Sun Java System Portal Server Secure Remote Access 7.2 Administration Guide

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

Part No: 820–2049–10
May 2008

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun^{TM} Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

080522@20209

Contents

Figures

Tables

Examples

Preface

This guide explains how to administer the Sun Java™ System Portal Server Secure Remote Access 7.2 server.

The Sun Java System Portal Server Secure Remote Access (SRA) server enables remote users to securely access their organization's network and its services over the Internet. Additionally, the SRA provides your organization with a secure internal portal, providing access to content, applications, and data to any targeted audiences such as employees, business partners, or the general public.

This preface has the following sections:

- "Who Should Use This Book" on page 17
- "Before You Read This Book" on page 18
- "How This Book Is Organized" on page 18
- "Related Books" on page 19
- "Other Server Documentation" on page 20
- "Related Third-Party Web Site References" on page 20
- "Shell Prompts in Command Examples" on page 21

Who Should Use This Book

The Sun Java System Portal Server Secure Remote Access 7.2 Administration Guide is intended for users that configure and administer the Secure Remote Access server.

The Sun Java System Portal Server Secure Remote Access 7.2 Administration Guide assumes that you are a network or system administrator experienced in managing UNIX systems and TCP/IP networks. You do not need root access to the required machines for installing the various components of the Secure Remote Access server. You do need the required administrative privileges to carry out other operations such as configuring users and services.

Before You Read This Book

Portal Secure Remote Access server administrators should understand the following technology:

- Sun Java System Portal Server
- Sun Java System Directory Server
- Sun Java System Access Manager
- Your web container, such as:
  - Sun Java System Application Server 8.2
  - Sun Java System Web Server 7.0
- Your operating system
- Basic UNIX® administrative procedures
- Lightweight Directory Access Protocol (LDAP)
- Web Services for Remote Portlets (WSRP)

You also need to know the following to be able to write Rewriter rules:

- Understanding of Hypertext Markup Language (HTML) and HTML tags
- A fair knowledge of JavaScript™
- Basic knowledge of Extensible Markup Language (XML)

How This Book Is Organized

This book is organized as follows:

- Part I
  - Chapter 1, "Introduction to Portal Server Secure Remote Access Server," describes the relationship between Sun Java System Portal Server and Portal Server Secure Remote Access.
  - Chapter 2, "Working With Gateway," explains Gateway related concepts and tasks to manage the Gateway.
  - Chapter 3, "Working With Proxylet," describes Proxylet, which enables users to access intranet web pages through the Gateway without parsing the web pages.
  - Chapter 4, "Working with Rewriter," describes how to access the intranet web pages through the Gateway using Proxylet and Rewriter.
  - Chapter 5, "Working with NetFile," describes how to access and operate remote file systems and directories using NetFile.
  - Chapter 6, "Working with Netlet," explains how to securely run common TCP/IP services over insecure networks such as the Internet using Netlet.

- Part II
  - Chapter 7, "Configuring the Secure Remote Access Server Access Control," describes how to manage access to the Portal Server administration console.
  - Chapter 8, "Configuring the Secure Remote Access Gateway," explains how to configure the Gateway attributes from the Portal Server management console.
  - Chapter 9, "Configuring Rewriter in the Gateway Service," explains how you can use Gateway services under the Rewriter tab to perform various tasks.
  - Chapter 10, "Working with Certificates," describes managing certificates and installing self-signed certificates from a Certificate Authority.
  - Chapter 11, "Configuring the Netlet," describes configuring the Netlet attributes from the Portal Server management console.
  - Chapter 12, "Configuring Netlet With Private Domain Certificates," describes configuring the client browser's Java Plug–in, so that Netlet can be used with PDC.
  - Chapter 13, "Configuring Proxylet" describes configuring Proxylet from the Portal Server management console.
  - Chapter 14, "Configuring NetFile," describes using the Portal Server management console to set up NetFile options, privileges, and preferences.
  - Chapter 15, "Configuring Secure Socket Layer Accelerators," describes configuring various accelerators for Portal Server Secure Remote Access Server.
- Part III
  - Chapter 16, "Managing the Gateway," explains the way to create a Gateway Profile and Gateway instances.
  - Chapter 17, "Federation Management Scenarios," explains the various scenarios in maintaining a network identity.
- Appendix A, "Configuration Attributes," describes attributes that you can configure for Sun Java System Portal Server Secure Remote Access through the Portal Server administration console for each Portal Server Secure Remote Access component.
- Appendix B, "Log Files," contain debug and other types of information.
- Appendix C, "Country Codes," lists the two-letter country codes that you need to specify during certificate administration.

Related Books

- *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 Administration Guide*
- *Sun Java System Portal Server 7.2 Command-Line Reference*
- *Sun Java System Portal Server 7.2 Release Notes*

- *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*

An introduction to Portal Server concepts and components is available in the *Sun Java System Portal Server 7.2 Technical Overview*.

Other Server Documentation

For other server documentation, go to the following:

- Directory Server documentation at `http://docs.sun.com/coll/1224.1`
- Access Manager documentation at `http://docs.sun.com/coll/1292.2`
- Web Server documentation at `http://docs.sun.com/coll/1308.3`
- Application Server documentation at `http://docs.sun.com/coll/1310.3`
- Web Proxy Server documentation at `http://docs.sun.com/coll/1311.4`

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell	`machine_name%`
C shell for superuser	`machine_name#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell for superuser	`#`

PART I

Secure Remote Access Server Components

◆ ◆ ◆ CHAPTER 1

Introduction to Portal Server Secure Remote Access Server

This chapter describes the Sun Java™ System Portal Server Secure Remote Access and the relationship between the Sun Java System Portal Server and Sun Java System Portal Server Secure Remote Access components.

This chapter covers the following topics:

- "Introduction to Secure Remote Access" on page 25
- "Secure Remote Access Services" on page 28
- "Supported Applications" on page 30

Introduction to Secure Remote Access

Secure Remote Access enables remote users to securely access their organization's network and its services over the Internet. Additionally, it gives your organization a secure internet portal, providing access to content, applications, and data to any targeted audience such as employees, business partners, or the general public.

Secure Remote Access offers browser-based secure remote access to portal content and services from any remote device. Secure Remote Access is a secure access solution that is accessible to users from any device with a Java™ technology-enabled browser, eliminating the need for client software. Integration with Portal Server ensures that users receive secure encrypted access to the content and services that they have permission to access.

Secure Remote Access software is targeted toward enterprises deploying highly secure remote access portals. These portals emphasize security, protection, and privacy of intranet resources. The architecture of Secure Remote Access is well suited to these types of portals. Secure Remote Access software enables users to securely access intranet resources through the Internet without exposing these resources to the Internet.

Portal Server can function in two modes, Open Mode and Secure Mode as described in the following sections.

Open Mode

In open mode, Portal Server is installed without Secure Remote Access. Although HTTPS communication is possible in this mode, secure remote access is not possible. Users therefore cannot access secure remote file systems and applications.

The main difference between an open portal and a secure portal is that the services presented by the open portal typically reside within the demilitarized zone (DMZ) and not within the secured intranet. A DMZ is a small protected network between the public Internet and a private intranet, usually demarcated with a firewall on both ends.

If the portal does not contain sensitive information both of either deploying public information and allowing access to free applications, then responses to access requests by a large number of users is faster than using secure mode.

In Open Mode, Portal Server is installed on a single server behind the firewall. Multiple clients access Portal Server across the Internet through the single firewall.

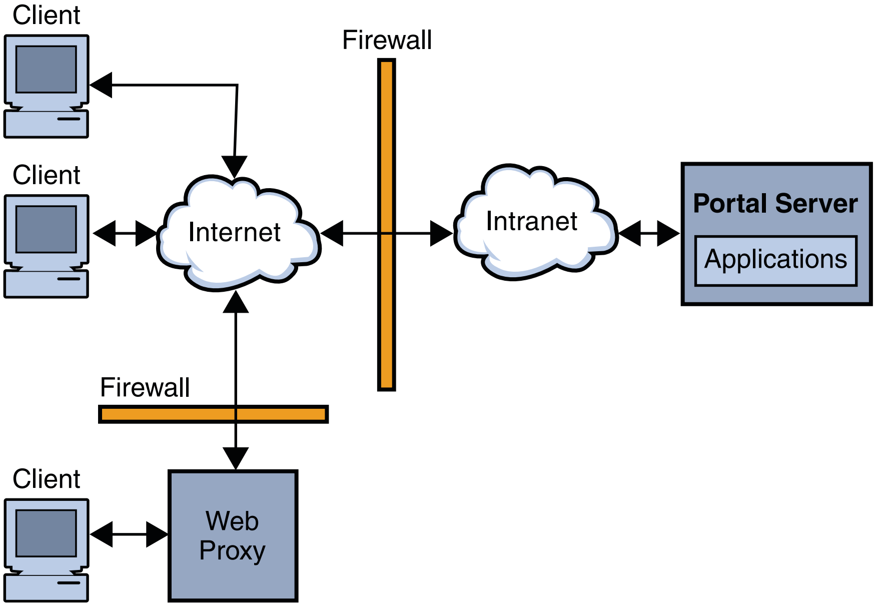


FIGURE 1–1 Portal Server in Open Mode with Secure Remote Access

Secure Mode

Secure mode provides users with secure remote access to required intranet file systems and applications.

The Gateway resides in the demilitarized zone (DMZ). The Gateway provides a single secure access point to all intranet URLs and applications, thus reducing the number of ports to be opened in the firewall. All other Portal Server services such as Session, Authentication, and the standard Portal Desktop reside behind the DMZ in the secured intranet. Communication from the client browser to the Gateway is encrypted using HTTP over Secure Sockets Layer (SSL). Communication from the Gateway to the server and intranet resources can be either HTTP or HTTPS.

In Secure Mode, SSL is used to encrypt the connection between the client and the Gateway over the Internet. SSL can also be used to encrypt the connection between the Gateway and the server. The presence of the Gateway between the intranet and the Internet extends the secure path between the client and the Portal Server.

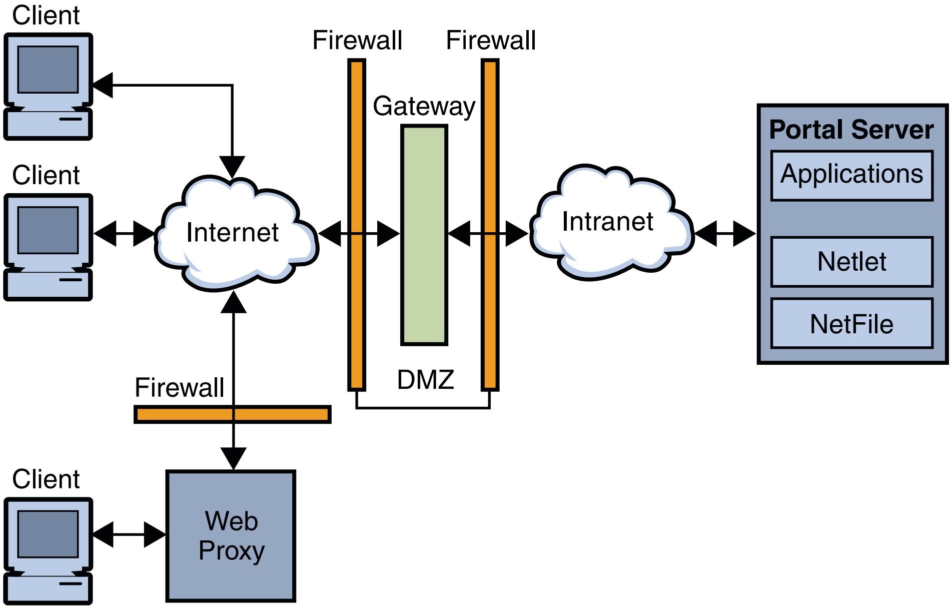


FIGURE 1–2 Portal Server in Secure Mode with Secure Remote Access

Additional servers and gateways can be added for site expansion. Secure Remote Access software can be configured in various ways based on the business requirement. For more information on how to accommodate your business requirements, see *Sun Java System Portal Server 7.2 Deployment Planning Guide*.

Secure Remote Access Services

Secure Remote Access software has five major components:

- **Gateway**

  The SRA Gateway provides the interface and security barrier between remote user sessions originating from the Internet and a corporate intranet. The gateway presents content securely from internal web servers and application servers through a single interface to a remote user.

  Web servers use web-based resources such as HTML, JavaScript and XML to communicate between the client and the Gateway. Rewriter is the Gateway component used to make web content available.

  Application servers use binary protocol such as telnet and FTP to communicate between the client and Gateway. Netlet, which resides on the Gateway, is used for this purpose. See Chapter 2, "Working With Gateway," for more detail.

- **Rewriter**

  Rewriter enables end users to browse the intranet and makes links and other URL references on those pages operate correctly. Rewriter prepends the Gateway URL in the location field of the web browser, thereby redirecting content requests through the Gateway. See Chapter 4, "Working with Rewriter," for details.

- **Netfile**

  NetFile is a file manager application that enables remote access and operation of file systems and directories. NetFile includes a Java based user interface. See Chapter 5, "Working with NetFile," for details.

- **Netlet**

  Netlet facilitates the running of popular or company-specific applications on remote desktops in a secure manner. After you implement Netlet at your site, users can securely run common TCP/IP services, such as Telnet and SMTP, and HTTP-based applications such as pcANYWHERE or Lotus Notes. See Chapter 6, "Working with Netlet," for details.

- **Proxylet**

  Proxylet is a dynamic proxy server that runs on a client machine. Proxylet redirects a URL to the Gateway, by reading and modifying the proxy settings of the browser on the client machine so that they point to the local proxy server or Proxylet.

Configuring the Secure Remote Access Attributes

You configure Secure Remote Access attributes on the Portal Server administration console using the following services:

- Access Control

  This service enables you to allow or restrict access to specific URLs and to manage the single sign-on feature. For more information, see Chapter 7, “Configuring the Secure Remote Access Server Access Control.”
- Gateway

  Profiles (Gateway Instances) This service enables you to configure all Gateway related attributes such as enabling components, cookie management, proxy management, security settings, performance tuning, rewriter mapping management. For more information, see Chapter 8, “Configuring the Secure Remote Access Gateway.”
- NetFile

  This service enables you to configure all NetFile related attributes such as common hosts, MIME types, and access to different types of hosts. For more information, see Chapter 14, “Configuring NetFile.”
- Netlet

  This service enables you to configure all Netlet related attributes such as Netlet rules, access to required rules, organizations and hosts, and the default algorithm. For more information, see Chapter 11, “Configuring the Netlet.”
- Rewriter

  This service enables you to download, upload and delete all rewriter rulesets.
- Proxylet

  This service enables you to configure Proxylet related attributes such as Proxylet Applet Bind IP address and port number. For more information, see Chapter 13, “Configuring Proxylet.”

Caution – The Gateway does not receive notifications for attribute changes that are made while Gateway is running. Restart the Gateway for updated profile attributes (belonging to the Gateway or any other service) to take effect. For more information, see “Configuring Gateway Attributes Using the Command Line Options” on page 180.

Setting Conflict Resolution

▼ To Set the Conflict Resolution Level

1. **"To Login to the Management Console" in** *Sun Java System Portal Server 7.2 Administration Guide*

2. **Select the Secure Remote Access tab and click the required service tab: Netlet, Netfile, or Proxylet.**

3. **Select the Organization or Role from the Select DN drop-down menu.**

4. **Select the required Conflict Resolution Level from the COS Priority drop-down box.**

5. **Click Save to complete.**

Supported Applications

SRA supports the following applications:

- Sun Java System Calendar Server Release 5.1.1 and later
- Sun Java System Messenger Express 6 2005Q1 - Sun Java System Messaging Server 5.2 and later
- Sun Java System Communications Express 6 2005Q1

Before You Begin

▼ To Enable SRA for a Portal

1. **Switch SRA status by using the command** `PortalServer_base/psadmin switch-sra-status -u amadmin -f <passwordfile> on`**.**

2. **Provision the SRA status by using the command** `PortalServer_base/psadmin provision-sra -u amadmin -f <passwordfile> -p <portal-id> --gateway-profile <profile-name> --enable`**.**

◆ ◆ ◆ CHAPTER 2

Working With Gateway

This chapter describes Gateway related concepts. For information on managing the gateway, see Chapter 16, “Managing the Gateway.” For information about configuring the Gateway, see Chapter 8, “Configuring the Secure Remote Access Gateway.”

This chapter covers the following topics:

- “Introduction to Gateway” on page 31
- “Understanding the platform.conf File” on page 34
- “Using Web Proxies” on page 40
- “Using Automatic Proxy Configuration” on page 46
- “Using a Netlet Proxy” on page 48
- “Using a Rewriter Proxy” on page 52
- “Using a Reverse Proxy with the Gateway” on page 54
- “Obtaining Client Information” on page 54
- “Using Authentication Chaining” on page 56
- “Using Wild Card Certificates” on page 56
- “Disabling Browser Caching” on page 56
- “Customizing the Gateway Service User Interface” on page 57

Introduction to Gateway

The Gateway provides the interface and security barrier between remote user sessions originating from the Internet and your corporate intranet. The Gateway presents content securely from internal web servers and application servers through a single interface to a remote user.

For each Gateway instance you must complete the following tasks:

- “Creating a Gateway Profile” on page 32
- “Creating Multiple Instances of a Gateway” on page 32
- Chapter 8, “Configuring the Secure Remote Access Gateway.”

Other gateway related topics include the following:

- "Restarting the Gateway" on page 33
- "Configuring the Gateway Watchdog" on page 33
- "Specifying a Virtual Host" on page 33
- "Specifying a Proxy to Contact Access Manager" on page 34

Creating a Gateway Profile

A gateway profile contains all the information related to gateway configuration, such as the port on which the Gateway listens, SSL options, and proxy options. When you install a Gateway, if you choose the default values, a default gateway profile called "default" is created. A configuration file corresponding to the default profile exists at: `/etc/opt/SUNWportal/platform.conf.default`.

Where `/etc/opt/SUNWportal` is the default location for all the `platform.conf.*` files. For more information on the `platform.conf` file, see "Understanding the platform.conf File" on page 34.

When working with profiles, you can perform the following tasks:

- Create multiple profiles, define attributes for each profile, and assign these profiles to different Gateways as required.
- Assign a single profile to Gateway installations on different machines.
- Assign different profiles to instances of a single Gateway running on the same machine.

Caution – Do not assign the same profile to different instances of the Gateway running on the same machine. This setup causes a conflict because the port numbers are the same.

Do not specify the same port numbers in the different profiles created for the same Gateway. Running multiple instances of the same Gateway with the same port causes a conflict.

Creating Multiple Instances of a Gateway

To create multiple instances of a gateway, see Chapter 4, "Installing and Configuring a Gateway With Portal Server," in *Sun Java System Portal Server 7.2 Installation and Configuration Guide*

Creating Multi-homed Gateway Instances

Multi-homed gateway instances are multiple gateways on one Portal Server. To create these instances, modify the `platform.conf` file as follows:

```
gatewaybindipaddress = 0.0.0.0
```

Creating Gateway Instances Using the Same LDAP

If you are creating multiple gateway instances that use the same LDAP, after creating the first Gateway on all subsequent Gateways:

In `/etc/opt/SUNWam/config/`, modify the following areas in `AMConfig-`*instance-name*`.properties` to be consistent with the first installed instance of the Gateway.

See "To Create Gateway Instances Using the Same LDAP" on page 240

Restarting the Gateway

Normally, you do not need to restart the Gateway. You need to restart only if any of the following events occur:

- You have created a new profile and need to assign the new profile to the Gateway.
- You have modified some attributes in the existing profile and need the changes to take effect.
- Gateway crashes due errors such as OutOfMemory error.
- Gateway stops responding and does not service any requests.

Configuring the Gateway Watchdog

You can configure the time interval at which the watchdog monitors the status of the Gateway. To start or to stop the watchdog, run the command;`./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off`. This time interval is set to 60 seconds by default. To change this value, edit the following line in the crontab utility:

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

See the `crontab` man page to configure the `crontab` entries.

