          AQICxIPLNc0WWQT22gQnGgnCp9rUf+FuaqpY
          <==  REPLACE THIS STRING WITH THE ONE FROM FIRST PORTAL INSTALL
     </DirPassword>
```

3 **重新启动 Access Manager 服务。**

◆ ◆ ◆ 第 17 章

17 联合管理方案

以下主题在讨论之列：

使用联合管理

“联合管理”使用户能够将各自的本地身份聚合成为一个网络身份。“联合管理”通过该网络身份允许用户在一个服务提供者的站点上登录即可访问其他服务提供者的站点，而不必重新进行身份验证。这称为单点登录。

在 Portal Server 上，可在开放模式和安全模式中配置联合管理。Portal Server 管理指南描述了如何在开放模式中配置联合管理。在安全模式下使用 Portal Server Secure Remote Access 服务器配置“联合管理”之前，确保其工作于开放模式。如果要使用户既在开放模式中、又在安全模式中使用来自同一浏览器的“联合管理”，则他们必须清除 cookie 并从浏览器进行高速缓存。

有关“联合管理”的详细信息，参见 Access Manager 联合管理指南。

联合管理方案

初始服务提供者对用户进行验证。服务提供者是提供网络服务的商业性或非盈利性组织。这一广泛的范畴可以包括网络门户、零售商、运输供应商、金融机构、娱乐公司、图书馆、高等院校和政府机构。

服务提供者使用 cookie 存储用户在客户机浏览器中的会话信息。cookie 还包括用户的身份认证提供者。

身份认证提供者是专门提供验证服务的服务提供者。作为验证的管理服务，他们也维护和管理身份信息。由身份认证提供者完成的验证得到了与其联合的所有服务提供者的认同。

当用户试图访问不属于该身份认证提供者的服务时，身份认证提供者将 cookie 发往相应的服务提供者。该服务提供者随后可以访问在 cookie 中调用的身份认证提供者。

然而，不能跨越不同的 DNS 域读取 cookie。因此“通用域 Cookie 服务”被用于将服务提供者重定向到正确的身份认证提供者，从而启用用户的单点登录。

配置联合管理资源

可基于要配置内容所在的位置，在网关配置文件中配置联合资源、服务提供者、身份认证提供者和通用域 Cookie 服务(CDCS)。本节说明如何配置以下三种方案：

▼ 配置联合管理资源

1. **全部资源都在公司内联网内**
2. **全部资源都不在公司内联网内或者身份认证提供者驻留在 Internet 上。**
3. **全部资源都不在公司内联网内，或者身份认证提供者被网关保护而服务提供者又是驻留在 Internet 上的第三方。**

配置 1

在此配置中，服务提供者、身份认证提供者和“通用域 Cookie 服务”被部署在同一个公司内联网中并且不在 Internet“域名服务器” (DNS) 中发布身份认证提供者。CDCS 是可选的。

在此配置中，网关指向服务提供者 Portal Server。此配置对于多个 Portal Server 实例有效。

▼ 配置服务提供者 (Portal Server) 的网关

1. **以管理员身份登录到 Portal Server 管理控制台。**
2. **选择“ Secure Remote Access ”选项卡，然后选择相应的网关配置文件以修改其属性。**
   将显示“编辑网关配置文件”页面。
3. **选择“ 核心 ”选项卡。**

4 选中“启用 Cookie 管理”复选框以启用 cookie 管理。

5 选择“安全”选项卡。

6 在 Portal Server 字段中，输入 Portal Server 名称以使用相对 URL，例如：“免验证 URL”列表中列出的 /amserver 或 /portal/dt。例如：

http:// *idp-host:port*/amserver/js

http:// *idp-host:port*/amserver/UI/Login

http://*idp-host:port* /amserver/css

http://*idp-host:port* /amserver/SingleSignOnService

http://*idp-host:port*/amserver/UI/blank

http://*idp-host:port* /amserver/postLogin

http:// *idp-host:port*/amserver/login_images

7 在 Portal Server 字段中，输入 Portal Server 名称。例如，/amserver。

8 单击“保存”。

9 选择“安全”选项卡。

10 在“免验证 URL”列表中，添加联合资源。例如：

/amserver/config/federation

/amserver/IntersiteTransferService

/amserver/AssertionConsumerservice

/amserver/fed_images

/amserver/preLogin

/portal/dt

11 单击“添加”。

12 单击“保存”。

13 如果到达“免验证 URL”列表中列出的 URL 需要使用 Web 代理，请选择“部署”选项卡。

14 在“域和子域的代理”字段中，输入所需的 Web 代理。

15 单击“添加”。

16 单击“保存”。

17 从终端窗口重新启动网关：

```
./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>– t <gateway>
```

配置 2

在此配置中，服务提供者、身份认证提供者和“通用域 Cookie 提供者”(CDCP) 未部署在公司内联网中，或者身份认证提供者是驻留在 Internet 上的第三方提供者。

在此配置中，网关指向服务提供者 Portal Server。此配置对于多个 Portal Server 实例有效。

▼ 配置服务提供者 (Portal Server) 的网关

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“Secure Remote Access”选项卡，然后选择相应的网关配置文件以修改其属性。

3 选择“核心”选项卡。

4 选中“启用 Cookie 管理”复选框以启用 cookie 管理。

5 在 Portal Server 字段中，输入服务提供者的门户服务器名称以使用相对 URL，例如：“免验证 URL”列表中列出的 `/amserver` 或 `/portal/dt`。

```
http://idp-host:port/amserver/js

http://idp-host:port /amserver/UI/Login

http://idp-host:port /amserver/css

http:// idp-host:port/amserver/SingleSignOnService

http://idp-host:port /amserver/UI/blank

http://idp-host:port /amserver/postLogin

http:// idp-host:port/amserver/login_images
```

6 单击“保存”。

7 单击“安全”选项卡。

8 在“免验证 URL”列表中，添加联合资源。例如：

```
/amserver/config/federation
```

```
/amserver/IntersiteTransferService

/amserver/AssertionConsumerservice

/amserver/fed_images

/amserver/preLogin

/portal/dt
```

9 单击“添加”。

10 单击“保存”。

11 如果到达“免验证 URL”列表中列出的 URL 需要使用 Web 代理，请选择“部署”选项卡。

12 在“域和子域的代理”字段中，输入有关 Web 代理的信息。

13 单击“添加”。

14 单击“保存”。

15 从终端窗口重新启动网关：

`./psadmin start-sra-instance –u amadmin – f` <*password file*> `–N` <*profile name*>`– t` <*gateway*>

配置 3

在此配置中，服务提供者、身份认证提供者和“通用域 Cookie 提供者”(CDCP) 未部署在公司内联网中，或者服务提供者是驻留在 Internet 上的第三方提供者并且身份认证提供者受网关保护。

在此配置中，网关指向身份认证提供者 Portal Server。

此配置对于多个 Portal Server 实例有效。此配置在 Internet 上是不可能的，然而，一些公司网络可以在其内联网内采取这种配置，也即，身份认证提供者位于受防火墙保护的子网中，并且可从公司网络内部直接访问服务提供者。

▼ 配置身份认证提供者 (Portal Server) 的网关

1 以管理员身份登录到 Portal Server 管理控制台。

2 选择“Secure Remote Access”选项卡，然后选择相应的网关配置文件以修改其属性。

3 选择“核心”选项卡。

4 选中“ 启用 Cookie 管理 ”复选框以启用 cookie 管理。

5 在 Portal Server 字段中，输入身份认证提供者的门户服务器名称以使用相对 URL，例如：“ 免验证 URL ”列表中列出的 /amserver 或 /portal/dt。

http://*idp-host:port*/amserver/js

http://*idp-host:port* /amserver/UI/Login

http://*idp-host:port* /amserver/css