Specifying a Virtual Host

A virtual host is an additional host name that points to the same machine IP and a host name. For example, if a host name `abc` points to the host IP address 192.155.205.133, you can add another host name `cde` which points to the same IP address.

Specifying a Proxy to Contact Access Manager

You can specify a proxy host to be used by the Gateway to contact SRA Core (RemoteConfigServlet) that is deployed over the Portal Server. This proxy is used by the Gateway to reach the Portal Server and Access Manager. See, "To Specify a Proxy" on page 243.

Understanding the platform.conf File

The `platform.conf` file is located by default at: `/etc/opt/SUNWportal`.

The `platform.conf` file contains the details that the Gateway needs. This section provides a sample `platform.conf` file and describes all the entries.

The advantage of including all the machine-specific details in the configuration file is that a common profile can be shared by Gateways running on multiple machines.

The following is a sample of the `platform.conf` file.

```
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
```

The following table lists and describes all the fields in the platform.conf file.

TABLE 2–1 File Properties

Entry	Default Value	Description
`gateway.user`	noaccess	The Gateway runs as this user. The Gateway must be started as root and after initialization, it loses its root privileges to become this user.
`gateway.jdk.dir`		This is the location of the JDK directory that the Gateway uses.
`gateway.dsame.agent`		This is the URL of the Access Manager that the Gateway contacts while starting up to get its profile.

`portal.server.protocol`
`portal.server.host`
`portal.server.port`
		This is the protocol, host and port that the default Portal Server installation is using.

`gateway.protocolgateway.hostgateway.port`		This is the Gateway protocol, host and port. These values are the same as the mode and port that you specified during installation. These values are used to construct the notification URL.
`gateway.trust_all_server_certs`	true	This indicates whether the Gateway has to trust all server certificates, or only those that are in the Gateway certificate database.
`gateway.trust_all_server_cert_domains`	false	When an SSL communication is between the Gateway and a server, a server certificate is presented to the Gateway. By default, the Gateway checks if the server host name is the same as the server certificate CN. If this attribute value is set to true, the Gateway disables the domain check for the server certificate that it receives.
`gateway.virtualhost`		If the Gateway machines has multiple hostnames configured, you can specify a different name and identity provider address in this field.

TABLE 2–1 File Properties *(Continued)*

Entry	Default Value	Description

`gateway.virtualhost.defaultOrg=org`		This specifies the default Org to which the user logs into.
		For example, suppose the virtual host field entries are the following:
		`gateway.virtualhost=test.com employee.test.com`
		`Managers.test.com`
		with the default org entries as:
		`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`
		The user can use `https://manager.test.com` to log into the manager's org instead of `https://test.com/o=Manager,dc=test,dc=com`
		Note – virtualhost and defaultOrg are case sensitive in the `platform.conf file`, but not when using it in the URL.

`gateway.notification.url`		A combination of the Gateway host, protocol and port is used to construct the notification URL. This is used to receive session notification from the Access Manager. Ensure that the notification URL is not the same as any organization name. If the notification URL matches an organization name, a user trying to connect to that organization gets a blank page instead of the login page.
`gateway.retries`		This is the number of times that the Gateway tries to contact the Portal Server while starting up.

TABLE 2–1 File Properties *(Continued)*

Entry	Default Value	Description
`gateway.debug`	`error`	This sets the debug level of the Gateway. The debug log file is located at *debug-directory*`/files`. The debug file location is specified in the `gateway.debug.dir` entry. The debug levels are: ■ error - Only serious errors are logged in the debug file. The Gateway usually stops functioning when such errors occur. ■ warning - Warning messages are logged. ■ message - All debug messages are logged. ■ on - All debug messages are displayed on the console. The debug files are: `srapGateway.`*gateway-profile-name* - Contains the Gateway debug messages. `Gateway_to_from_server.`*gateway-profile-name* - In message mode, this file contains all the requests and response headers between the Gateway and internal servers. To generate this file, change the write permission on `/var/opt/SUNWportal/debug` directory. `Gateway_to_from_browser.`*gateway-profile-name* - In message mode, this file contains all the requests and response headers between the Gateway and the client browser. To generate this file, change the write permission on `/var/opt/SUNWportal/debug` directory.
`gateway.debug.dir`		This is the directory where all the debug files are generated. This directory should have sufficient permissions for the user mentioned in `gateway.user` to write to files.
`gateway.logdelimiter`		Not used currently.
`gateway.external.ip`		In case of a multi-homed Gateway machine (one with multiple IP addresses), you need to specify the external IP address here. This IP is used for Netlet to run FTP.
`gateway.certdir`		This specifies the location of the certificate database.
`gateway.allow.client.caching`	true	Allow or disallow client caching. If allowed, client browsers can cache static pages and images for better performance (by reduced network traffic). If disallowed, nothing is cached and security is higher but performance drops with the higher network load.

TABLE 2–1 File Properties *(Continued)*

Entry	Default Value	Description
`gateway.userProfile.cacheSize`		This is the number of user profile entries that get cached at the Gateway. If the number of entries exceeds this value, frequent retries occur to cleanup the cache.
`gateway.userProfile.cacheSleepTime`		Sets the sleep time, in seconds, for the cache cleanup.
`gateway.userProfile.cacheCleanupTime`		The maximum time in seconds after which a profile entry can get removed.
`gateway.bindipaddress`		On a multihomed machine, this is the IP address to which the Gateway binds its serversocket. To configure the Gateway to listen to all interfaces, replace the IP address so that the `gateway.bindipaddress=0.0.0.0`
`gateway.sockretries`	3	Not used currently.
`gateway.enable.accelerator`	false	If set to true external accelerator support is allowed.
`gateway.enable.customurl`	false	If set to true the administrator is allowed to specify a custom URL for the Gateway to rewrite pages to.
`gateway.httpurl`		The HTTP reverse proxy URL for a custom URL for the Gateway to rewrite pages to. When Proxylet is enabled use this entry.
`gateway.httpsurl`		The HTTPS reverse proxy URL for a custom URL for the Gateway to rewrite pages to. Do not use this entry if Proxylet is enabled.
`gateway.favicon`		The URL to which the Gateway redirects requests for the `favicon.icon` file. This is used for the "favorite icon" in Internet Explore and Netscape 7.0 and higher. If left empty, the Gateway sends a 404 not found message back to browser.
`gateway.logging.password`		The LDAP password of the user `amService-srapGateway` that gateway uses for creating its application session. This can be either encrypted or in plain text.
`http.proxyHost`		This proxy host is used to contact the Portal Server.
`http.proxyPort`		This is the port for the host used to contact Portal Server.
`http.proxySet`		This property is set to true if a proxy host is required. If the property is set to false, `http.proxyHost` and `http.proxyPort` are ignored.

TABLE 2–1 File Properties *(Continued)*

Entry	Default Value	Description
`portal.server.`*`instance`*		The value of this property is the corresponding `/etc/opt/SUNWam/config/AMConfig-`*instance-name*.properties file. If the value is default, then it points to `AMConfig.properties.`
`gateway.cdm.cacheSleepTime`	60000	The time out value for cache Client Detection Module responses sent to the Gateway from the Access Manager.
`gateway.cdm.cacheCleanupTime`	300000	The time out value for cache Client Detection Module responses sent to the Gateway from the Access Manager.
`netletproxy.port`	10555	The Netlet Proxy deamon listens for requests on this port.
`rewriterproxy.port`	10555	The Rewriter Proxy deamon listens for requests on this port.
`gateway.ignoreServerList`	false	If set to true, the Access Manager server URL is constructed using the values specified in the `AMConfig.properties` file. Set this property to true when the Access Manager server is behind a load balancer.
`rewriterproxy.accept.from.gateways`		This is a list of IP addresses from which the Rewriter Proxy can be made to accept requests from. This works in HTTP and HTTPS modes both. This is for added security, only requests coming from this set is accepted and all other requests are not handled. This can be comma separated IP addresses. Default value is empty which is treated as legacy mode, i.e all requests coming to Rewriter Proxy are honored.
`rewriterproxy.checkacl=`	false	With this property enabled Rewriter Proxy can be made to check ACL values just like the Gateway. The legacy mode value is "false". When set to true, the Rewriter Proxy will check the URL against the values specified in the gateway access service, at the given DN and will allow/deny requests as per the list set there. This value is useful both in HTTP and HTTPS modes.

Using Web Proxies

You can configure the Gateway to contact HTTP resources using third party web proxies. Web proxies reside between the client and the Internet.

Web Proxy Configuration

Different proxies may be used for different domains and subdomains. These entries tell the Gateway which proxy to use to contact specific subdomains in specific domains. The proxy configuration specified in the Gateway works as follows:

- Creates a list of domains and subdomains along with the required proxies in the Proxies for Domains and Subdomains field in the Gateway service.

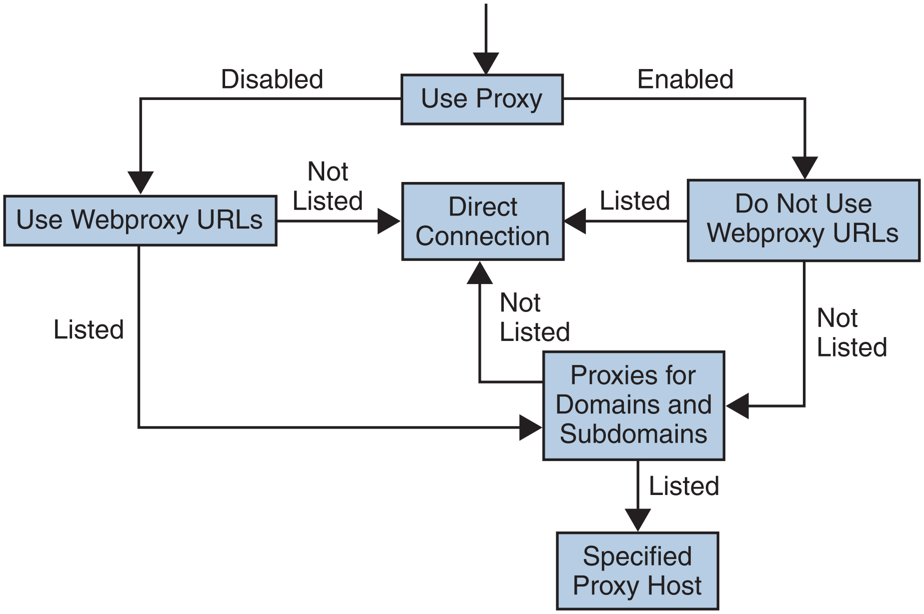
- With the Use Proxy option enabled:
  - The proxies specified in the Proxies for Domains and Subdomains field are used for the specified hosts.
  - To enable direct connections for certain URLs within the domains and subdomains specified in the Proxies for Domains and Subdomains list, specify these URLs in the Do Not Use Web Proxy URLS field.

With the Use Proxy option disabled:

- To ensure that proxies are used for certain URLs within the domains and subdomains specified in the Proxies for Domains and Subdomains field, specify these URLs in the Use Webproxy URLs list.

  Although the Use Proxy option is disabled, a proxy is used to connect to the URLs listed under Use Webproxy URLs. The proxies for these URLs are obtained from the Proxies for Domains and Subdomains list.

  The following illustration shows how the web proxy information is resolved based on the proxy configuration in the Gateway service.



In “Web Proxy Configuration” on page 40, if Use Proxy is enabled, and the requested URL is listed in the Do Not Use Webproxy URLs list, the Gateway connects to the destination host directly.

If Use Proxy is enabled, and the requested URL is not listed in the Do Not Use Webproxy URLs list, the Gateway connects to the destination host through the specified proxy. The proxy, if specified, is looked up in the Proxies for Domains and Subdomains list.

If Use Proxy is disabled, and the requested URL is listed in the Use Webproxy URLs list, the Gateway connects to the destination host using the proxy information in the Proxies for Domains and Subdomains list.

If Use Proxy is disabled, and the requested URL is not listed in the Use Webproxy URLs list, the Gateway connects to the destination host directly.

If none of the above conditions are met, and a direct connection is not possible, the Gateway displays an error saying that connection is not possible.

Note – If you are accessing the URL through the Bookmark channel of the standard Portal Desktop, and none of the above conditions are met, the Gateway sends a redirect to the browser. The browser accesses the URL using its own proxy settings.

Syntax

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

Example

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

* is a wild card that matches everything

where,

`sesta.com` is the domain name and `wp1` is the proxy to contact on port 8080.

`red` is a subdomain and `wp2` is the proxy to contact on port 8080.

`yellow` is a subdomain. Since no proxy is specified, the proxy specified for the domain is used, that is, `wp1` on port 8080.

* indicates that for all other subdomains `wp3` needs to be used on port 8080.

Note – Port 8080 is used by default if you do not specify a port.

Processing the Web Proxy Information

When a client tries to access a particular URL, the host name in the URL is matched with the entries in the Proxies for Domains and Subdomains list. The entry that matches the longest suffix of the requested host name is considered. For example, suppose that the requested host name is `host1.sesta.com`. The following searches occur in order until a match is found.

- The Proxies for Domains and Subdomains is scanned for host1.sesta.com. If a matching entry is found, the proxy specified against this entry is used to connect to this host.
- Else, the list is scanned for *.sesta.com. If an entry is found, the corresponding proxy is used.
- Else, the list is searched for sesta.com. If an entry is found, the corresponding proxy is used.
- Else, the list is searched for *.com. If an entry is found, the corresponding proxy is used.
- Else the list is searched for com. If an entry is found, the corresponding proxy is used.
- Else the list is searched for *. If an entry is found, the corresponding proxy is used.
- If no matching centers are found, a direct connection is attempted.

Consider the following entries in the Proxies for Domains and Subdomains list:

```
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
```

The Gateway internally maps these entries into a table as shown in the following table.

TABLE 2–2 Mapping of Entries in the Proxies for Domains and Subdomains List

Number	Entry in Proxies for Domains and Subdomains List	Proxy	Description
1	com	p1	As specified in the list.
2	host1.com	p2	As specified in the list.
3	host2.com	p1	The proxy for the domain is used because no proxy is specified for host2.
4	*.com	p3	As specified in the list.
5	sesta.com	p4	As specified in the list.
6	host5.sesta.com	p5	As specified in the list.
7	*.sesta.com	p6	As specified in the list.
8	florizon.com	Direct	See the description for entry 14 for details.
9	host6.florizon.com	–	See the description for entry 14 for details.

TABLE 2–2 Mapping of Entries in the Proxies for Domains and Subdomains List *(Continued)*

Number	Entry in Proxies for Domains and Subdomains List	Proxy	Description
10	abc.sesta.com	p8	As specified in the list.
11	host7.abc.sesta.com	p7	As specified in the list.
12	host8.abc.sesta.com	p8	As specified in the list.
13	*.abc.sesta.com	p9	As specified in the list. For all hosts other than `host7` and `host8` under the `abc.sesta.com` domain, `p9` is used as the proxy.
14	host6.florizon.com	p10	This entry is the same as entry 9. Entry 9 indicates a direct connection, whereas this entry indicates that proxy p10 should be used. In a case with two entries such as this, the entry with the proxy information is considered as the valid entry. The other entry is ignored.
15	host9.sesta.com	p11	As specified in the list.
16	siroe.com	Direct	A direct connection is attempted because no proxy is specified for `siroe.com,.`
17	host12.siroe.com	p12	As specified in the list.
18	host13.siroe.com	p13	As specified in the list.
19	host14.siroe.com	Direct	A direct connection is attempted because no proxy is specified for `host14`.
20	*.siroe.com	p14	See the description for entry 23.
21	host15.siroe.com	p15	As specified in the list.
22	host16.siroe.com	Direct	A direct connection is attempted because no proxy is specified for host16 or `siroe.com`.
23	*.siroe.com	p16	Similar to entry 20, but the proxies specified are different. In such a case, the exact behavior of the Gateway is not known. Either of the two proxies may be used.
24	*	p17	If no other entry matches the requested URL, `p17` is used as the proxy.

Tip – Instead of separating the proxy entries in the Proxies for Domains and Subdomains list with the | symbol, you can place individual entries on separate lines in the list. For example, instead of an entry such as:

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

you can specify this information as:

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

This list format makes it easier to track repeated entries or any other ambiguities.

Rewriting Based on the Proxies for Domains and Subdomains List

The entries in the Proxies for Domains and Subdomains list are also used by Rewriter. Rewriter rewrites all URLs whose domains match the domains listed in the Proxies for Domains and Subdomains list.

Caution – The * entry in the Proxies for Domains and Subdomains list is not considered for rewriting. For example, entry 24 is not considered.

For information on Rewriter, see Chapter 4, “Working with Rewriter” .

Default Domain and Subdomain

When the destination host in the URL is not a fully qualified host name, the default domain and subdomain are used to arrive at the fully qualified name.

Assume that the entry in the Default Domains field of the administration console is:

```
red.sesta.com
```

Note – You need to have the corresponding entry in the Proxies for Domains and Subdomains list.

In the example above, `sesta.com` is the default domain and the default subdomain is `red`.

If the requested URL is `host1`, this entry is resolved to `host1.red.sesta.com` using the default domain and subdomain. The Proxies for Domains and Subdomains list is then checked for `host1.red.sesta.com`.

Using Automatic Proxy Configuration

To ignore the information in the Proxies for Domains and Subdomains list, enable the Automatic Proxy Configuration feature.

When using a Proxy Auto Configuration (PAC) file:

- Portal Server, Gateway, Netlet, and Proxylet use *Rhino* software to parse the PAC file. You can install the `SUNWrhino` package from the Java Enterprise System Accessory CD.

  This package contains the `js.jar` file which must be present in the `/usr/share/lib directory`. Add this directory to the `webserver/appserver` class path on the Gateway and Portal Server machine, otherwise the Portal Server, Gateway, Netlet, and Proxylet cannot parse the PAC file.
- The `js.jar` must be present in the `$JRE_HOME/lib/ext` directory on the Gateway machine, otherwise the Gateway cannot parse the PAC file.
- The Gateway fetches the PAC file at bootup from the location specified in the gateway profile Automatic Proxy Configuration File location field.
- The Gateway uses the URLConnection API to reach this location. If the proxy needs to be configured to reach the Gateway, the proxy needs to be configured in the following way:
  1. From the command-line, edit the following file:

     `/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*
  2. Add the following entries:

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

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

     `http.proxySet=true`
  3. Restart the Gateway to use the specified proxy:

     `./psadmin start-sra-instance –u amadmin – f` <*password file*> `–N` <*profile name*>`– t` <*gateway*>
- If PAC file initialization fails, then the Gateway uses the information in the Proxies for Domains and Subdomains list.
- If "" (empty string) or "null" is returned from the PAC file, then the Gateway assumes that the host does not belong to the intranet. This is similar to the host not being in the Proxies for Domains and Subdomains list.

  If you want the Gateway to use a direct connection to the host, return "DIRECT". See "Example with Either DIRECT or NULL Return" on page 47.
- Gateway only uses the first proxy returned when multiple proxies are specified. It does not try to failover or loadbalance among the various proxies specified for a host.
- Gateway ignores SOCKS proxies and attempts a direct connection and assumes that the host is part of the intranet.

- To specify a proxy to be used to reach any host not part of the intranet, use the proxy type STARPROXY. This proxy type is an extension of the PAC file format and is similar to the entry * `proxyHost:port` in Proxies for Domains and Subdomains section of the gateway profile. See "Example with STARPROXY Return" on page 47

Sample PAC File Usage

The following examples show the URLs listed in the Proxies for Domains and Subdomains list and the corresponding PAC file.

Example with Either DIRECT or NULL Return

If these proxies are used for domains and subdomains:

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

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

the corresponding PAC file is:

```
// 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
```

Example with STARPROXY Return

If these proxies are used for domains and subdomains:

```
intranet1.com

intranet2.com.proxy.intranet1.com:8080

internetproxy.intranet1.com:80
```

the corresponding PAC file is:

```
// 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
```

In this case, if the request is for a host in `.intranet2.com` `domain`, the Gateway contacts `proxy.intranet1.com:8080`. If proxy.`intranet1.com:8080` is down, the request fails. The Gateway does not failover and contacts `proxy1.intranet1.com:8080`.