http:// *idp-host:port*/amserver/SingleSignOnService

http://*idp-host:port* /amserver/UI/blank

http://*idp-host:port* /amserver/postLogin

http:// *idp-host:port*/amserver/login_images

6 单击“ 保存 ”。

7 选择“ 安全 ”选项卡。

8 在“ 免验证 URL ”列表中，添加联合资源。例如：

/amserver/config/federation

/amserver/IntersiteTransferService

/amserver/AssertionConsumerservice

/amserver/fed_images

/amserver/preLogin

/portal/dt

9 单击“ 添加 ”。

10 单击“ 保存 ”。

11 如果到达“ 免验证 URL ”列表中列出的 URL 需要使用 Web 代理，请选择“ 部署 ”选项卡。

12 在“ 域和子域的代理 ”字段中，输入有关 Web 代理的信息。

13 单击“ 添加 ”。

14 单击“ 保存 ”。

15 从终端窗口重新启动网关：

`./psadmin start-sra-instance –u amadmin – f <password file> –N <profile name>– t <gateway>`

◆ ◆ ◆ 附 录 A

配置属性

本附录说明可通过每个 Portal Server Secure Remote Access 组件的 Portal Server 管理控制台为 Sun Java System Portal Server Secure Remote Access 配置的属性：

访问控制服务

第 247 页中的 “访问控制服务”列出了 “访问控制” 服务属性。

表 A–1 访问控制服务属性

属性	默认值	描述
拒绝的 URL		最终用户不能通过网关访问的 URL 列表。
允许的 URL	*	最终用户可通过网关访问的 URL 列表。
禁用单点登录主机		禁用一列主机的单点登录。
启用每个会话的单点登录		启用会话的单点登录。
允许的验证级别	*	表示对验证的信任程度使用星号以允许所有验证级别。有关验证级别的信息，参见 Access Manager 管理指南。

网关服务

单击网关服务时，右侧窗格将显示一个用于创建新配置文件的按钮和一个所有已创建的网关配置文件的列表。

如果单击“新建”，下一个窗格会提示您输入新网关配置文件名。可选择使用默认模板或者使用先前创建的网关配置文件作为模板。

如果单击所列出的网关配置文件名之一，将提供一个标签列表。分别是：

- 第 248 页中的 “核心”
- 第 250 页中的 “代理”
- 第 250 页中的 “安全”
- 第 252 页中的 “重写器”

核心

第 248 页中的 “核心”列出网关服务核心属性。

表 A–2 网关服务核心属性

属性	默认值	描述
启用 HTTPS 连接		启用 HTTPS 连接。
HTTPS 端口	443	指定 HTTPS 端口。
启用 HTTP 连接	*	启用 HTTP 连接。
HTTP 端口	80	指定 HTTP 端口。
启用重写器代理	*	实现网关与内部网之间的安全 HTTP 通信。重写器代理和网关使用相同的网关配置文件。
重写器代理列表		重写器代理列表。若重写器代理有多个实例，则以格式 *host-name:port* 输入每个实例的详细信息。
启用 Netlet	选中	启用 TCP/IP（例如 Telnet 和 SMTP）、HTTP 应用程序和固定端口应用程序的安全。
启用 Proxylet	选中	启用客户机上的 Proxylet 下载。
启用 Netlet 代理		通过将安全隧道从客户机经网关扩展到驻留在内联网中的 Netlet 代理，增强网关和内联网之间 Netlet 通信的安全性。如果不想通过 Portal Server 使用应用程序，则可禁用此选项。

表 A–2 网关服务核心属性 （续）

属性	默认值	描述
Netlet 代理主机		以如下格式列出“Netlet 代理主机”：hostname:port
启用 Cookie 管理		跟踪和管理允许用户访问的所有网站的用户会话。（对 Portal Server 用来跟踪 Portal Server 用户会话的 cookie 不适用）。
启用持久的 HTTP 连接	选中	可在网关处启用 HTTP 持久性连接，以防止为 Web 页面中的所有对象（如图像和样式表）打开套接字。
每个持久性连接的最大请求数量	10	指定每个持久性连接的请求数量。
持久套接字连接超时	50	指定套接字关闭之前需要的时间长短。
Grace 超时，以解决周转时间	20	指定浏览器在发送请求之后该请求到达网关需要的宽限时间，以及网关发送响应和浏览器实际接收到响应之间的时间。
用户会话 Cookie 转发到的 URL		可使 servlet 和 CGI 接收 Portal Server 的 cookie 并使用 API 来标识用户。