Specifying PAC File Location

The format for specifying the location of the PAC file depends upon it's location as follows:

- If the PAC file resides on a Web server, the PAC URL is:

  `http://`hostname/*pacfile_name*`.pac`
- If the pacfile is a local file (for example, `c:\\pacfile\\sample.pac`), for Java 1.4.1_x, enter the PAC URL as:

  `file://c:/pacfile/sample.pac`
- If the PAC file is a local file (for example, `c:\\pacfile\\sample.pac`), for Java 1.4.2_x, enter the PAC URL as:

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

Adding Services in Separate Sessions

When you add Portal Server services in separate sessions:

- List Portal Servers under Gateway > Core in the Portal Server administration console.
- List Portal Server URLs in the Non-authenticated URLs under Gateway > Security.

Using a Netlet Proxy

Netlet packets are decrypted at the Gateway and sent to the destination servers. However, the Gateway needs to access all Netlet destination hosts through the firewall between the demilitarized zone (DMZ) and the intranet. This setup requires opening a large number of ports in the firewall. The Netlet proxy can be used to minimize the number of open ports in the firewall.

The Netlet proxy enhances the security between the Gateway and the intranet by extending the secure tunnel from the client, through the Gateway to the Netlet proxy that resides in the intranet. With the proxy, Netlet packets are decrypted by the proxy and then sent to the destination.

The advantages of using Netlet proxy are:

- Adds an additional layer of security.
- Minimizes the use of extra IP addresses and ports from the Gateway through an internal firewall in a significantly sized deployment environment.
- Restricts the number of open ports between the Gateway and the Portal Server to 1. This port number can be configured during installation.
- Extends the secure channel between the client and the Gateway, up to the Portal Server as shown in the "With a Netlet Proxy Configured" section of "Using a Netlet Proxy" on page 48. The Netlet proxy offers improved security benefits through data encryption but may increase the use of system resources. See the Sun Java System Installation Guide for information on installing the Netlet proxy.

You can perform the following tasks:

- Install the Netlet proxy on the Portal Server node or on a separate node.
- Install multiple Netlet proxies and configure them for a single Gateway using the administration console. This is useful in load balancing.
- Configure multiple instances of the Netlet proxy on a single machine.
- Point multiple instances of the Gateway to a single installation of the Netlet proxy.
- Tunnel Netlet through a web proxy.

Shows three sample implementations of the Gateway and the Portal Server with and without a Netlet proxy installed. The components include a client, two firewalls, the Gateway that resides between the two firewalls, Portal Server, and Netlet destination servers.

The first scenario shows the Gateway and Portal Server without a Netlet proxy installed. The data encryption extends only from the client to the Gateway. A port is opened in the second firewall for each Netlet connection request.

The second scenario shows the Gateway and Portal Server with a Netlet proxy installed on Portal Server. The data encryption extends from the client all the way to the Portal Server. Since all Netlet connections are routed through a Netlet proxy, only one port needs to be opened in the second firewall for Netlet requests.

The third scenario shows the Gateway and the Portal Server with a Netlet proxy installed on a separate node. Installing a Netlet proxy on a separate node reduces the load on the Portal Server node. Again, only two ports need to be opened in the second firewall. One port services requests to the Portal Server, and the other port routes Netlet requests to the Netlet proxy server.

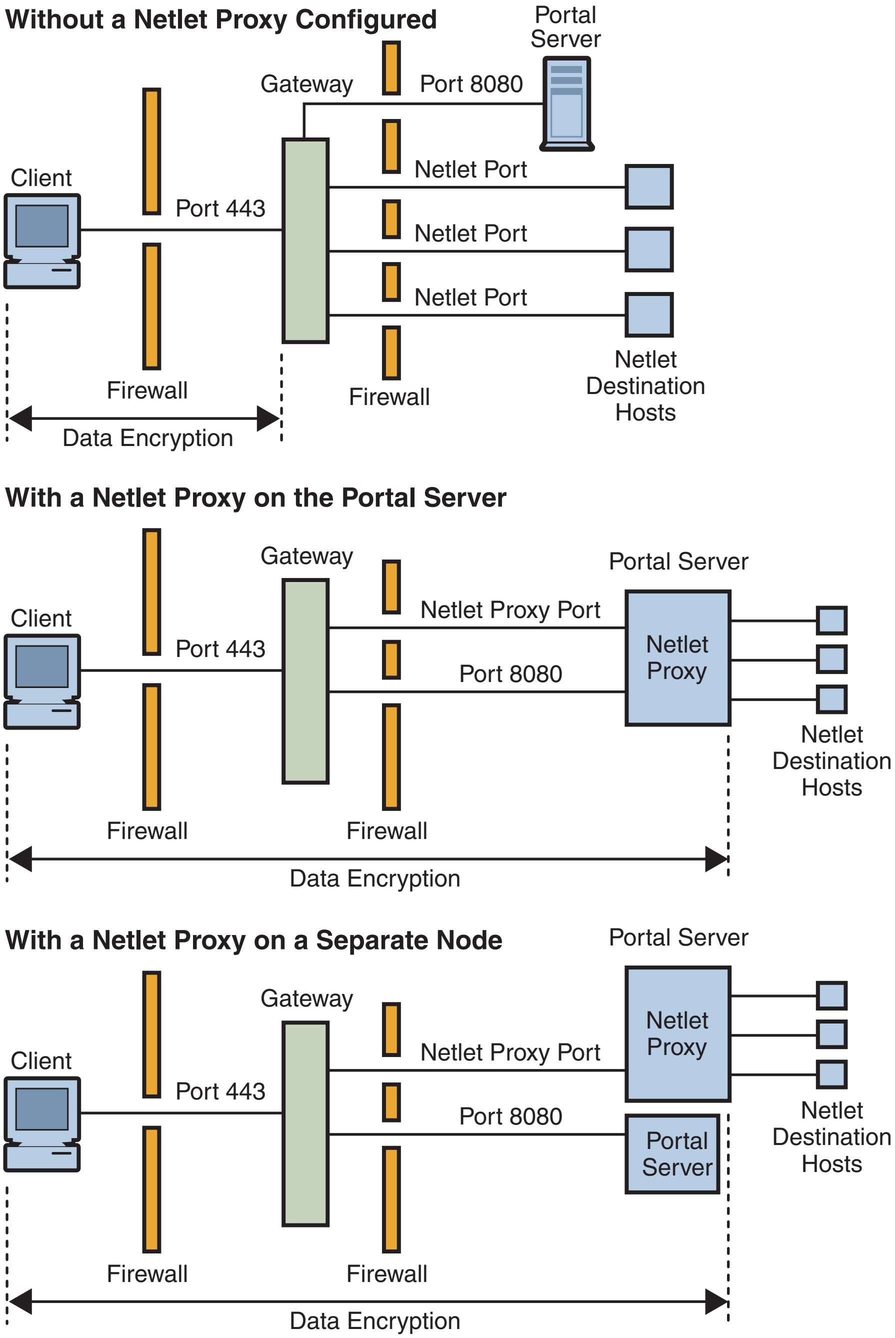


FIGURE 2–1 Implementation of Netlet Proxy

Enabling a Netlet Proxy

You enable a Netlet proxy through the Gateway service using the Portal Server administration console.

Restarting a Netlet Proxy

You can configure a Netlet proxy to restart whenever the proxy is killed accidentally. You can schedule a watchdog process to monitor a Netlet proxy and restart it if it goes down.

You can also restart a Netlet proxy manually. See "To Restart a Netlet Proxy" on page 244 for steps.

To Configure a Netlet Proxy Watchdog

You can configure the time interval at which the watchdog monitors the status of a Netlet proxy. This time interval is set to 60 seconds by default. To change this interval, add the following line to the `crontab` file:

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

Note – To start or to stop the watchdog, run the command;`./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off`.

Using a Rewriter Proxy

The Rewriter proxy is installed in the intranet. Instead of trying to retrieve the contents directly, the Gateway forwards all requests to the Rewriter proxy which fetches and returns the contents to the Gateway.

The advantages of using a Rewriter proxy are:

- If a firewall exists between the Gateway and server, the firewall needs to open only two ports - one between the Gateway and Rewriter proxy, and another between the Gateway and the Portal Server.
- HTTP traffic is secure between the Gateway and the intranet even if the destination server only supports the HTTP protocol and not HTTPS.

If you do not specify a Rewriter proxy, the Gateway component makes a direct connection to intranet computers when a user tries to access one.

If you are using the Rewriter proxy as a load balancer, be sure that the `platform.conf.instance_name` for Rewriter points to the load balancer URL. Also specify the load balancer host in the Portal Servers list.

If you have multiple instances of Rewriter proxies for each Gateway instance, which are not necessarily on the portal node, provide the details for each Rewriter proxy in the form of *host-name:port i*n the `platform.conf` file, rather than a single port entry for the Rewrite proxy.

Creating Instances of a Rewriter Proxy

Use the `rwpmultiinstance` script to create a new instance of a Rewriter proxy on the Portal Server node. Run this script after the gateway profile has been created.

See "To Create a Rewriter Proxy Instance" on page 245.

Enabling a Rewriter Proxy

Enable a Rewriter proxy through the Gateway service under SRA Configuration in the Access Manager administration console.

Restarting a Rewriter Proxy

You can configure to restart Rewriter proxy whenever the proxy is killed accidentally. You can schedule a watchdog process to monitor and restart it if this happens.

You can also restart a Rewriter proxy manually.

See "To Restart a Rewriter Proxy" on page 245.

Configuring a Rewriter Proxy Watchdog

You can configure the time interval at which the watchdog monitors the status of the Rewriter proxy. This time interval is set to 60 seconds by default. To to change the time interval, add the following line in the `crontab` file:

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

Note – To start or to stop the watchdog, run the command;`./psadmin sra-watchdog -u amadmin -f <password-file> -t <type> on|off`.

Using a Reverse Proxy with the Gateway

A proxy server serves Internet content to the intranet, while a reverse proxy serves intranet content to the Internet. You can configure deployments of reverse proxies to achieve load balancing and caching.

If the deployment has a third-party reverse proxy in front of the Gateway, the response has to be rewritten with the reverse proxy's URL instead of the Gateway's URL. For this, the following configurations are needed.

See "To Enable a Reverse Proxy" on page 246.

Obtaining Client Information

When the Gateway forwards a client request to any internal server, it adds HTTP headers to the HTTP request. You can use these headers to obtain additional client information and detect the presence of the Gateway.

To view the HTTP request headers, set the entry in the `platform.conf` file to `gateway.error=message`. And, then use the `request.getHeader()` from the servlet API. The following table lists the information in the HTTP headers.

TABLE 2–3 Information in HTTP Headers

Header	Syntax	Description
PS-GW-PDC	`X-PS-GW- PDC: true/false`	Indicates whether PDC is enabled at the Gateway.
PS-Netlet	`X-PS-Netlet:enabled=true/false`	Indicates whether Netlet has been enabled or disabled at the Gateway. If Netlet is enabled, then the encryption option is populated, indicating whether the Gateway is running in HTTPS (`encryption=ssl`) or in HTTP mode (`encryption=plain`) For example: ■ `PS-Netlet: enabled=false` Netlet is disabled. ■ `PS-Netlet: enabled=true; encryption=ssl` Netlet is enabled with the Gateway running in SSL mode. The `encryption=ssl` or `encryption=plain`is not populated when Netlet is not enabled.

TABLE 2–3 Information in HTTP Headers *(Continued)*

Header	Syntax	Description
PS-GW-URL	`X-PS-GW-URL: http(s)://gatewayURL(:port)`	Indicates the URL that the client is connected to. When the port is non-standard, for example, if the Gateway is in HTTP/HTTPS mode and the port is not 80/443, then the `:port` is also populated.
PS-GW-Rewriting-URL	`X-PS-GW-URL: http(s)://gatewayURL(:port)/[SessionInfo]`	Indicates the URL that the Gateway rewrites all the pages to. 1. When the browser supports cookies, the value of this header is the same as the PS-GW-URL header. 2. When the browser does not support cookies: ■ the destination host is in the "User Session to which User Session Cookie is Forwarded" field, the value is the actual URL to which the Gateway rewrites the page to (including the encoded SessionID information). ■ the destination host is not in the User Session to which User Session Cookie is Forwarded field, then the `SessionInfo` string is `$SessionID` **Note –** As part of the response, if the user's Access Manager sessionId changes (like response from authentication page) then the pages are rewritten with that value (and not the value that was previously indicated in the header). For example: ■ If the browser supports cookies: PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/ ■ If the browser does not support cookies and the endserver is in the User Session to which User Session Cookie is Forwarded field. PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/SessIDValCustomEncodedValue/ ■ If the browser does not support cookies and endserver is not in User Session to which User Session Cookie is Forwarded field. PS-GW-Rewriting-URL: https://siroe.india.sun.com:10443/$SessionID
PS-GW-CLientIP	`X-PS-GW-CLientIP:` *IP*	Indicates the IP that the Gateway obtained from `recievedSocket.getInetAddress().getHostAddress()` This value provides the client's IP if directly connected to the Gateway.

Using Authentication Chaining

Authentication chaining provides a higher level of security than the regular mechanism of authentication. You can enable users to be authenticated against more than one authentication mechanism.

The procedure described in this is only for enabling authentication chaining along with a Personal Digital Certificate (PDC) authentication at the Gateway. For information on authentication chaining without PDC authentication at the Gateway, see the *Access Manager Administration Guide*.

For example, if you chain the PDC and Radius authentication modules, the user will have to authenticate against all three modules to access the standard Portal Desktop.

See "To Add Authentication Modules to an Existing PDC Instance" on page 246 for steps.

Note – When enabled, PDC is always the first authentication module to be presented to the user.

Using Wild Card Certificates

A wild card certificate accepts a single certificate with a wild card character in the fully-qualified DNS name of the host.

With the certificate multiple hosts within the same domain are secured. For example, a certificate for `*.domain.com` can be used for `abc.domain.com`, and `abc1.domain.com`. This certificate is valid for any host in the `domain.com` domain.

Disabling Browser Caching

Because the Gateway component provides secure access to back end corporate data from any location using just a web browser, the information should not be cached locally by the client.

You can disable caching of pages redirected through the Gateway by modifying the attribute in the `platform.conf` file of the specific Gateway.

Disabling this option can have an impact on the Gateway performance. Every time the standard Portal Desktop is refreshed, the Gateway has to retrieve everything referenced by the page, such as images that might have been previously cached by the browser. However, enabling this feature, means that remotely accessing secure content will not leave a cached footprint on the client site. This factor could outweigh performance implications if the corporate network is being accessed from an Internet cafe or similar remote location that is not under corporate IT control.

See "To Disable Browser Caching" on page 247.

Customizing the Gateway Service User Interface

This section discusses the various Gateway property files that can be edited.

Modifying the srapGateway.properties File

You can edit this file for the following purposes:

- Customize the error messages that might appear when the Gateway is running.
  - HTML-CharSets=ISO-8859-1 specifies the character set that was used to create this file.
  - The number in braces (for example, {0}) indicates that the value displayed at run time. You can change the label associated with this number, or rearrange the labels as required. Ensure that the label corresponds to the message that to be displayed since the number and the message are associated.

  Customize the log information.

  By default the `srapGateway.properties` file is located under the *portal-server-install-root*/`SUNWportal/locale` directory. All messages that appear on the Gateway machine are located in this file, irrespective of the language of the messages.

  To change the language of the messages that appear on the client standard Portal Desktop, copy this file into the respective locale directory, for example *portal-server-install-root*/`SUNWportal/locale_en_US`.

Modifying the srapgwadminmsg.properties File

You can edit this file for the following reasons:

- Customize the labels that appear on buttons for the Gateway service on the administration console.
- Customize the status messages and error messages that appear when you are configuring the Gateway.

Sharing LDAP Directories

When two instances of Portal Server and Access Manager servers share the same LDAP directories, it shares the same LDAP directories for all subsequent instances of Portal Servers, Access Managers, and Gateways. See “To Share LDAP Directories” on page 247.

CHAPTER 3

Working With Proxylet

This chapter describes Proxylet which enables users to access intranet web pages through the Gateway without parsing the web pages.

Working with Proxylet

Overview of Proxylet

Proxylet is a Java applet that sets itself as a proxy server on the client machine. Proxylet reads and modifies the proxy settings in the Proxy Auto Config (PAC) file on the client machine so that the proxy settings point to the local proxy server (Proxylet).

Proxylet inherits the transport mode from the Gateway. If the Gateway is configured to run on SSL, Proxylet establishes a secure channel between the client machine and the Gateway or destination server. For encryption, Proxylet uses the JSSE API if the client JVM is 1.4 or higher or if the required jar files reside on the client machine. Otherwise it uses the KSSL API. Decryption occurs on the client machine.

The domain and subdomain for URLs that are to be directed to the Gateway are specified in the gateway profile. If a URL is not part of a domain that the gateway handles, the request is directed to the Internet. If a particular URL domain is listed in the gateway profile, then Proxylet resets the client proxy settings to point to the Gateway.

Proxylet supports client-side authentication if a Personal Digital Certificate (PDC) is enabled at the Gateway. To check whether PDC is enabled, see "Obtaining Client Information" on page 54.

Proxylet is enabled from the Portal Server administration console where the client IP address or proxy host name and port are specified. If Proxylet is enabled, it checks the client machine for the following information:

- Appropriate browser permissions
- Whether the browser is IE 6.0 sp2, IE 7, and Firefox 2.0
- Whether the machine or device can run a server application

If all the requirements are satisfied, an applet is downloaded and launched on the client machine. When the client does not have JRE 1.4.2 or later installed, then JRE is automatically downloaded with Proxylet if you have both internet connectivity and administration privileges.

When Proxylet is used, the proxy settings are retrieved from the Proxy Auto Configuration (PAC) file or from the proxy configuration list.

Note – Make sure users know that when using the Proxylet applets, browser pop-up blockers must be disabled.

HTTPS Support

Proxylet supports HTTPS with the following results:

- Decryption is done at the client server.
- Destination servers can accessed when running in SSL mode.
- Client certificates are directly presented to the destination server.
- Basic authentication single-sign (SSO) is not supported at the gateway. (The Gateway can not insert SSO information in http headers.)
- URL-based access control is not supported, only host-based access control.
- External accelerators and external Reverse proxies in front of the gateway are not currently supported.

Note – This support is not for Proxylet when Portal Server uses HTTPS.

Advantages of Using Proxylet

Unlike Rewriter, Proxylet requires little or no postinstallation changes. Integration with third party software such as Microsoft Exchange Server is easy. Also the performance of the Gateway increases because Proxylet does not touch web content. Because Proxylet does not modify content or change the data, users can download any type of content, such as `tar` and `gzip` files.

Configuring Proxylet

For information on enabling and configuring, Proxylet, see Chapter 13, “Configuring Proxylet.”

Note – If the user does not have the appropriate Java Virtual Machine (JVM) to run Proxylet, the browser connects to the sun web site to download the Java Runtime Environment. If the user's browser settings do not contain the correct values or if the user is using direct proxy settings without access to the Internet, then Proxylet cannot be downloaded.

◆ ◆ ◆ CHAPTER 4

Working with Rewriter

The Rewriter component of Secure Remote Access enables users to access intranet web pages through the Gateway by parsing the web pages.

This Chapter covers the following topics:

- “Character Set Encoding” on page 64
- “Rewriter Usage Scenarios” on page 64
- “Writing Rulesets” on page 65
- “Public Interface (RuleSet DTD)” on page 65
- “Troubleshooting Using Debug Logs” on page 94
- “Public Interface (RuleSet DTD)” on page 65
- “Working Samples” on page 97
- “Case Study” on page 123
- “Mapping of 6.x RuleSet with 3.0” on page 127

Introduction to Rewriter

The Rewriter component of Secure Remote Access enables end users to browse the intranet by modifying Uniform Resource Identifier (URI) references on web pages so that they point to the Gateway. A URI defines a way to encapsulate a name in any registered name space, and labels it with the name space. The most common kinds of URIs are Uniform Resource Locators (URLs). Rewriter supports only HTTP or HTTPS. This support is regardless of the capitalization of the protocol. Rewriter only supports backslashes when they are part of a relative URL.

EXAMPLE 4–1 Rewriting URLs

`http://abc.sesta.com\\index.html` is rewritten.

These URLs are not rewritten: `http:\\\\abc.sesta.com`. *http:/abc.com*

Character Set Encoding

HTTP standards require that HTTP headers or HTML meta tags specify a character set for web pages. However, sometimes this information is not available. The character set must be known so that encoding for the data is set and the data is displayed as intended by the creator.

To detect the character sets, install the `SUNWjchdt` package from the Java Enterprise System Accessory CD. If this product is installed, Rewriter will detect it and use it if necessary.

Note – Using this product can affect performance, so you should install it only when required. See the `jcharset_readme.txt` for details on installation, configuration and usage.

Rewriter Usage Scenarios

When a user tries to access intranet web pages through the Gateway, web pages are made available by using Rewriter. Rewriter is used by the URLScraper and the Gataway.

URLScraper

The URL Scraper provider gets content from configured URIs. Before sending these URIs to the browser, it expands all relative URIs to absolute URIs.

For example, if a user is trying to access a site as:

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

Rewriter translates this to:

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

where `http://yahoo.com/test/` is the base URL of the page.

See the *Sun Java SystemPortal Server Administration Guide* for details about the URLScraper provider.

Gateway

The Gateway obtains content from internet portals. Before sending the content to the browser, it prefixes the Gateway URI to the existing URI so that subsequent URI requests from the browser can reach the Gateway.

For example, a user who is trying to access an HTML page on an internet machine as:

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

Rewriter prefixes this URL with a reference to the Gateway as follows:

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

When the user clicks a link associated with this anchor, the browser contacts the Gateway. The Gateway fetches the content of `mypage.html` from `mymachine.intranet.com`.

The Gateway uses several rules to determine the elements of a fetched web page that will be rewritten.

Writing Rulesets

For more information about defining a ruleset, see the *Portal Server Administration Guide*. After creating a new ruleset, you need to define the required rules.

This section covers the following topics:

- "Public Interface (RuleSet DTD)" on page 65
- "Sample XML DTD" on page 67
- "Procedure to Write Rules" on page 68
- "Ruleset Guidelines" on page 68
- "Defining the RuleSet Root Element" on page 69
- "Using the Recursive Feature" on page 70
- "Rules for HTML Content" on page 70
- "Rules for JavaScript Content" on page 77
- "Rules for XML Content" on page 91
- "Rules for Cascading Style Sheets" on page 93
- "Rules for WML" on page 93

Public Interface (RuleSet DTD)

RuleSet 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 "*"
>
```

Note – You can use * as a part of rule value except that mandatory attribute values cannot be just *. Such rules are ignored, but the message is logged in the RuleSetInfo log file. For information on this log file, see "Debug File Names" on page 95.

Sample XML DTD

This section contains a sample rule set. The "Case Study," on page 140 is used to illustrate how these rules are interpreted by Rewriter.

```
<?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>
```

Procedure to Write Rules

The general procedure to write rules is:

- Identify the directories that contain the HTML pages whose content needs to be rewritten.
- In these directories, identify the pages that need to be rewritten.
- Identify the URLs that need to be rewritten on each page. An easy way identify most of the URLs is to search for "http" and "/".
- Identify the content type of the URL: HTML, JavaScript or XML.
- Write the rule required to rewrite each of these URLs by editing the required ruleset in the Rewriter service under Portal Server Configuration in the Access Manager administration console.
- Combine all the rules into a ruleset for that domain.

Ruleset Guidelines

When creating a ruleset, keep the following in mind:

- The order of precedence for specific hosts is based on the longest URI match. For example for the following rulesets

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

  `sfbay_ruleset` is used as it has the longest match.
- The rules in the ruleset are applied in order to each statement in the page until a rule matches a particular statement.

  While writing the rules, keep in mind the order of the rules. Rules are applied to the statements in a page, in the order in which they occur in the ruleset. If you have specific rules, and general rules that contain a "*", define the specific rules first, then the general rules. Otherwise, the general rule is applied to all statements, even before the specific rule is encountered.
- All rules need to be enclosed within the `<RuleSet> </RuleSet>` tags.
- Include all rules that need to rewrite HTML content in the `<HTMLRules> </HTMLRules>` section of the ruleset.
- Include all rules that need to rewrite JavaScript content in the `<JSRules> </JSRules>` section of the ruleset.
- Include all rules that need to rewrite XML content in the `<XMLRules> </XMLRules>` section of the ruleset.
- In your intranet pages, identify the URLs that need to be rewritten, and include the required rules in the appropriate sections (HTML, JSRules, or XMLRules) of the ruleset.
- Assign the ruleset to the required domain.
- Restart the Gateway to affect any changes:

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

Defining the RuleSet Root Element

The ruleset root element has two attributes:

- `RuleSetName`, for example, `default_ruleset.` This name is referenced in RuleSet to URI mapping.
- `Extends`. This attribute refers to the inheritance feature of rulesets. The value points to the ruleset from which you would like to derive a ruleset.

  Use the value `none` to signify that this new, independent ruleset does not depend on any other ruleset, or specify the *RuleSetName* to signify that your ruleset depends on another ruleset.

Using the Recursive Feature

Rewriter uses the recursive feature to search to the end of the matched string pattern for the same pattern.

For example, when Rewriter parses the following string:

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

the rule

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

rewrites only the first occurrence of the pattern, which would look like this:

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

If you use the recursive option

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

Rewriter searches to the end of the matched string pattern for the same pattern, so the output would be:

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

Defining Language Based Rules

Rules are based on the following languages:

- HTML
- JavaScript
- XML

Rules for HTML Content

HTML content in web pages can be further classified into attributes, forms and applets. Accordingly, the rules for HTML content are classified as:

- “Attribute Rules for HTML Content” on page 71
- “Form Rules for HTML Content” on page 73
- “Applet Rules for HTML Content” on page 74

Attribute Rules for HTML Content

This rule identifies the attributes of a tag whose value needs to be rewritten. The attribute values can be a simple URL, JavaScript, or DHTML content. For example:

- `src` attributes of an "img" tag point to an image location (simple URL)
- `onClick` attribute of a href attributes that handles on clicking of the link (DJS)

This section describes the following:

- "Attribute Rule Syntax" on page 71
- "Attribute Rule Example" on page 71
- "DJS Attribute Example" on page 72

Attribute Rule Syntax

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

where,

`attributeName` is the name of the attribute (mandatory)

`tag` is the tag to which the attribute belongs (optional, default * , meaning any tag)

`valuePatterns` See "Using Pattern Matching in Rules" on page 75.

`source` specifies the URI of the page in which this attribute is defined (optional, default * , meaning in any page)

`type` specifies the type of the value (optional). They can be:

`URL` - a simple URL (default value).

`DHMTL` - DHTML content. This kind of content is seen in standard HTML content and is used in Microsoft's HTC format files.

`DJS` - JavaScript content. All HTML event handlers such as onClick and onMouseover have JavaScript inlined with the HTML attribute.

Attribute Rule Example

Assume the base URL of the page is:

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

Page Content:

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

Rules

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

Output

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

Description

Because the URL to be rewritten is already an absolute URL, only the Gateway URL is prefixed to the URL.

DJS Attribute Example

Assume the base URL of the page is:

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

Page Content:

```
<Form>

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

</Form>
```

Rules

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

Output

<Form>

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

</Form>

Description

Two rules are required to rewrite the specified page content. The first rule identifies the `onClick` JavaScript token. The second rule identifies the parameter of the `check` function that needs to be rewritten. In this case, only the first parameter is rewritten because `paramPatterns` has the value `y` in place of first parameter.

The Gateway URL and the base URL of the page on which the JavaScript tokens appear are prefixed to the required parameter.

Form Rules for HTML Content

The HTML pages that a user browses may contain forms. Some form elements may take a URL as the value.

This section is divided into the following parts:

- "Form Rule Syntax" on page 73
- "Form Rule Example" on page 73

Form Rule Syntax

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

where

`name` is the name of the form (mandatory)

`field` is the field in the form whose value needs to be rewritten (mandatory)

`valuePatterns` See "Using Pattern Matching in Rules" on page 75

`source` is the URL of the html page where this form definition is present (optional, default `*`, meaning in any page)

Form Rule Example

Assume the base URL of the page is:

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

Page Content

Assume the page URI is `form.html` and is located in the root directory of the server.

```
<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>
```

To rewrite `/text.html` present in the value of hidden field named `abc1` which is part of `form1`. The following rules are needed.

Rules

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

Output

```
<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>
```

Description

The action tag is rewritten using some defined HTML attribute rule.

The input tag attribute value's value is rewritten as shown in the output. The specified valuePatterns is located, and all content following the matched valuePatterns is rewritten by prefixing the Gateway URL, and the base URL of the page. See “Using Pattern Matching in Rules” on page 75.

Applet Rules for HTML Content

A single web page may contain many applets, and each applet may contain many parameters. Rewriter matches the values specified in the rule with the HTML definition of the applet and modifies the URL values present as a part of the applet parameter definition. This replacement is carried out at the server and not when the user is browsing the particular web page. This rule identifies and rewrites the parameters in both the applet and object tags of the HTML content.

This section is divided into the following parts:

- “Applet Rule Syntax” on page 74
- “Applet Rule Example” on page 75

Applet Rule Syntax

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

where

code is the name of the applet or object class (mandatory)

param is the name of the parameter whose value needs to be rewritten (mandatory)

valuePatterns See “Using Pattern Matching in Rules” on page 75.

source is the URL of the page that contains the applet definition (optional, default is *, meaning, in any page)

Applet Rule Example

Assume the base URL of the page is:

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

Page Content:

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

Rules

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

Output

```
<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>
```

Description

The codebase attribute is rewritten because <Attribute name="codebase"/> is a defined rule in the default_gateway_ruleset.

All parameters whose names begin with Test are rewritten. The base URL of the page on which the applet code displays and the Gateway URL are prefixed to the value attribute of the param tag.

Using Pattern Matching in Rules

You can use the valuePatterns field to achieve pattern matching and identify the specific parts of a statement that need to be rewritten.

If you have specified valuePatterns as part of a rule, all the content that follows the matched pattern is rewritten.

Consider the sample form rule below.

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

where

`source` is the URL of the html page where the form displays.

`name` is the name of the form.

`field` is the field in the form whose value needs to be rewritten.

`valuePatterns` indicates the portion of the string that needs to be rewritten. All content appearing after `valuePatterns` is rewritten (optional, default "" means the full value needs to be rewritten).

Specifying Specialized Characters in valuePatterns

You can specify specialized characters by escaping them with a backslash. For example:

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

Using Wild Cards in valuePatterns

You can use the wildcard asterisk (*) character to achieve pattern matching for rewriting.

You cannot specify just an * in the `valuePatterns` field. Because * indicates a match with all text, no text follows the `valuePattern`. Therefore, Rewriter has no text to rewrite. You must use * in conjunction with another string such as `*abc`. In this case, all content that follows `*abc` is rewritten.

Note – An asterisk (*) can be used as a wildcard in any of the fields of the rule. However, all the fields in the rule cannot contain an *. If all fields contain a *, the rule is ignored. No error message is displayed.

You can use a * or ** along with the separation character (a semicolon or comma) that displays in the original statement to separate multiple fields. One asterisk (*) matches any field that is not to be rewritten, and two asterisks (**) to match any field that needs to be rewritten.

“Using Wild Cards in valuePatterns” on page 76 lists some sample usages of the * wildcard.

TABLE 4–1 Sample Usage of * Wildcard

URL	valuePatterns	Description
`url1, url2, url3, url4`	`valuePatterns = "**, *, **, *"`	`url1` and `url3` are rewritten because `**` indicates the portion to be rewritten
`XYZABChttp://host1.sesta.com/dir1.html`	`valuePatterns = "*ABC"`	only the portion `http://host1.sesta.com/dir1.html` is rewritten. Everything after `*ABC` needs to be rewritten.
`"0|dir1|dir2|dir3|dir4|test|url1`	`valuePatterns = "*|*|**|*|**|*|"`	`dir2`, `dir4` and `url1` are rewritten. The last field that needs to be rewritten does not have to be indicated by using `**`.

Rules for JavaScript Content

JavaScript can contain URLs in various locations. Rewriter cannot directly parse the JavaScript and determine the URL portion. A special set of rules need to be written to help the JavaScript processor to identify and translate the URL.

JavaScript elements with type URL are classified as follows:

- "Variables" on page 77
- "Function Arguments" on page 84

Variables

The generic syntax for variables is:

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

JavaScript variables can be subclassified into 5 categories depending on the type of value they hold:

- "URL Variables" on page 78
- "EXPRESSION Variables" on page 79
- "DHTML(Dynamic HTML) Variables" on page 80
- "DJS (Dynamic JavaScript) Variables" on page 82
- "SYSTEM Variables" on page 83

URL Variables

The variable value is a simple string which can be treated as a URL.

This section is divided into the following parts:

- "URL Variable Syntax" on page 78
- "URL Variable Example" on page 78

URL Variable Syntax

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

where

`variableName` is the name of the variable. The value of the `variableName` is rewritten (mandatory).

`type` is the URL variable (mandatory, and the value must to be a URL)

`source` is the URI of the page in which this JavaScript variable is found (optional, default is *, meaning in any page)

URL Variable Example

Assume the base URL is:

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

Page Content

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

Rules

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

Output

```
<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>
```

Description

All variables of type URL and name beginning with `imgsrc` are rewritten. For the first line of the output, the Gateway URL and the base URL of the page on which the variable displays are prefixed. The second line already contains the absolute path, and hence only the Gateway URL is prefixed. Third var `imagsrc2` would not be rewritten as it's value is not a string but another JavaScript value.

EXPRESSION Variables

Expression variables have an expression on the right hand side. The result of this expression is a URL. Rewriter appends a JavaScript function (`psSRAPRewriter_convert_expression`) to the HTML page as it cannot evaluate such expressions on the server. This function takes the expression as a parameter and evaluates it to the required URL at the client browser.

If you are not sure whether a statement contains a simple URL or an EXPRESSION URL, use EXPRESSION rules because it can handle both scenarios.

This section is divided into the following parts:

- "EXPRESSION Variable Syntax" on page 79
- "EXPRESSION Variable Example" on page 79

EXPRESSION Variable Syntax

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

where

`variableName` is the name of the JavaScript variable whose value is a expression (mandatory)

`type` is the type of JavaScript variable (optional, default value is EXPRESSION)

`source` is the URI of the pages (optional, default is *, meaning any source)

EXPRESSION Variable Example

Assume the base URL of the page is:

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

Page Content

```
<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>
```

Rules

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

Output

```
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";
```

Description

The function `psSRAPRewriter_convert_expression` is prefixed to the right side of the expression variable `expvar` in the first line. This function processes the expression and rewrites the content at runtime. In the third line the value is rewritten as a simple URL.

DHTML(Dynamic HTML) Variables

These are JavaScript variables that contain HTML content.

This section is divided into the following parts:

- “DHTML Syntax” on page 80
- “DHTML Example” on page 81

DHTML Syntax

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

where

`variableName` is the name of the JavaScript variable with DHTML content (mandatory)

`type` is the type of the variable (mandatory, the value must be DHTML)

`source` is the URL of the page (optional, the default is *, meaning in any page)

DHTML Example

Assume the base URL of the page is:

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

Page Content

```
<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>
```

Rules

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

Output

```
<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>
```

Description

The JavaScript parser reads the value of `dhtmlVar` as HTML content and sends the content through the HTML parser. The HTML parser applies the HTML rules where the href attribute rules are matched and hence the URL is rewritten.

DJS (Dynamic JavaScript) Variables

These are JavaScript variables that contain JavaScript content.

This section is divided into the following parts:

- "DJS Syntax" on page 82
- "DJS Example" on page 82

DJS Syntax

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

where

`variable` is the JavaScript varible whose value is javascript.

DJS Example

Assume the base URL of the page is:

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

Page Content

```
//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;"
```

Rules

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

Output

```
//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;"
```

Description

Two rules are required here. The first rule locates the dynamic JavaScript variable `dJSVar`. The value of this variable is again a JavaScript of type `URL`. The second rule is applied to rewrite the value of this JavaScript variable.

SYSTEM Variables

These are variables are not declared by the use and have limited support. They are available as a part of the JavaScript standard. For example, `window.location.pathname`.

This section is divided into the following parts:

- "SYSTEM Variable Syntax" on page 83
- "SYSTEM Variable Example" on page 83

SYSTEM Variable Syntax

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

where

`variableName` is the JavaScript system variable (mandatory and the values could be ones that match these patterns: document.URL, document.domain, location, doument.location, location.pathname, location.href, location.protocol, location.hostname, location.host and location.port. All these are present in the generic_ruleset. Do not modifiy these system var rules .

`type` specifies system type values (mandatory and value is DJS)

`source` is the URI of this pages (optional, default value is *, meaning in any page)

SYSTEM Variable Example

Assume the base URL of the page is:

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

Page Content

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

Rules

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

Output

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

Description

Rewriter locates the system variable which matches the rule, then the `psSRAPRewriter_convert_system` function is prefixed. This function processes the system variable at runtime and rewrites the resulting URL accordingly.

Function Arguments

Function parameters whose value needs to be rewritten are classified into 4 categories:

- "URL Parameters" on page 85
- "EXPRESSION Parameters" on page 86
- "DHTML Parameters" on page 88
- "DJS Parameters" on page 89

Generic Syntax

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

where

`name` is the name of the JavaScript function (mandatory)

`paramPatterns` specifies the parameters that need to be rewritten (mandatory)

`y` the position of `y` indicates the parameter that the needs to be rewritten. For example, in the syntax, the first parameter needs to be rewritten, but the second parameter should not be rewritten.

`type` specifies the kind of value this parameter needs (optional, default is EXPRESSION type)

`source` page source URI (optional, default is *, meaning in any page)

URL Parameters

Function takes this parameter as a string and this string could be treated as URL.

This section is divided into the following parts:

- "URL Parameter Syntax" on page 85
- "URL Parameter Example" on page 85

URL Parameter Syntax

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

where

`name` is the name of the function with a type parameter of URL (mandatory)

`paramPatterns` specifies the parameters that need to be rewritten (mandatory)

`y` the position of `y` indicates the parameter that needs to be rewritten. For example, in the syntax, the first parameter needs to be rewritten, but the second parameter should not be rewritten.

`type` is the type of the function (mandatory, and the value must be URL)

`source` is the URL of the page which has this function call (optional, default is *, meaning in any URL)

URL Parameter Example

Assume the base URL of the page is:

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

Page Content

```
<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>
```

Rules

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

Output

```
<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>
```

Description

The first rule specifies that the first two parameters in the function with name `test` need to be rewritten. Hence the first two parameters of the test function are rewritten. The second rule specifies that the first parameter of the `window.open` function needs to be written. The URL within the `window.open` function is prefixed with the Gateway URL and the base URL of the page that contains the function parameters.

EXPRESSION Parameters

These parameters take an expression value, which when evaluated, results in a URL.

This section is divided into the following parts:

- "EXPRESSION Parameter Syntax" on page 86
- "EXPRESSION Parameter Example" on page 87

EXPRESSION Parameter Syntax

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

where

`name` is the name of the function (mandatory).

`paramPatterns` specifies the parameters that need to be rewritten (mandatory)

`y` the position of `y` indicates the function parameter that needs to be rewritten. In the syntax above, only the first parameter is rewritten.

`type` specifies the value EXPRESSION (optional)

`source` URI of the page where this function is called.

EXPRESSION Parameter Example

Assume the base URL of the page is:

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

Page Content

```
<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>
```

Rules

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

Output