最大连接队列长度	50	指定网关可接受的最大并发连接数量。
网关超时时间（秒）	120	指定网关与浏览器连接超时前的时间间隔（以秒为单位）。
线程组合容量最大值	200	指定可在网关线程池中预先创建的最大线程数。
高速缓存套接字超时	200	指定网关与 Portal Server 连接超时前的时间间隔（以秒为单位）。
Portal Server		以 `http://` *portal server name*:*port* *-number* 的格式指定 Portal Server。网关试图以循环方式联络每个列出的 Portal Server 来为请求提供服务。
服务器重试时间间隔（秒）	120	指定在 Portal Server、重写器代理或 Netlet 代理不可用（如，崩溃或死机）之后，尝试启动它们的请求之间的时间间隔。
存储外部服务器 Cookie		允许网关存储和管理通过网关访问的任何第三方应用程序或服务器的 cookie。
从 URL 获取会话信息		无论支持 cookie 与否，均将会话信息作为 URL 的一部分进行编码。网关使用在 URL 中找到的会话信息进行验证，而不使用客户机浏览器发出的会话 cookie。

代理

第 250 页中的 “代理”列出网关服务代理属性。

表 A–3 网关服务代理属性

属性	默认值	描述
使用代理		启用网络代理的应用。
使用 Webproxy URL		列出网关只能通过 “域和子域的代理” 列表中列出的 Web 代理进行联系的 URL（即使禁用 “使用代理服务器” 选项）。
不可使用 Webproxy URL		列出网关可以直接连接到的 URL。
域和子域代理	iportal.com sun.com	指定用于联系特定域中的特定子域的代理。
代理密码列表		如果指定代理服务器需要验证才能访问某些或所有站点，则指定网关向该代理服务器进行验证所需的服务器名、用户名和密码。
启用自动代理配置支持		指定 “域和子域的代理” 字段中的信息将被忽略。
自动代理配置文件位置		指定要用于 PAC 支持的文件的位置。
启用通过网络代理开通 Netlet 隧道		将安全隧道从客户机经由网关扩展到驻留在内部网中的网络代理。

安全

第 250 页中的 “安全”列出网关服务安全属性。

表 A–4 网关服务安全属性

属性	默认值	描述
启用 HTTP 基本验证	选中	保存用户名和密码，这样用户在重新访问受 BASIC 保护的网站时，就无需重新输入其身份验证信息。

表 A–4 网关服务安全属性 （续）

属性	默认值	描述
免验证 URL	/portal/desktop/images /amserver/login_images /portal/desktop/css /amserver/jss /amconsole/console/css /portal/searchadmin/console/js /amconsole/console/js /amserver/css	指定不需要任何验证的 URL，如包含图像的目录。
已启用证书的网关主机		列出启用证书的网关主机。
允许 40 位加密		允许 40 位（弱）“加密套接字层”(SSL) 连接。如果没有选择此选项，则只支持 128 位连接。
启用 SSL 2.0 版本	选中	启用 SSL 2.0 版本。 禁用 SSL 2.0 意味着仅支持较早 SSL 2.0 的浏览器不能向 SRA 进行验证。这将确保更高级别的安全性。
启用 SSL 密码选择		启用 SSL 密码选择。您可选择支持所有预封装的密码，或者可以单独选择所需的密码。您可以为每个网关实例指定特定的 SSL 密码。
SSL2 密码		列出 SSL 2 版本的密码以供选择。
SSL3 密码		列出可以选择的 SSL 3.0 版密码。
TLS 密码		列出 TLS 密码。
启用 SSL 3.0 版本	选中	启用 SSL 3.0 版本。 禁用 SSL 3.0 意味着仅支持 SSL 3.0 的浏览器不能向 SRA 进行验证。这将确保更高级别的安全性。
启用空密码		启用空密码。
信任的 SSL 域		列出信任的 SSL 域。
把 Cookie 标记为安全		将 cookie 标记为安全。必须启用“启用 Cookie 管理”选项。

重写器

重写器标签有两个子部分：

- 第 252 页中的 “基本”
- 第 253 页中的 “高级”

基本

第 252 页中的 “基本”列出网关服务重写器基本属性。

表 A–5 网关服务重写器属性－基本

属性	默认值	描述
启用全部 URI 重写		指定重写所有 URI，但不检查 “域和子域代理” 列表中的条目。