```
<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>
```

Description

The rule specifies that the first parameter of the `jstest1` function needs to be rewritten by considering this as an EXPRESSION function param. In the sample page content, the first parameter is an expression that will be evaluated only at runtime. Rewriter prefixes this expression with the `psSRAPRewriter_convert_expression` function. The expression is evaluated, and the `psSRAPRewriter_convert_expression` function rewrites the output at runtime.

Note – In the above example, the variable `test1 is not required` as a part of the JavaScript variable rule. The function rule for `jstest1` takes care of the rewriting.

DHTML Parameters

Function parameter whose value is HTML

Native JavaScript methods such as `document.write()` that generate an HTML page dynamically fall under this category.

This section is divided into the following parts:

- "DHTML Parameter Syntax" on page 88
- "DHTML Parameter Example" on page 88

DHTML Parameter Syntax

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

where

`name` is the name of the function.

`paramPatterns` specifies the parameters that need to be rewritten (mandatory)

`y` the position of `y` indicates the function parameter that needs to be rewritten. In the syntax above, only the first parameter is rewritten.

DHTML Parameter Example

Assume the base URL of the page is:

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

Page Content

```
<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>
```

Rules

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

Output

```
<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>
```

Description

The first rule specifies that the first parameter in the function `document.write` needs to be rewritten. The second rule specifies that the first parameter in the function `document.writeln` needs to be rewritten. The third rule is a simple HTML rule that specifies that all attributes with the name `href` need to be rewritten. In the example, the DHTML parameter rules identify the parameters in the functions that need to be rewritten. Then the HTML attribute rule is applied to actually rewrite the identified parameter.

DJS Parameters

Function parameters whose value is JavaScript.

This section is divided into the following sections:

- "DJS Parameter Syntax" on page 90
- "DJS Parameter Example" on page 90

DJS Parameter Syntax

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

where

name is the name of the function where one parameter is DJS (mandatory)

paramPatterns specifies which parameter in the above function is DJS (mandatory)

y the position of y indicates the function parameter that needs to be rewritten. In the syntax above, only the first parameter is rewritten.

type is DJS (mandatory)

source is the URI of the page (optional, default is *, meaning any URI)

DJS Parameter Example

Assume the base URL of the page is:

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

Page Content

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

Rules

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

Output

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

Description

The first rule specifies that the second parameter of the function NavBarMenuItem which contains JavaScript needs is to be rewritten. Within the JavaScript, the variable top.location also needs to be rewritten. This variable is rewritten using the second rule.

Rules for XML Content

Web pages may contain XML content which in turn can contain URLs. XML content that needs to be rewritten is classified into two categories:

- "Tag Text" on page 91 (same as PCDATA or CDATA of the tag)
- "Attribute" on page 92

Tag Text

This rule is for rewriting the PCDATA of CDATA of the tag element.

This section is divided into the following parts:

- "Tag Text Syntax" on page 91
- "Tag Text Example" on page 91

Tag Text Syntax

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

where

`tagName` is the name of the tag

`attributePatterns` is the attributes and their value patterns for this tag (optional, meaning this tag has no attributes at all)

`source` is the URI of this xml file (optional, default is *, meaning, any xml page)

Tag Text Example

Assume the base URL of the page is:

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

Page Content

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

Rules

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

Output

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

Description

The first line in the page content has an “Attribute Example” on page 92. The second line in the page content does not contain an attribute with the attribute called name and value of attribute name to be `src`, and hence no rewriting is done. To rewrite this also we need to have `<TagText tag="attribute"/>`

Attribute

The rules for XML attributes are similar to the attribute rules for HTML. The difference is that attribute rules of XML are cases sensitive while HTML attribute rules are not. This is again due to case sensitivity built into XML and not into HTML.

Rewriter translates the attribute value based on the attribute name.

This section is divided into the following parts:

- “Attribute Syntax” on page 92
- “Attribute Example” on page 92

Attribute Syntax

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

where

`attributeName` is the name of the attribute (mandatory)

`tag` is the name of the tag, where this attribute is present (optional, deafult is * , meaning any tag)

`valuePatterns` See “Using Pattern Matching in Rules” on page 75.

`source` is the URI of this XML page (optional, default is *, meaning in any XML page)

Attribute Example

Assume the base URL of the page is:

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

Page Content

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

Rules

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

Output

```
<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>
```

Description

In the above example, only the fourth line is rewritten because it meets all the conditions specified in the rule. See "Using Pattern Matching in Rules" on page 75.

Rules for Cascading Style Sheets

The Cascading Style Sheets (including CCS2) in HTML pages are translated. No rules are defined for this translation as the URL presents only in the `url()` functions and import syntaxes of the CSS.

Rules for WML

WML is similar to HTML and hence HTML rules are applied for WML content.Use the generic ruleset for WML content. See "Rules for HTML Content" on page 70.

Using the Recursive Feature

Rewriter uses the recursive feature to search to the end of the matched string pattern for the same pattern.

For example, when Rewriter parses the following string:

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

the rule

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

rewrites only the first occurrence of the pattern and it would look like this:

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

but if you use the recursive option as,

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

Rewriter searches to the end of the matched string pattern for the same pattern, hence the output would be:

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

Troubleshooting Using Debug Logs

To troubleshoot a Rewriter problem, you need to enable debug logs.

Debug Messages are classified as follows.

- Error– errors that Rewriter cannot recover from
- Warning– warnings that do not critically affect the functioning of Rewriter. Rewriter is able to recover this type of error, but some misbehavior may or may not result. Some messages shown in warnings are informational. For example “Not rewriting image content” is logged as a warning message. This is fine as Rewriter is not supposed to rewrite the images.
- Message– the highest level of information that Rewriter provides.

Setting the Rewriter Debug Level

▼ To Set the Rewriter Debug Level

1 **Log in as root to the Gateway machine and edit the following file:**

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

2 **Set the debug level:**

```
com.iplanet.services.debug.level=
```

The debug levels are:

`error` - Only serious errors are logged in the debug file. Rewriter usually stops functioning when such errors occur.

`warning` - Warning messages are logged.

`message` - All debug messages are logged.

`off` - No debug messages are logged.

3 **Specify the directory for the debug files in the following property of the** `AMConfig-`*`instance-name`*`.properties` **file:**

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

where `/var/opt/SUNWam/debug` is the default debug directory.

4 **Restart the Gateway from a terminal window:**

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

Debug File Names

When the debug level is set to message, debug generates a set of files. "Debug File Names" on page 95 lists the Rewriter files and the information contained within them.

TABLE 4–2 Rewriter Debug Files

File Name	Information
RuleSetInfo	Contains all the rulesets which have been used for rewriting, are logged in this file.
Original Pages	Contains the page URI, resolveURI (if different than the page URI), content MIME, the ruleset that has been applied to the page, parser MIME, and the original content. Specific error/warning/messages related to parsing also appear in this file. In message mode full content is logged. In warning and error mode only exceptions that occurred during rewriting are logged.
Rewritten Pages	Contains the page URI, resolveURI (if different than the page URI), content MIME, ruleset that has been applied to the page, parser MIME, and the rewritten content. This is filled when the debug mode is set to message.
Unaffected Pages	Contains a list the pages that were not modified.
URIInfo Pages	Contains the URLs found and translated. Details of all the pages whose content remain same as original data are logged in this file. Details logged are: Page URI, MIME and Encoding data, rulesetID used for rewriting, and Parser MIME.

In addition to the above files, Rewriter generates a file for debug messages that are not captured in the above files. This file name consists of two parts: the first part is either `pwRewriter` or `psSRARewriter` and the second part is an extension using either `portal` or the *gateway-profile-name*.

The debug files are displayed on the portal or the Gateway. These files are in the directory indicated in the `AMConfig-`*instance-name*`.properties` file.

The Rewriter component generates the following set of files to help in debugging,

*prefix*_RuleSetInfo.*extension*

*prefix*_OrginalPages.*extension*

*prefix*_RewrittenPages.*extension*

*prefix*_UnaffectedPages.*extension*

*prefix*_URIInfo.*extension*

where

prefix is either `psRewriter` for URLScraper usage logs or `psSRAPRewriter` for Gateway usage logs.

extension is either `portal` for URLScraper usage or `gateway-profile-name` for Gateway usage.

For example, if the Rewriter on the Gateway is used to convert pages and the default gateway profile is used, debug creates these files:

`psSRAPRewriter_RuleSetInfo.default`

`psSRAPRewriter_OriginalPages.default`

`psSRAPRewriter_RewrittenPages.default`

`psSRAPRewriter_UnaffectedPages.default`

`psSRAPRewriter_URIInfo.default`

`psSRAPRewriter.default`

Working Samples

This section includes:

- Simple HTML pages with content that needs to be rewritten
- Rules required to rewrite the content
- Corresponding rewritten HTML page

These sample pages are available in the *portal-server-URL*/`rewriter` directory. You can browse through the page before the rule is applied, and then view the file with the rewritten output through your Gateway to see how the rule works. In some samples, the rule is already a part of the `default_gateway_ruleset`. In some samples, you may have to include the rule in the `default_gateway_ruleset`. This is mentioned at the appropriate places.

Note – Some of the statements appear in bold to indicate that they have been rewritten.

The following samples are available:

HTML

- "Sample for HTML Attributes" on page 98
- "Sample for HTML Forms" on page 102
- "Sample for HTML Applets" on page 104

JavaScript

- Variables
  - "Sample for JavaScript URL Variables" on page 105
  - "Samples for JavaScript Content" on page 105
  - "Sample for JavaScript DHTML Variables" on page 110
  - "Sample for JavaScript DJS Variables" on page 112
  - "Sample for JavaScript SYSTEM Variables" on page 114

Functions

- "Sample for JavaScript URL Functions" on page 115
- "Sample for JavaScript EXPRESSION Functions" on page 117
- "Sample for JavaScript DHTML Functions" on page 118
- "Sample for JavaScript DJS Functions" on page 120

XML

- Sample for XML Attributes

Samples for HTML Content

Sample for HTML Attributes

▼ To Use the HTML Attributes Sample

1 **This sample can be accessed from:**

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

2 **Ensure that `abc.sesta.com` and `host1.siroe.com` are defined in the Proxies for Domains and Subdomains list in the Gateway service.**

If this is not defined, a direct connection is assumed, and the Gateway URL is not prefixed.

You need not add the rule specified in this sample to the `default_gateway_ruleset` because the rule is already defined.

HTML Before Rewriting

```
<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>
```

Rule

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

HTML After Rewriting

```
<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>
```

// This URL is rewritten because the `<Attrib name="href"/>` rule is already defined in the `default_gateway_ruleset`. Because the URL is already absolute, only the Gateway URL is prefixed. Ensure that `abc.sesta.com` is defined in the Proxies for Domains and Subdomains list in the Gateway service. Otherwise, the Gateway URL is not prefixed, because a direct connection is assumed.

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

// Again, `host1.siroe.com` needs to be defined in the Proxies for Domains and Subdomains list in the Gateway service. Otherwise, the Gateway URL is not prefixed, because a direct connection is assumed.

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

// Because a relative path is specified, the Gateway URL and the portal-server-URL are prefixed along with the required subdirectories. This link will not work because a directory called `images` under the `HTML` directory is not specified in the sample structure provided.

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

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

// Because a relative path is specified, the Gateway URL and the Portal Server URL are prefixed along with the required subdirectories.

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

// Because a relative path is specified, the Gateway URL and the Portal Server URL are prefixed along with the required subdirectories. This link will not work because a directory called `images` under the `Rewriter` directory is not specified in the sample structure provided

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

Sample for HTML Dynamic JavaScript Tokens

This section discuses using the HTML JavaScript token sample

To Use the HTML JavaScript Token Sample:

1. **This sample can be accessed from:**

   portal-server-URL/rewriter/HTML/jstokens/JStokens.html

2. **Add the rule specified in this sample to the `default_gateway_ruleset` in the section "Rules for Rewriting JavaScript Source".**

3. **Edit the `default_gateway_ruleset` in the Rewriter service under the Portal Server Configuration in the Portal Server administration console.**

4. **Restart the Gateway from a terminal window:**

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

HTML Before Rewriting

```
<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>
```

Rule

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

Note – `<Function name="URL" name="Check" paramPatterns="y"/>` is a JavaScript function rule and is explained in detail in the JavaScript function sample.

HTML After Rewriting

```
<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;">
```

// All the statements are rewritten in this sample. The Gateway and Portal Server URLs are prefixed in each case. This is because rules for `onAbort`, `onBlur`, `onFocus`, `onChange`, and `onClick` are defined in the `default_gateway_ruleset` file. Rewriter detects the JavaScript tokens and passes it to the JavaScript function rules for further processing. The second rule listed in the sample tells Rewriter which parameter to rewrite.

```
</body>
<br>

Rewriting ends

</html>
```

Sample for HTML Forms

To Use the Form Sample

1. **Access the sample from:**

   portal-server-URL/rewriter/HTML/forms/formrule.html

2. **Ensure that `abc.sesta.com` is defined in the Proxies for Domains and Subdomains list in the Gateway service.**

   If this is not defined, a direct connection is assumed, and the Gateway URL is not prefixed.

3. **Add the rule specified in this sample to the `default_gateway_ruleset` in the section "Rules for Rewriting HTML Attributes".**

4. **Edit the `default_gateway_ruleset` in the Rewriter service under the Portal Server Configuration in the Portal Server administration console.**

5. **Restart the Gateway from a terminal window:**

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

HTML Page Before Rewriting

```
<!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>
```

Rule

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

HTML Page After Rewriting

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

// This URL is rewritten because `<Attribute` `name="action"/>` is defined as part of the HTML rules in the `default_gateway_ruleset`. Because the URL is already absolute, only the Gateway URL needs to be prefixed. Ensure that `abc.sesta.com` is defined in the Proxies for Domains and Subdomains list in the Gateway service. Else, the Gateway URL is not prefixed because a direct connection is assumed.

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

// Here the form name is `form1`, and the field name is `name1`. This matches the form name and field name specified in the rule. The rule states the `valuePatterns` as `0|1234|` which matches the `value` in this statement. Hence the URL occurring after the `valuePattern` is rewritten. The Portal Server URL and the Gateway URL are prefixed. See "“Using Pattern Matching in Rules” on page 75 for details on `valuePatterns`.

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

// This URL is not rewritten because the `name` does not match the `field` name specified in the rule.

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

// This URL is rewritten because `<Attribute` `name="action"/>` is defined as part of the HTML rules in the default ruleset. Because the URL is already absolute, only the Gateway URL needs to be prefixed.

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

// This URL is not rewritten because the form name does not match the name specified in the rule.

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

Sample for HTML Applets

▼ To Use the Sample for Applets

1. **Obtain the applet class file. The `RewriteURLinApplet.class` file is present in the following location:**

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

   The base URL of the page where the applet code is present is:

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

2. **Add the rule specified in this sample to the `default_gateway_ruleset` in the section "Rules for Rewriting HTML Attributes".**

3. **Edit the `default_gateway_ruleset` in the Rewriter service under the Portal Server Configuration in the Portal Server administration console.**

4. **Restart the Gateway:**

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

HTML Before Rewriting

```
<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>
```

Rule

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

HTML After Rewriting

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

// This URL is rewritten because the rule `<Attribute name="codebase"/>` is already present as part of the `default_gateway_ruleset` file. the Gateway and the Portal Server URLs are prefixed along with the path up to the `appletcode` directory.

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

// This URL is rewritten because the base URL of the page is `rule1.html`, and the param name matches the param `Test*` specified in the rule. Because `index.html` is specified to be at the root level, the Gateway and Portal Server URLs are prefixed directly.

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

// This URL is rewritten because the base URL of the page is `rule1.html`, and the param name matches the param `Test*` specified in the rule. The path is prefixed as required.

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

// This URL is rewritten because the base URL of the page is `rule1.html`, and the param name matches the param `Test*` specified in the rule. The path is prefixed as required.

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

Samples for JavaScript Content

Sample for JavaScript URL Variables

▼ To Use the JavaScript URL Variables Sample

1 **This sample can be accessed from:**

portal-server-URL/rewriter/JavaScript/variables/url/js_urls.html

2. **Ensure that `abc.sesta.com` is defined in the Proxies for Domains and Subdomains list in the Gateway service.**

   If this is not defined, a direct connection is assumed, and the Gateway URL is not prefixed.

3. **Add the rule specified in this sample to the `default_gateway_ruleset` in the section "Rules for Rewriting JavaScript Source".**

4. **Edit the `default_gateway_ruleset` in the Rewriter service under Portal Server Configuration in the Portal Server administration console.**

5. **If you added the rule, restart the Gateway:**

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

HTML Page Before Rewriting

```
<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>
```

Rule

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

HTML Page After Rewriting

```
<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";
```

// All the above URLs are JavaScript variables of type URL and name `imgsrc` as specified in the rule. Hence they are prefixed with the Gateway and the Portal Server URLs. The path following the Portal Server URL is prefixed as required.

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

// This line is rewritten because the rule `<Attribute` name="`src`"/> is defined in the `default_gateway_ruleset`

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

Sample for JavaScript EXPRESSION Variables

▼ To Use the JavaScript Expression Variables Sample

1. **This sample can be accessed from:**

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

2. **Add the rule specified in this sample (if it does not already exist) to the** `default_gateway_ruleset` **in the section "Rules for Rewriting JavaScript Source".**

3. **Edit the** `default_gateway_ruleset` **in the Rewriter service under Portal Server Configuration in the Portal Server administration console.**

4. **If you added the rule, restart the Gateway:**

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

HTML Page Before Rewriting

```
<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>
```

Rule

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

HTML Page After Rewriting

```
<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);
```

// Rewriter recognizes the right hand side of this statement to be a JavaScript EXPRESSION variable. Rewriter is not able to resolve the value of this expression at the server end. Hence, the `psSRAPRewriter_convert_expression` function is prefixed to the expression. The expression is evaluated at the client end, and rewritten as required.

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

// The rewritten value of `expvar` from the previous statement is used to arrive at the value of this expression. Because the result is a valid URL (a graphic exists at this location in the sample), the link will work.

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

// Rewriter recognizes the right hand side of `expvar` to be a string expression. This can be resolved at the server side, and hence is rewritten directly.

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

// The rewritten value of `expvar` from the previous statement is used to arrive at the value of this expression. Because the result is a not a valid URL (a graphic does not exist at the resultant location), the link will not work.

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

Sample for JavaScript DHTML Variables

▼ To Use the JavaScript DHTML Variables Sample

1. **This sample can be accessed from:**

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

2. **Ensure that `abc.sesta.com` is defined in the Proxies for Domains and Subdomains list in the Gateway service. If this is not defined, a direct connection is assumed, and the Gateway URL is not prefixed.**

3. **Add the rule specified in this sample (if it does not already exist) to the `default_gateway_ruleset` in the section "Rules for Rewriting JavaScript Source". Edit the `default_gateway_ruleset` in the Rewriter service under Portal Server Configuration in the Portal Server administration console.**

4. **If you added the rule, restart the Gateway:**

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

HTML Page Before Rewriting

```
<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>
```

Rule

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

HTML Page After Rewriting

```
<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>"
```

// The JavaScript DHTML rule identifies the right hand side of the `dhtmlVar` as dynamic HTML content. Hence, the HTML rules in the `default_gateway_ruleset` file are applied. The dynamic HTML contains a `href` attribute. The `default_gateway_ruleset` defines the rule `<Attribute name="href"/>`. Hence the value of the `href` attribute is rewritten. But the URL is not absolute; therefore, the relative URL is replaced with the base URL of the page, and the required subdirectories. This in turn is prefixed with the Gateway URL to derive the final rewritten output.

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

// Although the base URL of the page is appended, and the Gateway URL is prefixed, the resultant URL will not work. This is because the initial URL `/../images/test.html` is inaccurate.

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

// Here again, the JavaScript DHTML rule identifies the right hand side to be dynamic HTML content, and passes it to the HTML rules. The HTML rule `<Attribute name="href"/>` from the `default_gateway_ruleset` is applied, and the statement is rewritten as shown. The Gateway URL and Portal Server URL are prefixed.