将 URI 映射至规则集	*://*.iportal.com*/portal/* |default_gateway_ruleset */portal/NetFileOpenFileServlet* |null_ruleset *|generic_ruleset REPLACE_WITH_IPLANET_MAIL_SERVER_NAME| iplanet_mail_ruleset REPLACE_WITH_EXCHANGE_SERVER_ NAMEexchange_2000sp3_owa_ruleset *://*.iportal.com*/amconsole/* |default_gateway_ruleset REPLACE_WITH_INOTES_SERVER_NAME|inotes_ruleset http*://*/portal/NetFileController* |null_ruleset	使用 “将 URI 映射至规则集” 列表将域与规则集相关联。规则集是在 Access Manager 管理控制台中的 Portal Server 配置下创建的。
将解析器映射到 MINE 类型	JAVASCRIPT=application/x-java XML=text/xml HTML=text/html;text/htm;text/x-component;text /wml;text/vnd.wap.wml CSS=text/css	将新的 MIME 类型与 HTML、JAVASCRIPT、CSS 或 XML 相关联。使用分号或逗号分隔多个条目。

表 A–5 网关服务重写器属性－基本 （续）

属性	默认值	描述
禁止重写的 URI		列出禁止重写的 URI。注释：向该列表中添加 #* 可重写 URI（即使规则集中包含 href 规则）。
默认域		将主机名解析为默认域和子域。这是在安装期间指定的

高级

第 253 页中的 “高级” 列出网关服务重写器高级属性。

表 A–6 网关服务重写器属性－高级

属性	默认值	描述
启用 MIME 推测		未发送 MIME 时，启用 MIME 推测。必须将数据添加到 “将解析器映射至 URI” 列表框中。
将解析器映射至 URI 映射		将解析器映射到 URI。多个 URI 以分号进行分隔。 例如，HTML=*.html; *.htm;*Servlet 它表示会使用重写器来重写具有 html、htm 或 Servlet 扩展名的任何页的内容。
启用屏蔽		允许重写器重写 URI 以便使人们看不到页的 “内部网 URL”。
用于屏蔽的种子字符串		指定用于屏蔽 URI 的种子字符串。此随机字符串由屏蔽算法生成。
禁止屏蔽的 URI		指定不进行屏蔽的 Internet URI。应用程序（如 applet）需要 Internet URI 时，使用此项。 例如，如果添加了 */Applet/Param* 到列表框中，则当内容 URI http://abc.com/Applet/Param1.html 在规则集的规则中匹配时，将不会屏蔽此 URI。
使网关协议与原始 URI 协议相同		启用重写器以使用一致的协议访问 HTML 内容中的引用资源。 这样做只适用于静态 URI，不适用于 Javascript 中生成的动态 URI。

NetFile 服务

单击“NetFile 服务”时，右侧窗格显示选项卡。分别是：

主机

“主机”标签有两个子部分：

配置

第 254 页中的 “配置”列出 NetFile 主机配置属性。

表 A–7 NetFile 服务主机配置属性

属性	默认值	描述
OS 字符集	Unicode(UTF-8)	指定与主机进行通信时用作默认编码的字符集。
主机侦测顺序	WIN、NETWARE、FTP、NFS	指定主机侦测顺序。
通用主机		指定所有远程 NetFile 用户均可通过 NetFile 使用的主机。
默认域		指定 NetFile 联络允许主机时需要使用的默认域。
默认 Microsoft Windows 域/工作组		指定用户要访问的 Windows 主机的默认 Microsoft Windows 域或工作组。
默认 WINS/DNS 服务器		指定 NetFile 用于访问 windows 主机的 WINS/DNS 服务器。

访问

第 254 页中的 “访问”列出 NetFile 服务主机访问属性。

表 A–8 NetFile 服务主机访问属性

属性	默认值	描述
允许访问 Windows 主机	选中	允许访问 Microsoft Windows 主机。
允许访问 FTP 主机	选中	允许访问 FTP 主机。
允许访问 NFS 主机	选中	允许访问 NFS 主机。
允许访问 Netware 主机	选中	允许访问 Netware 主机。
允许的主机	*	指定用户可通过 NetFile 访问的主机。
拒绝的主机		指定用户不能通过 NetFile 访问的主机。

权限

如果您在用户开始使用 NetFile 后禁用了这些选项，则仅当用户从 NetFile 中注销并重新登录时，此更改才会生效。

第 255 页中的 “权限” 列出 NetFile 服务权限属性。

表 A–9 NetFile 服务权限属性

属性	默认值	描述
允许文件重命名	选中	允许用户重命名文件。
允许删除文件/文件夹	选中	允许用户删除文件和文件夹。
允许文件上载	选中	允许用户上载文件。
允许文件/文件夹下载	选中	允许用户下载文件和文件夹。
允许文件搜索	选中	允许用户进行搜索。
允许文件邮件	选中	允许邮寄文件。
允许文件压缩	选中	允许文件压缩。
允许改变用户 Id	选中	允许用户使用不同的 ID。
允许改变 Windows 域	选中	允许用户更改 Microsoft Windows 域。

视图

第 255 页中的 “视图” 列出 NetFile 服务视图属性。

表 A–10 NetFile 服务视图属性

属性	默认值	描述
窗口大小	700|400	在用户桌面上以像素为单位指定 NetFile 窗口的大小。如果输入了无效值，NetFile 会使用默认值。
窗口位置	100|50	指定 NetFile 窗口在用户桌面上的显示位置。如果输入了无效值，NetFile 会使用默认值。

操作

“操作”标签有下列子部分：

- 第 256 页中的 “通信量”
- 第 256 页中的 “搜索”
- 第 257 页中的 “压缩”

通信量

第 256 页中的 “通信量”列出 NetFile 服务操作通信属性。

表 A–11 NetFile 服务操作 - 通信属性

属性	默认值	描述
临时目录位置	/tmp	指定各种 NetFile 文件操作的临时目录。 确保运行 Web 服务器的 ID（如 nobody 或 noaccess）对所指定的目录具有 rwx 权限。还要确保此 ID 对于到所需临时目录的完整路径具有 rx 权限。 最好为 NetFile 创建一个单独的临时目录。如果所指定的临时目录对于 Portal Server 的所有模块来说是公用目录，磁盘空间可能很快就会用完。如果临时目录没有空间，NetFile 便无法工作。
文件上载限制 (MB)	5	指定可以上载的最大文件大小。如果输入无效值，NetFile 会将其重置为默认值。请确保输入整数值。 可为不同用户指定不同的文件上载大小限制。

搜索

第 256 页中的 “搜索”列出 NetFile 服务操作搜索属性。

表 A–12 NetFile 服务操作 - 搜索属性

属性	默认值	描述
搜索目录限制	100	指定单个搜索操作中可搜索的最大目录数。

压缩

第 257 页中的 “压缩” 列出 NetFile 服务操作压缩属性。

表 A–13 NetFile 服务操作 - 压缩属性

属性	默认值	描述
默认压缩类型	Zip	指定 Zip 或 Gzip 压缩类型。
默认压缩级别	6	指定压缩级别，1 和 9 之间的一个数字。

常规

第 257 页中的 “常规” 列出 Netfile 服务常规属性。

表 A–14 NetFile 服务－常规属性

属性	默认值	描述
MIME 类型配置文件位置	/opt/S1PS62/SUNWportal/samples/config/netfile	指定要发送到客户机浏览器的响应内容类型。

Netlet 服务

第 257 页中的 “Netlet 服务” 列出 Netlet 服务属性。

表 A–15 Netlet 服务属性

属性	默认值	描述
Netlet 规则		选择添加或删除规则。
如果添加规则，下列九个属性是必需的：		
--规则名称		为规则指定唯一名称。
--加密密码		指定所需密码。

表 A–15 Netlet 服务属性 （续）

属性	默认值	描述
--URL		指定要调用的应用程序的 URL。
--下载 Applet		需要下载 applet 时指定。如果使用 applet，相关编辑框中的语法为： local-port:server-host:server-port
--扩展会话		确保与该规则相对应的 Netlet 会话运行期间，延长 Portal Server 会话时间。
--将本地端口映射至目标服务器端口		指定本地端口、目标主机和目标端口。输入这些值（在此表的后三行中）之后，单击添加，使其出现在列表中。
--本地端口		指定 Netlet 进行侦听时所在的本地端口。对于 FTP 规则，本地端口值必须是 30021。
--目标主机		静态规则包含用于 Netlet 连接的目标机器的主机名。 动态规则包含单词 "TARGET"。
--目标端口		指定目标主机上的端口。
默认本地 VM 密码		为 Netlet 规则指定默认密码。在使用不包含密码的现有规则时，此功能非常有用。
默认 Java Plugin 密码		为 Netlet 规则指定默认密码。在使用不包含密码的现有规则时，此功能非常有用。
默认回环端口	58000	当通过 Netlet 下载 applet 时，指定用于客户机的端口。在 Netlet 规则中，可以覆盖默认值。
连接时重新验证		确保用户在每次需要建立 Netlet 连接时，均输入 Netlet 密码。
显示连接时弹出警告	选中	当用户在 Netlet 上运行应用程序或是有入侵者企图通过侦听端口获得桌面的访问权时，显示消息。
在端口警告对话框中显示复选框	选中	当 Netlet 尝试连接到用户标准“Portal 桌面”上的目标主机时，为用户提供禁止“弹出警告对话框”的选项。
保活间隔（分钟）	0	如果客户机是通过 Web 代理连接到网关的，则会因代理超时而断开空闲的 Netlet 连接。为防止出现这种情况，此参数的给定值应小于代理超时值。