```
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>"
```

// The JavaScript DHTML rule identifies the dynamic HTML content on the right hand side, and passes the statement to the HTML rules. The `<Attribute name="src"/>` rule in the default_gateway_ruleset is applied. Because the URL is absolute, only the Gateway URL needs to be prefixed. Ensure that abc.sesta.com is defined in the Proxies for Domains and Subdomains list for this URL to be rewritten.

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

// This line is rewritten because the rule <Attribute name="src"/> is defined in the default_gateway_ruleset.

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

Sample for JavaScript DJS Variables

▼ To Use the JavaScript DJS Variables Sample

1. **This sample can be accessed from:**

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

2. **Ensure that abc.sesta.com is defined in the Proxies for Domains and Subdomains list in the Gateway service. If this is not defined, a direct connection is assumed, and the Gateway URL is not prefixed.**

3. **Add the two rules specified in this sample (if it does not already exist) to the default_gateway_ruleset in the section "Rules for Rewriting JavaScript Source". Edit the default_gateway_ruleset in the Rewriter service under Portal Server Configuration in the Portal Server administration console.**

4. **Restart the Gateway:**

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

HTML Page Before Rewriting

```
<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>
```

Rule

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

HTML Page After Rewriting

```
<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;"
```

// All the above statements are rewritten with the Gateway and Portal Server URLs. The required path is prefixed as appropriate. The first rule identifies the right hand side of `dJSVar` as a dynamic JavaScript variable. This is then passed to the second rule which identifies the right hand side of `dJSimgsrc` as a JavaScript variable of type URL. This is rewritten accordingly.

```
//-->
</SCRIPT>
<br>
Testing Dynamic JavaScript Variables
```

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

// This line is rewritten because the rule `<Attribute name="src"/>` is defined in the `default_gateway_ruleset`.

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

Sample for JavaScript SYSTEM Variables

▼ To Use the JavaScript System Variables Sample

1 **This sample can be accessed from:**

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

2 **Add the rule specified in this sample (if it does not already exist) to the** `default_gateway_ruleset` **in the section "Rules for Rewriting JavaScript Source".**

3 **Edit the** `default_gateway_ruleset` **in the Rewriter service under Portal Server Configuration in the Portal Server administration console.**

4 **Restart the Gateway:**

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

HTML Page Before Rewriting

```
<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>
```

Rule

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

HTML After Rewriting

```
<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”));
```

// Rewriter identifies `window.location.pathname` as a JavaScript SYSTEM variable. The value of this variable cannot be determined at the server end. So the Rewriter prefixes the variable with the `psSRAPRewriter_convert_pathname` function. This wrapper function determines the value of the variable at the client end and rewrites as required.

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

Sample for JavaScript URL Functions

▼ To Use the JavaScript URL Functions Sample

1. **This sample can be accessed from:**

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

2 **Add the rule specified in this sample (if it does not already exist) to the** `default_gateway_ruleset` **in the section "Rules for Rewriting JavaScript Source". Edit the** `default_gateway_ruleset` **in the Rewriter service under the Portal Server Configuration in the Portal Server administration console.**

3 **Restart the Gateway:**

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

HTML Page Before Rewriting

```
<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>
```

Rule

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

HTML Page After Rewriting

```
<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>
```

Sample for JavaScript EXPRESSION Functions

▼ To Use the JavaScript Expressions Function Sample

1. **This sample can be accessed from:**

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

2. **Add the rule specified in this sample (if it does not already exist) to the** `default_gateway_ruleset` **in the section** `Rules for Rewriting JavaScript Source`**.**

3. **Edit the** `default_gateway_ruleset` **in the Rewriter service using the Portal Server administration console.**

4. **Restart the Gateway:**

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

HTML Page Before Rewriting

```
<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>
```

Rule

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

HTML Page After Rewriting

```
<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()));
```

// The rule states that the first parameter in the function `jstest1` which is of type EXPRESSION needs to be rewritten. The value of this expression is `/test/images/test.html`. This is prefixed with the Portal Server and the Gateway URLs.

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

Sample for JavaScript DHTML Functions

▼ To Use the JavaScript DHTML Functions Sample

1. **This sample can be accessed from:**

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

2 **Add the rule specified in this sample (if it does not already exist) to the** `default_gateway_ruleset` **in the section "Rules for Rewriting JavaScript Source".**

3 **Edit the** `default_gateway_ruleset` **in the Rewriter service under Portal Server Configuration in the Portal Server administration console.**

4 **Restart the Gateway:**

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

HTML Page Before Rewriting

```
<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>
```

Rule

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

HTML Page After Rewriting

```
<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)
```

// The first rule specifies that the first parameter of the DHTML JavaScript function `document.write` needs to be rewritten. Rewriter identifies the first parameter to be a simple HTML statement. The HTML rules section in the `default_gateway_ruleset` has the rule `<Attribute name="href" />` which indicates that the statement needs to be rewritten.

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

// The second rule specifies that the first parameter of the DHTML JavaScript function `document.writeln` needs to be rewritten. Rewriter identifies the first parameter to be a simple HTML statement. The HTML rules section in the `default_gateway_ruleset` has the rule `<Attribute name="href" />` which indicates that the statement needs to be rewritten.

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

// The above statements are not rewritten although the DHTML rule identifies the functions `document.write` and `document.writeln`. This is because the first parameter in this case is not simple HTML. It could be any string, and Rewriter does not know how to rewrite this.

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

Sample for JavaScript DJS Functions

▼ To Use the JavaScript DJS Functions Sample

1. **This sample can be accessed from:**

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

2. **Ensure that `abc.sesta.com` is defined in the Proxies for Domains and Subdomains list in the Gateway service.**

   If this is not defined, a direct connection is assumed, and the Gateway URL is not prefixed.

3. **Add the rule specified in this sample (if it does not already exist) to the `default_gateway_ruleset` in the section "Rules for Rewriting JavaScript Source". Edit the `default_gateway_ruleset` in the Rewriter service under Portal Server Configuration in the Portal Server administration console.**

4 Restart the Gateway:

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

HTML Page Before Rewriting

```
<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>
```

Rule

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

HTML Page After Rewriting

```
<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` is an entry in the Proxies for Domains and Subdomains list in the Gateway service. Hence Rewriter needs to rewrite this URL. But because an absolute URL, the Portal Server URL need not be prefixed. The DJS rule states that the second parameter of the DJS function `NavBarMenuItem` needs to be rewritten. But the second parameter is again a JavaScript variable. A second rule is required to rewrite the value of this variable. The second rule specifies that the value of the JavaScript variable `top.location` needs to be rewritten. Because all these conditions are met, the URL is rewritten.

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

// Although the DJS rule specifies that the second parameter of the function `NavBarMenuItem` needs to be rewritten, it does not happen in this statement. This is because Rewriter does not recognize the second parameter as simple HTML.

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

Sample for XML Attributes

▼ To Use the XML Attributes Sample

1. **This sample can be accessed from:**

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

2. **Add the rule specified in this sample (if it does not already exist) to the** `default_gateway_ruleset` **in the section "Rules for Rewriting XML Source".**

3. **Edit the** `default_gateway_ruleset` **in the Rewriter service under the Portal Server Configuration in the Portal Server administration console.**

4. **Restart the Gateway:**

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

XML Before Rewriting

```
<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>
```

Rule

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

HTML After Rewriting

```
<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>
```

// This statement is rewritten because it matches the conditions specified in the rule. The `Attribute` name is `href`, `tag` is `check` and the `valuePatterns` is `1234.` The string following `valuePatterns` is rewritten. See "Using Pattern Matching in Rules" on page 75 for details on `valuePatterns`.

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

Case Study

This section includes the source HTML pages for a sample mail client. This case study does not cover all possible scenarios and rules. This is just a sample ruleset to help you put together the rules for your intranet pages.

Assumptions

The following assumptions are made for this case study:

- The base URL of the mail client is assumed to be `abc.siroe.com`
- The Gateway URL is assumed to be `gateway.sesta.com`
- Relevant entries exist in the Proxies for Domains and Subdomains list in the Gateway service

Sample page 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>
```

Description

"Description" on page 125 shows the mapping between the sample ruleset and the case study.

TABLE 4–3 Mapping Between Sample Ruleset and Case Study

Page Content	Rule Applied	Rewriter Output	Description

`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` is a variable whose value is a simple URL.
			This rule tells Rewriter to search for a variable `g_szVirtualRoot` of type URL. If such a variable exists in the web page, Rewriter converts this to an absolute URL, and prefixes the Gateway 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` is the name of an attribute, and does not have any tag or valuePattern attached to it.
			This rule tells Rewriter to search for all attributes with the name `src`, and rewrite the value of that attribute.

`href="http://abc.siroe.com`
`/mailclient/destin/Inbox/`
`?Cmd=contents&amp;Page=1"`
	`<Attribute`
	`name="href"/>`
		`href="http://gateway.sesta.com/`
		`http://abc.siroe.com`
		`/mailclient/destin/`
		`Inbox/?Cmd=contents&amp;Page=1"`
			`href` is the name of an attribute, and does not have any tag or valuePattern attached to it.
			This rule tells Rewriter to search for all attributes with the name `href`, and rewrite the value of that attribute.

Note – The order of priority for applying the ruleset is `hostname-subdomain-domain`.

For example, assume that you have the following entries in the Domain-based rulesets list:

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

`ruleset3` is applied for all pages on `host1`.

`ruleset2` is applied for all pages in the `eng` subdomain, except for pages retrieved from `host1`.

`ruleset1` is applied for all pages in the `sesta.com` domain, except for pages retrieved from the `eng` subdomain, and from `host1`.

1. Click Save to complete.
2. Restart the Gateway from a terminal window:

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

Ruleset for Outlook Web Access

Secure Remote Access server supports MS Exchange 2000 SP3 installation and MS Exchange 2003 of Outlook Web Access (OWA) on the Sun Java System Web Server and the IBM application server.

▼ To Configure the OWA Ruleset

1 **Log into the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab, and select the Gateway profile for which you want to set the attribute.**

3 **In the Map URIs to RuleSets field, enter the server name where Exchange 2000 is installed followed by the Exchange 2000 Service Pack 4 OWA ruleset.**

For example:

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

Using Public Folders

On the Exchange side Public Folders are configured to use NTLM Authorization. It needs to be changed to use HTTP Basic Authorization.

To do this, go to the Exchange server and select the Control Panel-->Administrative Tools, then open Internet Information Services.

Under Default Web Site there is a tab for Public Folders called Public. Right Click and select properties. Click on Directory Security Tab. Select "Edit.." on the Anonymous Access and Authentication control panel. Unselect everything else and select only Basic Authentication.

Mapping of 6.x RuleSet with 3.0

The following table lists the mapping of the Secure Remote Access server Rewriter rules with the previous releases of the Portal Server product.

TABLE 4–4 Mapping of Rules with SP3

Rewriter 6.0 DTD Element	Rewriter 3.0 List Box Name
Rules for HTML Content	
Attribute - URL	Rewrite HTML Attributes
Attribute - DJS	Rewrite HTML Attributes containing JavaScript
Form	Rewrite Form Input Tag List
Applet	Rewrite Applet/Object Parameter Values List
Rules for JavaScript Content	
Variable - URL	Rewrite JavaScript Variables in URL
Variable - EXPRESSION	Rewrite JavaScript Variables Function
Variable - DHTML	Rewrite JavaScript Variables in HTML
Variable - DJS	Rewrite JavaScript Variables in JavaScript
Variable - SYSTEM	Rewrite JavaScript System Variables
Function - URL	Rewrite JavaScript Function Parameters
Function - EXPRESSION	Rewrite JavaScript Function Parameters Function
Function - DHTML	Rewrite JavaScript Function Parameters in HTML
Function - DJS	Rewrite JavaScript Function Parameters In JavaScript
Rules for XML Content	
Attribute - URL	Rewrite Attribute value of XML Document
TagText	Rewrite Text data of XMl Document
Rules for CSS Content	
Rules are not required. By default, all URLs are translated	

TABLE 4–4 Mapping of Rules with SP3 *(Continued)*

Rewriter 6.0 DTD Element	Rewriter 3.0 List Box Name
Rules for WML Content	
No rules defined. WML is treated at HTML and HTML rules are applied.	
Rules for WMLScript Content	
No support for WML Script	

CHAPTER 5

Working with NetFile

This chapter describes NetFile and its operation. To configure NetFile, see Chapter 14, "Configuring NetFile."

- "Introduction to NetFile" on page 129
- "Supported File Access Protocols" on page 130

Introduction to NetFile

NetFile is a file manager application that enables the user to access and operate on remote file systems and directories.

The NetFile component of Secure Remote Access is available as Java2 applets. The Java2 applet has a better interface and increased ease of accessibility.

NetFile provides the following key features:

- Facility to add or remove shares or folders
- File upload and download
- Search for files and folders
- File compression using GZIP and ZIP
- Mail facility within the NetFile environment
- Save the current NetFile session information
- Drag and Drop of files

Supported File Access Protocols

NetFile allows you to access remote systems using FTP, NFS, and jCIFS (Microsoft Windows) protocols. It includes the following file access protocol features:

- If the user specifies AUTODETECT to add a system, NetFile uses the following sequence to automatically detect which protocol to use:
  - Checks the host for FTP server on port 21. If the FTP response contains the string "NetWare", this is considered a NETWARE host.
  - Checks the host for NFS server on port 2049.
  - Checks the host for Microsoft Windows on port 139.
  - If all of the above fail, a message saying unable to determine the host type is displayed.

    The first file system type that is detected is used to connect to the requested host. The host detection order can be changed in the Portal Server administration console (PSConsole).

Note – The connection fails if the servers are running on non-standard ports.

- NetFile enables users to select the file server and protocol of their choice.

  For each of these protocols, the platforms that are supported are listed below.

TABLE 5–1 File Systems and Supported Protocols

File System/Protocol	Platform
FTP	Novell FTP 5.1 Server on Novell Netware MS FTP Server 4.0 on Win NT 4.0 MS FTP Server 5.0 on Win NT 2000 Solaris FTP Server WU_FTP 2.6.1 ProFTPD 1.2.8 vsFTPd 1.2.0
NFS	Solaris 2.6 and higher
jCIFS	Windows 95/98/NT/2000/ME/XP

Note – To upload files to a ProFTPD server using NetFile, "AllowStoreRestart" needs to be set to "on" in the `proftpd.conf` file on the host running ProFTPD server.

Support for Novell Netware is only through FTP server and not through native access.

To access Microsoft Windows (SMB/CIFS) file systems jCIFS must be installed on the Portal Server. jCIFS is an Open Source client library that implements the CIFS/SMB networking protocol.

▼ To Create a NetFile Policy

1. **Login to Portal Administration Console as administrator.**
2. **Select Secure Remote Access tab, and select NetFile tab.**
3. **Select the Organization/Role/User from Select DN drop-down box.**
4. **Set the privileges to access/deny hosts and services.**
5. **Click Save.**
6. **Restart the gateway.**

CHAPTER 6

Working with Netlet

This chapter describes how to use Netlet to run applications securely between users' remote desktops and the servers running applications on your intranet. To configure Netlet, see Chapter 11, "Configuring the Netlet."

This chapter contains the following sections:

Introduction to Netlet

Sun Java System Portal Server software users may want to run popular or company-specific applications on their remote desktops in a secure manner. You can provide secure access to these applications by setting up Netlet on your platform.

Netlet enables users to securely run common TCP/IP services over insecure networks such as the Internet. You can run TCP/IP applications (such as Telnet and SMTP), HTTP applications, and any fixed port applications.

If an application is TCP/IP-based or it uses fixed ports, you can run the application over Netlet.

Note – Dynamic ports are supported only when FTP is used. To use Microsoft Exchange, use OWA (Outlook Web Access).

Ensure that you notify the users to disable the pop-up blockers options in their browser, when using Netlet.

Netlet Components

The various components used by Netlet are shown in "Netlet Components" on page 134.

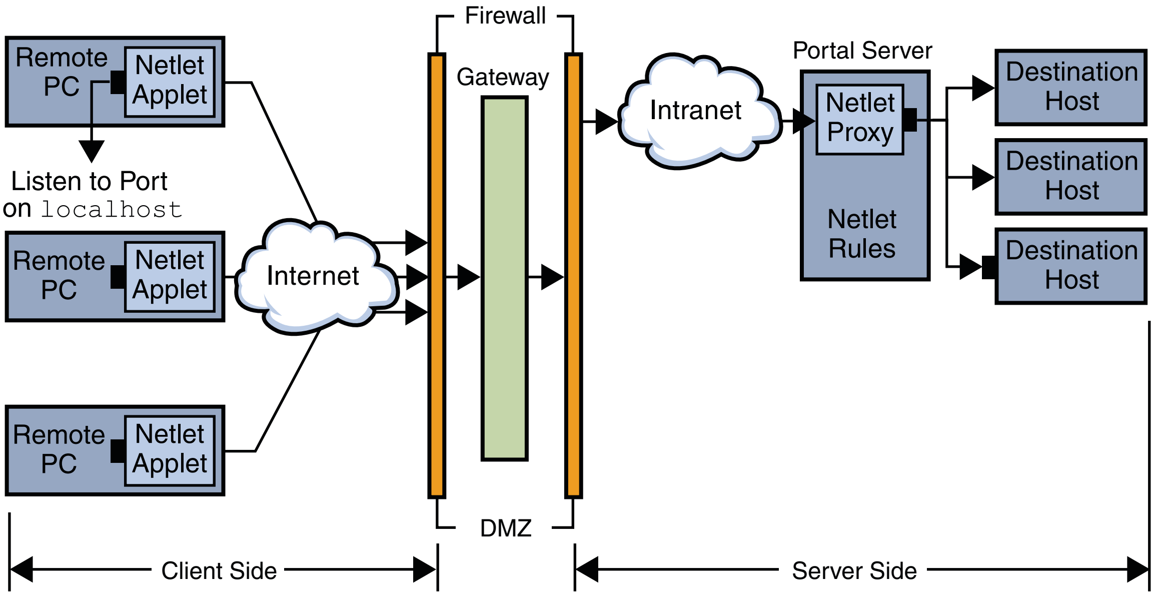


FIGURE 6–1 Netlet Components

Listen Port on localhost

This is the port on the client machine on which the Netlet applet listens. The client machine is the localhost.

Netlet Applet

The Netlet applet is responsible for setting up an encrypted TCP/IP tunnel between the remote client machine and intranet applications such as Telnet, Graphon or Citrix. The applet encrypts the packets and sends them to the Gateway, and decrypts the response packets from the Gateway and sends them to the local application.

For static rules the Netlet applet is downloaded automatically when the user logs into the portal. For dynamic rules, the applet is downloaded when the user clicks on the link corresponding to the dynamic rule. See "Types of Rules" on page 140 for details on static and dynamic rules.

To run Netlet in a Sun Ray Environment, see "Running Netlet in a Sun Ray Environment" on page 153.

Netlet Rules

A Netlet rule maps an application that needs to run on a client machine to the corresponding destination host. This means that Netlet operates only on packets sent to ports defined in the Netlet rule. This ensures greater security.

As an administrator, you need to configure certain rules for the functioning of Netlet. These rules specify various details such as the cipher to be used, URL to invoke, the applets to be downloaded, the destination port and the destination host. When a user on a client machine makes a request through Netlet, these rules help determine how the connection must be established. See "Defining Netlet Rules" on page 137 for details.

Netlet Provider

This is the UI component of Netlet. The provider allows users to configure the required applications from the Portal Server desktop. A link is created in the provider, and the user clicks on this to run the required application. Users can also specify the destination host for a dynamic rule in the desktop Netlet provider. See "Defining Netlet Rules" on page 137.

Netlet Proxy (Optional)

The Gateway ensures a secure tunnel between the remote client machine and the Gateway. The Netlet proxy is optional and you may choose not to install this proxy during the installation. For information on the Netlet proxy, see "Using a Netlet Proxy" on page 48.

Netlet Usage Scenario

The following sequence of events are involved in using Netlet:

1. The remote user logs into the Portal Server desktop.
2. If a static Netlet rule has been defined for a user, role or organization, the Netlet applet is automatically downloaded to the remote client.

   If a dynamic rule has been defined for a user, role, or organization, the user needs to configure the required application in the Netlet provider. The Netlet applet is downloaded when the user clicks on the application link in the Netlet provider. See "Defining Netlet Rules" on page 137 for details on static and dynamic rules.
3. Netlet listens on the local ports defined in the Netlet rules.

4. Netlet sets up a channel between the remote client and host over the ports specified in the Netlet rule.

Working With Netlet

For Netlet to work as required for various users across different organizations, you need to do the following:

1. Determine whether you need to create static or dynamic rules based on the user requirements. See "Types of Rules" on page 140.
2. Configure the options for the Netlet service from the Portal Server administration console. For information on configuring Netlet, see Chapter 11, "Configuring the Netlet."
3. Determine whether the rules should be organization, role, or user based and make modifications as required at each level. See the Portal Server Administration Guide for details on organization, role and user.

Note – Do not localize the value for the frameset parameter in the `srapNetletServlet.properties` file.

Downloading an Applet From a Remote Host

Sometimes a page is returned by a URL that contains an embedded applet that needs to be fetched from a remote machine. However Java security does not allow an applet to communicate with a host that it is not downloaded from. To allow the applet to communicate with the Gateway through the local network port, you need to check the Download Applet field on the Access Manager administration console and specify the following syntax:

local-port:server-host:server-port

where

local-port is the local port where Netlet listens for traffic originating from the applet

server-host is where the applet is to be downloaded from

server-port is the port used to download the applet

Defining Netlet Rules

Netlet configuration is defined by Netlet rules that are configured using the Portal Server administration console under the Secure Remote Access configuration tab. Netlet rules can be configured for organizations, roles, or users. If the Netlet rule is for a role or user, select the desired role or user after selecting the organization.

⚠ **Caution –** Netlet rules do not support multibyte entries. Do not specify multibyte characters for any of the fields in Netlet rules.

Netlet rules cannot contain any port number higher than 64000.

"Defining Netlet Rules" on page 137 lists the fields in a Netlet rule.

TABLE 6–1 Fields in a Netlet Rule

Parameter	Description	Value
Rule Name	Designates a name for this Netlet rule. You need to specify a unique name for each rule. This is useful while defining user access to specific rules.	
Encryption Ciphers	Defines the encryption cipher, or specifies the list of ciphers that the user can choose from.	The ciphers that you select appear in the Netlet provider as a list. The user can choose the required ciphers from the selected list. Default - The Default VM Native Cipher and the Default Java Plugin Cipher specified in the Netlet administration console are used.
Remote Application URL	Specifies the URL that the browser opens when the user clicks the associated link in the Netlet provider. The browser opens the window for the application and connects to `localhost` at the local port number specified later in the rule. You need to specify a relative URL.	URL to the application invoked by the Netlet rule. For example, `telnet://localhost:30000`. Specify a URL if the application uses an applet to invoke the application. `null`– Value that you set if the application is not started by a URL or controlled by the desktop. This is normally true for non-web-based applications.

TABLE 6–1 Fields in a Netlet Rule *(Continued)*

Parameter	Description	Value
Enable Download Applet	Indicates whether it is necessary to download an applet for this rule.	■ *Client Port* indicates the destination port on the client. This port must be different from the default loopback port. Specify a unique `local port` for each rule. ■ *Server Host* is the name of the server from which to download the applet. ■ *Server Port* represents the port on the server used to download the applet. If an applet is to be downloaded, and if the server is not specified, the applet is downloaded from the Portal Server host.
Enable Extend Session	This controls the idle time-out of a Portal Server session when Netlet is active.	Select this checkbox to keep the portal session alive when only Netlet is active and the rest of the portal application is idle. By default, this option is not selected.

TABLE 6–1 Fields in a Netlet Rule *(Continued)*

Parameter	Description	Value
Map Local Port to Destination Server Port	Local Port	Port on the client where Netlet listens. The value of *local-port* must be unique. You cannot specify a particular port number in more than one rule. Specify multiple local ports if you are specifying multiple hosts for multiple connections. See "Static Rule With Multiple Host Connections" on page 145 for the syntax. For an FTP rule the local port value must be 30021.
	Destination Host	Port on the client where Netlet listens. Recipient of the Netlet connection. *host* - Name of the host to receive the Netlet connection. This is used in a static rule. Use either the simple host name such as `siroe`, or a fully-qualified DNS-style host name such as `siroe.mycompany.com`. Specify multiple hosts for the following reasons: The value of *local-port* must be unique. You cannot specify a particular port number in more than one rule. Specify multiple local ports if you are specifying multiple hosts for multiple connections. See "Static Rule With Multiple Host Connections" on page 145 for the syntax. For an FTP rule the local port value must be 30021. to establish connection with each host specified. You need to specify the corresponding client and destination ports for each host specified. See "Static Rule With Multiple Host Connections" on page 145 for the syntax. to try to connect to any available host from the list of hosts specified. See "Static Rule with Multiple Host Selection" on page 145 for the syntax. `TARGET` - Rules that specify `TARGET` in the syntax are dynamic rules. `TARGET` indicates that end-users can specify the required destination host or hosts in the Netlet provider of the desktop. You cannot have a combination of a static host and TARGET in a single rule.

TABLE 6–1 Fields in a Netlet Rule *(Continued)*

Parameter	Description	Value
Destination Port	The port on the destination host In addition to the host and destination host, you must specify a destination port. You can specify multiple destination ports in case of multiple destination hosts. Specify multiple ports in the format `port1+port2+port3-port4+port5`. The plus (+) sign between ports numbers indicates the alternative ports for a single destination host. The minus (-) sign between port numbers is the separator between the port numbers for different destination hosts. Here, Netlet tries to connect to the first destination host specified using `port1`, `port2` and `port3` in order. If this fails, Netlet tries to connect to the second host using `port4` and `port5` in that order. You can configure multiple ports only for static rules.	

For the Gateway to get the session notification from Portal Server, add the following:

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

to the following property file

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

Types of Rules

Two types of Netlet rules are based on how the destination host is specified in the rule.

Static Rule

A static rule specifies a destination host as part of the rule. If you create a static rule, the user does not have the option to specify the required destination host. In the following example, `sesta` is the destination host.

Rule Name	Encryption Cipher	URL	Enable Download Applet	Enable Extend Session	Map Local Port to Destination Server Port
ftpstatic	`SSL_RSA_WITH_RC 4_128_MD5`	null	false	true	■ Local Port: 30021 ■ Destination Host: sesta ■ Destination Port: 21

You can configure multiple destination hosts and ports for static rules. See "Static Rule With Multiple Host Connections" on page 145 for an example.

Dynamic Rule

In a dynamic rule, the destination host is not specified as a part of the rule. The user can specify the required destination host in the Netlet provider. In the following example, `TARGET` is the placeholder for the destination host.

Rule Name	Encryption Cipher	Remote Application URL	Enable Download Applet	Enable Extend Session	Map Local Port to Destination Server Port
ftpdynamic	`SSL_RSA_WIT H_RC4_128_MD5`	null	Select checkbox	Select checkbox	■ Local Port: 30021 ■ Destination Host: TARGET ■ Destination Port: 21

Encryption Ciphers

Based on the encryption cipher, Netlet rules can be further classified as follows:

- *User Configurable Cipher Rules* - In this rule, you can specify a list of ciphers that users can choose from. These optional ciphers appear as a list in the Netlet provider. The user can choose the required cipher from the list. In the following example, the user can choose from multiple ciphers.

Rule Name	Encryption Cipher	Remote Application URL	Enable Download Applet	Enable Extend Session	Map Local Port to Destination Server Port
Telnet	`SSL_RSA_WITH_RC4 _128_SHA`	null	Select checkbox	Select checkbox	■ Local Port: 30000 ■ Destination Host: TARGET ■ Destination Port: 23
	`SSL_RSA_WITH_RC4 _128_MD5`				

Note – Although the Portal Server host may have various ciphers enabled, the user can choose only from the list that is configured as part of the Netlet rule.

See “Supported Ciphers” on page 142 for a list of the ciphers supported by Netlet.

- *Administrator Configured Cipher Rules* - In this rule, the cipher is defined as part of the Netlet rule. The user does not have the option to choose the required cipher. In the following example, the cipher is configured to be SSL_RSA_WITH_RC4_128_MD5.

Rule Name	Encryption Cipher	Remote Application URL	Enable Download Applet	Enable Extend Session	Map Local Port to Destination Server Port
Telnet	SSL_RSA_WITH_RC4_128_MD5	null	Select checkbox	Select checkbox	■ Local Port: 30000 ■ Destination Host: TARGET ■ Destination Port: 23

See “Supported Ciphers” on page 142 for a list of ciphers supported by Netlet.

Supported Ciphers

“Supported Ciphers” on page 142 lists the ciphers supported by Netlet.

TABLE 6–2 List of Supported Ciphers

Ciphers
Native VM Ciphers
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 Plugin Ciphers
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA

TABLE 6–2 List of Supported Ciphers *(Continued)*

Ciphers
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
TLS_RSA_WITH_AES_256_CBC_SHA

Backward Compatibility

Earlier versions of Portal Server did not support ciphers as part of the Netlet rules. For backward compatibility with existing rules without ciphers, a default cipher is used by the rules. An existing rule without ciphers such as:

Rule Name	Encryption Cipher	Remote Application URL	Enable Download Applet	Enable Extend Session	Map Local Port to Destination Server Port
Telnet		`telnet://localhost:30000`	Do not select checkbox	Select checkbox	■ Local Port: 30000 ■ Destination Host: TARGET ■ Destination Port: 23

is interpreted as:

Rule Name	Encryption Cipher	Remote Application URL	Enable Download Applet	Enable Extend Session	Map Local Port to Destination Server Port
Telnet	Default ciphers	`telnet://localhost:30000`	Do not select checkbox	Select checkbox	■ Local Port: 30000 ■ Destination Host: TARGET ■ Destination Port: 23

This is similar to an Administrator Configured Rule with the Encryption cipher field chosen as Default.

Note – Netlet rules cannot contain any port number higher than 64000.

Netlet Rule Examples

This section contains some examples of Netlet rules to illustrate how Netlet syntax works.

- "Basic Static Rule" on page 144
- "Static Rule With Multiple Host Connections" on page 145
- "Dynamic Rule to Invoke a URL" on page 146
- "Dynamic Rule to Download an Applet" on page 148

Basic Static Rule

This rule supports a Telnet connection from the client to the machine `sesta`.

Rule Name	Encryption Cipher	Remote Application URL	Download Applet	Extend Session	Map Local Port to Destination Server Port
myrule	`SSL_RSA_WITH_RC4_128_MD5`	null	Do not select the checkbox	true	Local Port: 1111 Destination Host: sesta Destination Port: 23

where

`myrule` is the name of the rule.

SSL_RSA_WITH_RC4_128_MD5 indicates the cipher to be used.

`null` indicates that this application is not invoked by a URL or run through the desktop.

`false` indicates that the client does not download an applet to run this application.

`true` indicates that Portal Server should not time out when the Netlet connection is active.

`1111` is the port on the client where Netlet listens for a connection request from the destination host.

`sesta` is the name of the recipient host in the Telnet connection.

`23` is the port number on the destination host for the connection, in this case the well-known port for Telnet.

The desktop Netlet provider does not display a link, but Netlet automatically starts and listens on the port specified (`1111`). Instruct the user to start the client software - in this case a Telnet session that connects to `localhost` on port 1111.

For example, to start the Telnet session, the client needs to type the following on the UNIX command line in a terminal:

```
telnet localhost 1111
```

Static Rule With Multiple Host Connections

This rule supports a Telnet connection from the client to two machines, `sesta` and `siroe`.

Rule Name	Encryption Cipher	Remote Application URL	Enable Download Applet	Enable Extend Session	Map Local Port to Destination Server Port
myrule	`SSL_RSA_WITH_RC4_128_MD5`	null	Do select the checkbox	Select the checkbox	■ Local Port: 1111–1234 ■ Destination Host: sesta-siroe ■ Destination Port: 23

where

23 is the port number on the destination host for the connection– reserved port for Telnet.

`1111` is the port on the client where Netlet listens for a connection request from the first destination host `sesta`.

`1234` is the port on the client where Netlet listens for a connection request from the second destination host `siroe`.

The first six fields in this rule are the same as in "Basic Static Rule" on page 144. The difference is that three more fields identify the second destination host.

When you add additional targets to a rule, you must add three fields, `local port`, `destination host`, and `destination port`, for each new destination host.

Note – You can have multiple sets of three fields describing the connection to each destination host. Listen port numbers which are less than 2048 must not be used if the remote client is UNIX-based because low numbered ports are restricted and you must be root to start a listener.

This rule works the same as the previous rule. The Netlet provider does not display any link, but Netlet automatically starts and listens on the two ports specified (1111 and 1234). The user needs to start the client software, in this case a Telnet session that connects to `localhost` on port 1111 or the `localhost` on port 1234 to connect to the host in the second example.

Static Rule with Multiple Host Selection

Use this rule to specify multiple alternative hosts. If connection to the first host in the rule fails, Netlet tries to connect to the second host specified and so on.

Rule Name	Encryption Cipher	Remote Application URL	Enable Download Applet	Enable Extend Session	Map Local Port to Destination Server Port
gojoe	`SSL_RSA_WITH_RC4_128_MD5`	gojoe.html	■ Client Port: 8000 ■ Server Host: gojoeserver ■ Server Port: 8080	Select the checkbox	■ Local Port: 10491 ■ Destination Host: siroe+sesta ■ Destination Port: 35+26+491-35+491

where

`10491` is the port on the client where Netlet listens for a connection request from the destination host.

Netlet tries to establish connection with `siroe` on port `35`, port `26` and port `491` in the same order, depending on which one is available.

If connections to `siroe` are not possible, Netlet tries to connect to `sesta` on port `35` and `491` in the same order.

The plus (+) sign between hosts indicates alternative hosts.

The plus (+) sign between ports numbers indicates the alternative ports for a single destination host.

The minus (-) sign between port numbers is the separator between the port numbers for different destination hosts.

Note – Connections to hosts provided in the chain is attempted serially. For example, if the rule is `siroe+ sesta`, then a connection to `siroe` is attempted first. If the connection fails then the connection to `sesta` is attempted . If the hosts listed first in the rule are physically unavailable in an active network, the time taken to connect to the next available host will increase as the number of unavailable hosts in the rule increases.

Dynamic Rule to Invoke a URL

This rule enables a user to configure the destination host required, enabling the user to telnet to various hosts over Netlet.

Rule Name	Encryption Cipher	Remote Application URL	Enable Download Applet	Enable Extend Session	Map Local Port to Destination Server Port
myrule	`SSL_RSA_WITH_RC4_128_MD5`	telnet://localhost:30000	Do not select the checkbox	Select the checkbox	■ Local Port: 30000 ■ Destination Host: TARGET ■ Destination Port: 23

where

`myrule` is the name of the rule.

SSL_RSA_WITH_RC4_128_MD5 indicates the cipher to be used.

`telnet://localhost:30000` is the URL invoked by the rule.

`false` indicates that no applets are to be downloaded.

`Extend Session(true)` indicates that the Portal Server should not time out when the Netlet connection is active.

`30000` is the port on the client where Netlet listens for connection requests for this rule.

`TARGET` indicates that the destination host needs to be configured by the user using the Netlet provider.

`23` is the port on the destination host opened by Netlet, in this case the well-known port for Telnet.

▼ To Run Netlet After a Rule is Added

After this rule is added, the user must complete some steps to get Netlet running as expected. The user needs to do the following on the client side:

1. **Click Edit in the Netlet provider section of the standard Portal Server desktop.**

   The new Netlet rule is listed under Rule Name in the Add New Target section.

2. **Choose the rule name and type the name of the destination host.**

3. **Save the changes.**

   The user returns to the desktop with the new link visible in the Netlet provider section.

4. **Click the new link.**

   A new browser is launched that goes to the URL given in the Netlet rule.

Note – You can add more than one destination host for the same rule by repeating these steps. Only the last link selected is active.

Dynamic Rule to Download an Applet

This rule defines a connection from the client to hosts that are dynamically allocated. The rule downloads a GO-Joe applet from the server on which the applet is located, to the client.

Rule Name	Encryption Cipher	Remote Application URL	Enable Download Applet	Extend Session	Map Local Port to Destination Server Port
gojoe	SSL_RSA_WITH_RC4_128_MD5	/gojoe.html	■ Client Port: 8000 ■ Server Host: gojoeserver ■ Server Port: 8080	Select the checkbox	■ Local Port: 3399 ■ Destination Host: TARGET ■ Destination Port:58

where

gojoe is the name of the rule.

SSL_RSA_WITH_RC4_128_MD5 indicates the cipher to be used.

/gojoe.html for example is the path of the HTML page containing the applet, the path should be relative to the documentation root of the web container on which portal is deployed.

8000:server:8080 indicates that port 8000 is the destination port on the client to receive the applet, gojoeserve is the name of the server providing the applet, and 8080 is the port on the server from which the applet is downloaded.

Extended Session (true) indicates that the Portal Server should not time out when the Netlet connection is active.

3399 is the port on the client where Netlet listens for connection requests of this type.

TARGET indicates that the destination host needs to be configured by the user using the Netlet provider.

58 is the port on the destination host opened by Netlet, in this case the port for GoJoe. Port 58 is the port that the destination host listens to for its own traffic. Netlet passes information to this port from the new applet.

Sample Netlet Rules

"Sample Netlet Rules" on page 149 lists sample Netlet rules for some common applications.

The table has 7 columns corresponding to the following fields in a Netlet rule: Rule Name, URL, Download Applet, Local Port, Destination Host, Destination Port. The last column includes a description of the rule.

Note – "Sample Netlet Rules" on page 149 does not list the Cipher and Extend Session fields of the Netlet rule. Assume these to be "SSL_RSA_WITH_RC4_128_MD5" and "true" for the samples provided.

TABLE 6–3 Sample Netlet Rules

Rule Name	Remote Application URL	Enable Download Applet	Map Local Port to Destination Server Port	Description
IMAP	null	Do not select the checkbox	■ Local Port: 10143 ■ Destination Host: imapserver ■ Destination Port: 143	The Netlet `local port` on the client side need not be the same as the `destination port` on the server side. If you use anything other than the standard IMAP and SMTP ports, make sure that the client is configured to connect on a port that is different from the standard port. Solaris client users cannot connect to port numbers lower than 1024 unless they are running as root.
SMTP	null	Do not select the checkbox	■ Local Port: 10025 ■ Destination Host: smtpserver ■ Destination Port: 25	
Lotus Web Client	null	Do not select the checkbox	■ Local Port: 80 ■ Destination Host: lotus-server ■ Destination Port: 80	This rule tells Netlet to listen for the client on port 80, and connect to the server lotus-server on port 80. A requirement of the Lotus Web Client is that the client listen port must match the server port.

TABLE 6–3 Sample Netlet Rules *(Continued)*

Rule Name	Remote Application URL	Enable Download Applet	Map Local Port to Destination Server Port	Description
Lotus Notes Non-web Client	null	Do not select the checkbox	■ Local Port: 1352 ■ Destination Host: lotus-domino ■ Destination Port: 1352	With this rule, the Lotus Notes client can connect to a Lotus Domino server through Netlet. Ensure that when the client tries to connect to the server it must not point to `localhost` as the server name. It must point to the actual server name of the Lotus Domino server. The server name must be the same as the system name for the server. The client must resolve that name to `127.0.0.1` when using Netlet. Two ways to accomplish this are: ■ Set the server name to point to `127.0.0.1` in the client host table. ■ Export a DNS entry of the name of the server that points to `127.0.0.1`. The server name must be the same server name that was used to configure the Domino server during setup.

TABLE 6–3 Sample Netlet Rules *(Continued)*

Rule Name	Remote Application URL	Enable Download Applet	Map Local Port to Destination Server Port	Description
Microsoft Outlook and Exchange Server This will not work for Windows NT, 2000 and XP. Use Outlook Web Access through the Rewriter for Windows NT, 2000, and XP.	null	Do not select the checkbox	■ Local Port: 135 ■ Destination Host: exchange ■ Destination Port: 135	This rule tells Netlet to listen at port 135 on the client and connect to the server `exchange` on port 135. The Outlook client uses this port to make an initial attempt to contact the Exchange server and determine what subsequent ports to use to talk to the server. On the client machine: ■ The user must change the hostname of the Exchange server that is configured in the Outlook client to `localhost`. The location of this option varies with the version of Outlook. ■ The user must map the hostname (single and fully qualified) of the Exchange server to the IP address `127.0.0.1` using the hosts file. ■ On Windows 95 or 98, the file is in `\\Windows\\Hosts` ■ On Windows NT4, the file is in `\\WinNT\\System32\\drivers\\etc\\Hosts`. The entry looks like this: `127.0.0.1 exchange exchange.company.com` The Exchange server sends back its own name to the Outlook client. This mapping ensures that the Outlook client uses the Netlet client to connect back to the server.

TABLE 6–3 Sample Netlet Rules *(Continued)*

Rule Name	Remote Application URL	Enable Download Applet	Map Local Port to Destination Server Port	Description
FTP	null	Do not select the checkbox	■ Local Port: 30021 ■ *Destination Host: your-ftp_server.your-domain* ■ Destination Port: 21	You can provide FTP service to a single FTP Server, with controlled end-user accounts. This will ensure secure remote FTP transfers from an end-user system to a single location. Without a username, an FTP URL is interpreted as an anonymous FTP connection. You *must* define port 30021 as the local port for your Netlet FTP rule. Dynamic FTP is supported using a Netlet connection.
Netscape 4.7 Mail Client	null	Do not select the checkbox	■ Local Port: 30143, 30025. ■ Destination Host: TARGET ■ Destination Port: 10143	In the Netscape client, the user needs to specify: `localhost:30143` for IMAP or incoming mails `localhost:30025` for SMTP or outgoing mails
Graphon	third_party/xsession_start.html	Select the checkbox	■ Local Port: 10491 ■ Destination Host: TARGET ■ Destination Port: 491	This is the rule used to access Graphon through the Netlet. `xsession_start.html` is bundled with Graphon.
Citrix	third_party/citrix_start.html	Select the checkbox	■ Local Port: 1494 ■ Destination Host: TARGET ■ Destination Port: 1494	This is the rule used to access Citrix through the Netlet. `citrix_start.html` is bundled with Citrix.
RemoteControl	third_party/pca_start.html	Select the checkbox	■ Local Port: 5631 5632 ■ Destination Host: TARGET TARGET TARGET ■ Destination Port: 5631 5632	This is the rule used to access Remote Control through Netlet. `pca_start.html` is bundled with Remote Control.

Netlet Logging Information

Client side logs for the netlet applet or the jws appear on the java console of the client.

Server side logs for the netlet appear in the `portal.0.0.log` file present under the `/var/opt/SUNWportal/portals/<portal_ID>/logs/<INSTANCE_ID>` directory.

Running Netlet in a Sun Ray Environment

If you want to run an application which requires the applet to be downloaded to the client machine on a Sun Ray environment, you need to change the HTML file. Here is a sample file showing you the necessary modifications that need to be done.

New HTML File