门户注销时中止 Netlet	选中	用户从 Portal Server 注销时，确保所有连接均终止。

表 A–15 Netlet 服务属性 （续）

属性	默认值	描述
Netlet 访问规则	*	为某些组织、角色或用户定义对特定 Netlet 规则的访问。
Netlet 拒绝规则		拒绝某些组织、角色或用户对特定 Netlet 规则的访问。
允许的主机	*	为某些组织、角色或用户定义对特定主机的访问。
拒绝的主机		拒绝对组织内特定主机的访问。

Proxylet 服务

第 259 页中的 “Proxylet 服务”列出 Proxylet 服务属性。

表 A–16 Proxylet 服务属性

属性	默认值	描述
自动下载 Proxylet Applet		如果选中复选框，用户登录时 Proxylet 会被下载到客户机上。
默认 Proxylet Applet 绑定 IP	127.0.0.1	Proxylet Applet 驻留的 IP 地址。
默认 Proxylet Applet 端口	58081	这是 Proxylet 进行侦听的端口。

◆ ◆ ◆ 附 录 B

日志文件

以下日志文件位于默认 /var/opt/SUNWportal/debug 目录中，并且包含调试信息和其他类型的信息。

关于日志文件

表 B–1 信息文件和调试文件

文件名	内容
以下日志文件由默认目录 /etc/opt/SUNWam/debug/file 下的 AMConfig-*instance-name*.properties 文件中的调试参数控制。有关 Linux 路径名，参见“Solaris 与 Linux 路径名比较”。	
amconsole	Netfile、Netlet 及网关管理文件
srapNetFile	NetFile 信息文件
srapNetlet	Netlet 信息文件
srapProxylet	Proxylet 信息文件
以下日志文件由默认目录 /etc/opt/SUNWportal 中 platform.conf.gateway-profile-name 文件内的调试参数 gateway.debug 控制。有关 Linux 路径名，参见“Solaris 与 Linux 路径名比较”。	
srapGateway.gateway-profile-name	网关信息

表 B–1 信息文件和调试文件 （续）

文件名	内容
Gateway_to_from_server.gateway-profile-name	
Gateway_to_from_browser.gateway-profile-name	
srapNetletProxy.gateway-profile-name	
srapRewriterProxy.gateway-profile-name	
rwproxy.log.rewriter-proxy-instance-name	重写器代理的开始时间和停止时间
nlproxy.log.netlet-proxy-instance-name	Netlet 代理的开始时间和停止时间
gateway.log.gateway.instance.name	网关的开始时间和停止时间
以下重写器文件由默认目录 `/var/opt/SUNWam/config/file` 下的 `AMConfig-`*instance-name* `.properties` 文件中的调试参数控制。有关详细信息，参见第 88 页中的 “使用调试日志排除故障” 。	
RuleSetInfo	本文件记录重写时已使用的所有规则集。
Original Pages	包含页面 URI、解析 URI（如果解析 URI 不同于页面 URI）、内容 MIME、已应用于页面的规则集、解析器 MIME，以及原始内容。 与解析有关的特定错误/警告/消息也出现在本文件中。 在消息模式下，会记录全部内容；在警告和错误模式下，只记录重写期间出现的异常。
Rewritten Pages	包含页面 URI、解析 URI（如果解析 URI 不同于页面 URI）、内容 MIME、已应用于页面的规则集、解析器 MIME，以及重写后的内容。 当将调试模式设置为消息时，将会填写本文件。
Unaffected Pages	包含未经修改的页列表。
URIInfo Pages	本文件包含已找到并经过转换的 URL。}该文件会记录内容仍与原始数据相同的所有页的详细信息。 所记录的详细信息有：页 URI、MIME 及编码数据、重写时所用的规则集 ID，以及解析器 MIME。

◆ ◆ ◆ 附 录 C

国家代码

下表列出了在证书管理期间需要指定的两字母国家/地区代码。

国家/地区代码列表

表 C–1 两字母国家代码

ad	安道尔公国
ae	阿拉伯联合酋长国
af	阿富汗伊期兰共和国
ag	安提瓜和巴布达
ai	安圭拉
al	阿尔巴尼亚
am	亚美尼亚
an	荷属安的列斯群岛
ao	安哥拉
aq	南极洲
ar	阿根廷
arpa	老式阿帕网
as	美属萨摩亚
at	奥地利
au	澳大利亚

表 C–1 两字母国家代码 （续）

aw	阿鲁巴
az	阿塞拜疆
ba	波斯尼亚-黑塞哥维纳
bb	巴巴多斯
bd	孟加拉国
be	比利时
bf	布基纳法索
bg	保加利亚
bh	巴林
bi	布隆迪
bj	贝宁
bm	百慕大
bn	文莱达鲁萨兰国
bo	玻利维亚
br	巴西
bs	巴哈马
bt	不丹
bv	博维特岛
bw	博茨瓦纳
by	白俄罗斯
bz	伯利兹
ca	加拿大
cc	科科斯（基灵）群岛
cf	中非共和国
cd	刚果民主共和国
cg	刚果
ch	瑞士
ci	象牙海岸
ck	库克群岛

表 C–1 两字母国家代码 （续）

cl	智利
cm	喀麦隆
cn	中国
co	哥伦比亚
com	商业
cr	哥斯达黎加
cs	前捷克斯洛伐克
cu	古巴
cv	佛得角
cx	圣诞岛
cy	塞浦路斯
cz	捷克共和国
de	德国
dj	吉布提
dk	丹麦
dm	多米尼加
do	多米尼加共和国
dz	阿尔及利亚
ec	厄瓜多尔
edu	教育
ee	爱沙尼亚
eg	埃及
eh	西撒哈拉
er	厄立特里亚
es	西班牙
et	埃塞俄比亚
fi	芬兰
fj	斐济
fk	福克兰群岛

表 C–1 两字母国家代码 （续）

fm	密克罗尼西亚
fo	法罗群岛
fr	法国
fx	法国（欧洲领土）
ga	加蓬
gb	大不列颠
gd	格林纳达
ge	格鲁吉亚
gf	法属圭亚那
gh	加纳
gi	直布罗陀
gl	格陵兰
gm	冈比亚
gn	几内亚
gov	美国政府
gp	瓜德罗普（法属）
gq	赤道几内亚
gr	希腊
gs	南 乔治亚和南桑德韦奇群岛
gt	危地马拉
gu	关岛（美属）
gw	几内亚比绍
gy	圭亚那
hk	中国香港特别行政区
hm	赫德岛和麦克唐纳群岛
hn	洪都拉斯
hr	克罗地亚
ht	海地
hu	匈牙利

表 C–1 两字母国家代码 （续）

id	印度尼西亚
ie	爱尔兰
il	以色列
in	印度
int	国际
io	英属印度洋地区
iq	伊拉克
ir	伊朗
is	冰岛
it	意大利
jm	牙买加
jo	约旦
jp	日本
ke	肯尼亚
kg	吉尔吉斯共和国（吉尔吉斯斯坦）
kh	柬埔寨王国
ki	基里巴斯
km	科摩罗
kn	圣基茨和尼维斯安圭拉
kp	朝鲜
kr	韩国
kw	科威特
ky	开曼群岛
kz	哈萨克斯坦
la	老挝
lb	黎巴嫩
lc	圣卢西亚
li	列支敦士登
lk	斯里兰卡

表 C–1 两字母国家代码 （续）

lr	利比里亚
ls	莱索托
lt	立陶宛
lu	卢森堡
lv	拉脱维亚
ly	利比亚
ma	摩洛哥
mc	摩纳哥
md	摩尔达维亚
mg	马达加斯加
mh	马绍尔群岛
mil	美国军方
mk	马其顿
ml	马里
mm	缅甸
mn	蒙古
mo	中国澳门特别行政区
mp	北马里亚纳群岛
mq	马提尼克（法属）
mr	毛里塔尼亚
ms	蒙特塞拉特
mt	马耳他
mu	毛里求斯
mv	马尔代夫
mw	马拉维
mx	墨西哥
my	马来西亚
mz	莫桑比克
na	纳米比亚

表 C–1 两字母国家代码 （续）

nato	北约（1996 年清除了此项 - 参见 hq.nato.int）
nc	新喀里多尼亚（法属）
ne	尼日尔
net	网络
nf	诺福克岛
ng	尼日利亚
ni	尼加拉瓜
nl	荷兰
no	挪威
np	尼泊尔
nr	瑙鲁
nt	中立区
nu	纽埃
nz	新西兰
om	阿曼
org	非盈利组织（原文）
pa	巴拿马
pe	秘鲁
pf	玻利尼西亚（法属）
pg	巴布亚新几内亚
ph	菲律宾
pk	巴基斯坦
pl	波兰
pm	圣皮埃尔和密克隆
pn	皮特克恩岛
pr	波多黎各
pt	葡萄牙
pw	帕劳
py	巴拉圭

表 C–1 两字母国家代码 （续）

qa	卡塔尔
re	留尼旺（法属）
ro	罗马尼亚
ru	俄罗斯联邦
rw	卢旺达
sa	沙特阿拉伯
sb	所罗门群岛
sc	塞舌尔
sd	苏丹
se	瑞典
sg	新加坡
sh	圣赫勒拿
si	斯洛文尼亚
sj	斯瓦尔巴特和扬马延群岛
sk	斯洛伐克共和国
sl	塞拉利昂
sm	圣马力诺
sn	塞内加尔
so	索马里
sr	苏里南
st	圣多美和普林西比
su	前苏联
sv	萨尔瓦多
sy	叙利亚
sz	斯威士兰
tc	特克斯和凯科斯群岛
td	乍得
tf	法属南部领土
tg	多哥

表 C–1 两字母国家代码 （续）

th	泰国
tj	塔吉克斯坦
tk	托克劳
tm	土库曼斯坦
tn	突尼斯
to	汤加
tp	东帝汶
tr	土耳其
tt	特立尼达和多巴哥
tv	图瓦卢
tw	中国台湾
tz	坦桑尼亚
ua	乌克兰
ug	乌干达
uk	英国
um	美国边远小岛
us	美国
uy	乌拉圭
uz	乌兹别克斯坦
va	圣座（梵蒂冈城国）
vc	圣文森特和格林纳丁斯
ve	委内瑞拉
vg	维尔京群岛（英属）
vi	维尔京群岛（美属）
vn	越南
vu	瓦努阿图
wf	瓦利斯和富图纳群岛
ws	萨摩亚
ye	也门

表 C–1 两字母国家代码 （续）

yt	马约特
yu	南斯拉夫
za	南非
zm	赞比亚
zr	扎伊尔
zw	津巴布韦

索引

O

P

R

S

T

U

主

自

组