```
<!-- @(#)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>
```

Deprecated HTML File

```
<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>
```

PART II

Configuring the Secure Remote Access Server

Most attributes can be set using the options available under the Secure Remote Access tab in the Portal Server management console. Any new organization or user that is created inherits these values, by default.

You can configure the attributes related with Secure Remote Access at the organization, role, and user levels, with the following exceptions:

- Conflict Resolution Level cannot be set at the user level. See "Setting Conflict Resolution" on page 30.
- MIME types Configuration File Location attribute can be set only at the organization level.

Values set at the level of an organization are inherited by all the roles and users under it. Values set at the user level override the values set at the organization or role levels.

You can make changes to the attribute values at the Service Configuration level. These new values are reflected only when new organizations are added.

This section has the following chapters:

- Chapter 7, "Configuring the Secure Remote Access Server Access Control"
- Chapter 8, "Configuring the Secure Remote Access Gateway"
- Chapter 9, "Configuring Rewriter in the Gateway Service"

- Chapter 10, “Working with Certificates”
- Chapter 11, “Configuring the Netlet”
- Chapter 12, “Configuring Netlet With Private Domain Certificates”
- Chapter 13, “Configuring Proxylet”
- Chapter 14, “Configuring NetFile”
- Chapter 15, “Configuring Secure Socket Layer Accelerators”

CHAPTER 7

Configuring the Secure Remote Access Server Access Control

This chapter describes allowing or denying access to the users from the Sun Java System Portal Server administration console.

Configuring Access Control

You can specify the list of URLs that end users cannot access through the Gateway using this field. The Gateway checks the Denied URLs list before checking the Allowed URLs list.

You can specify all the URLs that can be accessed by the end user through the Gateway. By default, this list has a wild card entry (*), which means that all URLs can be accessed. If you want to allow access to all URLs, and restrict access only to specific URLs, add the restricted URLs to the Denied URL list. In the same way, if you want to allow access only to specific URLs, leave the Denied URLs field blank, and specify the required URLs in the Allowed URLs field.

The Access Control service in SRA software allows you to control the single sign-on feature for various hosts. For the single sign-on feature to be available, the Enable HTTP Basic Authentication option in the Gateway service must be enabled..

With the Access Control service, you can disable single sign-on for certain hosts. This means that an end user needs to authenticate each time to connect to the hosts that require HTTP basic authentication, unless you enable single sign-on per session.

If you have disabled single sign-on for a certain host, the user can reconnect to that host within a single Portal Server session. For example, assume that you have disabled single sign-on to `abc.sesta.com`. The first time the user connects to this site, authentication is required. The user may browse other pages and return to this page later, and if the page is in the same Portal Server session, authentication is not required.

▼ To Configure the Access Control

1. **Log onto the Portal Server administration console as administrator.**
2. **Select the Secure Remote Access tab.**
3. **Select the Access Control tab.**
4. **Modify the following attributes:**

Attribute Name	Description
COS Priority	Specifies the value used to determine the inheritance of the attribute value. For more information on this attribute, see the Sun Java System Directory Server Administration Guide.
Single Sign On per Session	Select the Enable checkbox to enable a single-sign on session.
Single Sign On Disabled Hosts	Enter the host name in the format `abc.siroe.com`.
Allowed Authentication Levels	Enter the allowed authentication levels. Use an asterisk to allow all levels. The default value is asterisk.
Allow/Deny access to URL's	Enter the URL to allow or deny access through the Gateway in the in the URL field. The format for entering the URL is: `http://abc.siroe.com`. Under Action drop down list, click the appropriate Allow or Deny option. You can also use regular expressions such as `http://*.siroe.com`. In this case, users are denied access to all hosts in the `siroe.com` domain. The Gateway first checks the URLs that have been denied access before checking the allowed URLs list. **Note –** The Allowed URLs field has a * by default which means that all URLs can be accessed through the Gateway.

Note – When you install SRA, the Access Control l service is not available to all users by default. This service is enabled only to the `amadmin` user that is created by default during installation. Other users cannot access the desktop through the Gateway without this service. Log in as `amadmin`, and assign this service to all the users.

5. **Click Save to complete.**

CHAPTER 8

Configuring the Secure Remote Access Gateway

This chapter describes configuring the Gateway attributes from the Sun Java System Portal Server administration console.

This chapter contains the following sections:

- "Configuring the Profile Core Options" on page 159
- "Configuring the Deployment Options" on page 165
- "Configuring the Security Options" on page 168
- "Configuring the Performance Options" on page 171
- "Configuring the Rewriter Options" on page 173
- "Configuring the Map Parser to MIME Types" on page 175
- "Configuring the Map URIs to RuleSets" on page 173
- "Configuring Personal Digital Certificate Authentication" on page 176
- "Configuring Gateway Attributes Using the Command Line Options" on page 180

Before you Begin

- To create a gateway profile, see "Creating a Gateway Profile" on page 32

Configuring the Profile Core Options

This section explains the following tasks:

- "Configuring the Startup Mode" on page 159
- "Configuring the Core Components" on page 161

Configuring the Startup Mode

The Gateway runs in HTTPS mode after installation if you have chosen to run the Gateway in the HTTPS mode during installation. In the HTTPS mode, the Gateway accepts SSL connections from browsers and rejects non-SSL connections. However, you can also configure the Gateway to run in HTTP mode. This speeds Gateway performance as the overhead involved in managing SSL sessions and encrypting and decrypting the SSL traffic are not involved.

▼ To Configure the Startup Mode

1. **Log onto the Portal Server administration console as administrator.**
2. **Select the Secure Remote Access tab, and click the profile name to modify its attributes.**
3. **Select the Core tab.**
4. **Modify the following attributes:**

HTTP Connections	Select the HTTP Connections checkbox to allow Gateway to accept non-SSL connections.
HTTP Port	Enter the HTTP port number. The default value is 80.
HTTPS Connections	Select the HTTPS Connections checkbox to allow Gateway to accept SSL connections. By default, this options is selected.
HTTPS Port	Enter the HTTPS port number. The default value is 443.

Note – The following attributes can be modified using "psadmin set-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference*

/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 \n Original URI Protocol sunPortalGatewayEnableCookieManager=Store External Server Cookies
- sunPortalGatewayMarkCookiesSecure=Mark Cookies as secure

5 **Restart the Gateway from a terminal window:**

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

Configuring the Core Components

Netlet enables users to securely run common TCP/IP services over insecure networks such as the Internet. You can run TCP/IP applications (such as Telnet and SMTP), HTTP applications, and any fixed port applications. If Netlet is enabled, the Gateway needs to determine whether the incoming traffic is Netlet traffic or Portal Server traffic. Disabling Netlet reduces this overhead since the Gateway assumes that all incoming traffic is either HTTP or HTTPS traffic. Disable Netlet only if you are sure you do not want to use any application with Portal Server.

▼ To Configure the Components

1 Log onto the Portal Server administration console as administrator.

2 Select the Secure Remote Access tab and click the profile name to modify its attributes.

3 Select the Core tab.

4 Modify the following attributes:

Attribute Name	Description
Netlet	Select the Enable checkbox to initiate the Netlet service. By default this option is selected.
Proxylet	Select the Enable checkbox to initiate the Proxylet service. By default this option is selected.

5 Restart the Gateway from a terminal window using the following command options:

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

Configuring the Basic Options

About the Cookie Management Attribute

Many web sites use cookies to track and manage user sessions. When the Gateway routes requests to web sites that set cookies in the HTTP header, the Gateway either discards or passes-through those cookies in the following manner:

- Cookies are not rewritten if Enable Cookie Management attribute is not selected in the Gateway service. So, the cookies from the browser might not reach the intranet hosts and vice-versa.
- Gateway rewrites cookies if the Enable Cookie Management attribute is selected. Gateway ensures that the cookies from the browser reach the intended intranet hosts and vice-versa.

This setting does not apply to the cookies used by Portal Server to track Portal Server user sessions. The setting is controlled by the configuration of the URLs to which User Session Cookie is Forwarded URL option.

This setting applies to all web sites that the user is permitted to access (that is, you cannot choose to discard cookies from some sites and retain cookies from others).

Note – Do not remove URLs from the Cookie Domain list, even in a Gateway without cookies. See the *Access Manager Administration Guide* for information on the Cookie Domain list.

About the HTTP Basic Authentication Attribute

HTTP basic authentication can be set in the Gateway service.

Web sites may be protected with HTTP Basic Authentication, requiring visitors to enter a username and password before viewing the site (the HTTP response code is 401 and WWW-authenticate: BASIC). Portal Server can save the username and password so that users need not re-enter their credentials when they revisit BASIC-protected web sites. These credentials are stored in the user profile on the directory server.

This setting does not determine whether or not a user may visit BASIC-protected sites, but only whether the credentials the user enters are saved in the user\qs profile.

This setting applies to all web sites that the user is permitted to access (that is, HTTP basic authentication caching cannot be enabled for some sites and disabled for others).

Note – Browsing to URLs served by Microsoft\qs Internet Information Server (IIS) protected by Windows NT challenge/response (HTTP response code 401, WWW-Authenticate: NTLM) instead of BASIC authentication is not supported.

You can also enable single sign-on using the Access Control service in the administration console.

About the Portal Servers Attribute

You can configure multiple Portal Servers for the Gateway to service requests. While installing the Gateway, you would have specified the Portal Server that the Gateway needs to work with. This Portal Server is listed in the Portal Servers field by default. You can add more Portal Servers to the list in the format `http://`*portal- server-name:port number*. The Gateway tries to contact each of the Portal Servers listed in a round robin manner to service the requests.

About the URLs to Which User Session Cookie is Forwarded Attribute

Portal server utilizes a cookie to track user sessions. This cookie is forwarded to the server when the Gateway makes HTTP requests to the server (for example, when the desktop servlet is called to generate the user\qs desktop page). Applications on the server use the cookie to validate and identify the user.

The Portal Server\qs cookie is not forwarded to HTTP requests made to machines other than the server, unless URLs on those machines are specified in the URLs to which User Session Cookie is Forwarded list. Adding URLs to this list therefore enables servlets and CGIs to receive the Portal Server\qs cookie and use the APIs to identify the user.

URLs are matched using an implicit trailing wildcard. For example, the default entry in the list:

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

causes the cookie to be forwarded to all URLs starting with `http://server:8080`.

Adding:

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

causes the cookie to be forwarded to all URLs starting with that exact string.

For this example, the cookie is not forwarded to any URLs starting with "`http://newmachine.eng/subdir`", since this string does not start with the exact string in the forward list. To have cookies forwarded to URLs starting with this variation of the machine\qs name, an additional entry has to be added to the forward list.

Similarly, the cookie is not forwarded to URLs starting with "`https://newmachine.eng.siroe.com/subdir`" unless an appropriate entry is added to the list.

About the Obtain Session from URL Attribute

When the Obtain Session from a URL option is selected, session information is encoded as part of the URL, whether cookies are supported or not. This means that the Gateway uses the session information found in the URL for validation rather than using the session cookie that is sent from the client's browser.

▼ To Configure the Basic Options

1. **Log onto the Portal Server administration console as administrator.**
2. **Select the Secure Remote Access tab and click the profile name to modify its attributes.**
3. **Select the Core tab.**
4. **Modify the following attributes:**

Attribute Name	Description
Cookie Management	Select the Enable checkbox to enable cookie management. By default, this option is selected.

Attribute Name	Description
HTTP Basic Authentication	Select the Enable HTTP Basic Authentication checkbox to enable HTTP basic authentication.
Portal Servers	Enter the Portal Server in the format `http://portal-server-name:port-number` in the field and click Add. Repeat this step to add more Portal Server to the Portal Server list.
URLs to which User Session Cookie is Forwarded	Enter the URL to which User Session Cookie is Forwarded and click Add. Repeat this step to add more URLs to the URLs to which the User Session is Forwarded list.
Gateway Minimum Authentication Level	Enter the authentication level. By default, an asterisk is added to allow authentication at all levels.
Obtain Session from URL	Select Yes to retrieve information on a session from a URL. By default, the No option is selected.

Configuring the Deployment Options

Configuring the Proxy Settings

▼ To Configure the Proxy Settings

1. **Log onto the Portal Server administration console as administrator.**
2. **Select the Secure Remote Access tab and click the profile name to modify its attributes.**
3. **Select the Deployment tab.**
4. **Modify the following attributes:**

Attribute Name	Description
Use Proxy	Select the Use Proxy checkbox to enable the usage of web proxies.

<table>
<tr><th>Attribute Name</th><th>Description</th><th></th></tr>
<tr><td>Webproxy URLs</td><td>Enter the required URL in the Use Webproxy URLs edit box in the format <code>http://host name.subdomain.com</code>, and then cClick Add.
The URL is added to the Use Webproxy URLs list.</td><td>You can specify that the Gateway needs to contact certain URLs only through the webproxies listed in the Proxies for Domains and Subdomains list, even if the Use Proxy option is disabled. You need to specify these URLs in the Use Webproxy URLs field. See “Specifying a Proxy to Contact Access Manager” on page 34 for details on how this value affects the usage of proxies.</td></tr>
<tr><td>Proxies for Domains and Subdomains</td><td>The entry is added to the Proxies for Domains and Subdomains list box.
The format for entering the proxy information is as follows:
<code>domainname proxy1:port1|subdomain1 proxy2:port2|subdomain2 proxy3:port3|* proxy4:port4</code>
* indicates that the proxy defined after the * needs to be used for all domains and subdomains other than those specifically mentioned.
If you do not specify the port for the proxy, port 8080 is used by default.</td><td>See “Specifying a Proxy to Contact Access Manager” on page 34 for details on how the proxy information is applied to various hosts.</td></tr>
<tr><td>Proxy Password List</td><td>In the Proxy Password List field, enter the information for each proxy server, and then click Add.
The format for entering the proxy information is as follows:
<code>proxyserver|username|password</code>
The <code>proxyserver</code> corresponds to the proxy server defined in the Proxies for Domains and Subdomains list.</td><td>You need to specify the user name and password required for the Gateway to authenticate to a specified proxy server, if the proxy server requires authentication to access some or all the sites.</td></tr>
<tr><td>Automatic Proxy Configuration support</td><td>Select the Enable Automatic Proxy Configuration Support checkbox to enable PAC support.</td><td>If you select the option Enable Automatic Proxy Configuration, the information provided in the Proxies for Domains and Subdomains field is ignored. The Gateway uses the Proxy Automatic Configuration (PAC) file only for intranet configuration. See “Using Automatic Proxy Configuration” on page 46 for information on PAC files.</td></tr>
<tr><td>Automatic Proxy Configuration File location</td><td>In Location field, enter the name and location of the PAC file.</td><td></td></tr>
</table>

Configuring the Rewriter Proxy and Netlet Proxy

About NetLet Proxy

The Netlet proxy enhances the security of Netlet traffic between the Gateway and the intranet by extending the secure tunnel from the client, through the Gateway to the Netlet proxy that resides in the intranet.If the Netlet proxy is enabled, the Netlet packets are decrypted by the Netlet proxy and then sent to the destination server. This reduces the number of ports required to be opened in the firewall.

About Rewriter Proxy

The Rewriter proxy enables secure HTTP traffic between the Gateway and intranet. If you do not specify a Rewriter proxy, the Gateway component makes a direct connection to the intranet when a user tries to access a machine on the intranet.The Rewriter proxy does not run automatically after installation. You need to enable the Rewriter proxy as described below.

▼ To Configure the Rewriter Proxy and Netlet Proxy

1. **Log onto the Portal Server administration console as administrator.**
2. **Select the Secure Remote Access tab and click the profile name to modify its attributes.**

   Note – Ensure that the Rewriter proxy and the Gateway use the same gateway profile.

3. **Select the Deployment tab.**
4. **Modify the following attributes:**

Attribute Name	Description
Rewriter Proxy	Select the Rewriter Proxy checkbox to enable the Rewriter proxy service.
Rewriter Proxy List	a. Enter the host and port in the Rewriter Proxies edit box, in the format `hostname:port`. **Tip –** To determine if the port desired is available and unused, from the command line, enter: **`netstat -a | grep`** *port-number* **`| wc -l`** *port-number* is the required port. b. Click Add.

Attribute Name	Description
Netlet Proxy	Select the Enable Netlet Proxy checkbox to enable the Netlet proxy service.
Netlet Proxy Hosts	a. Enter the Netlet proxy host and port in the Netlet Proxy Hosts field, in the format `hostname:port`. **Tip –** To determine if the port desired is available and unused, from the command line, enter: **`netstat -a	grep`** *port-number* **`| wc -l`** *port-number* is the required port. b. Click Add.
Netlet Tunneling via Web Proxy	Select the Enable Netlet Tunneling via Web Proxy checkbox to enable tunneling.

5 **Run *portal-server-install-root*`/SUNWportal/bin/certadmin` on the server to create a certificate for the Rewriter proxy.**

You need to do this step only if you have not chosen to create a certificate while installing the Rewriter proxy.

6 **Log in as root to the machine where the Rewriter proxy is installed and start the Rewriter proxy:**

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

7 **Log in as root to the machine where the Gateway is installed and restart the Gateway:**

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

Configuring the Security Options

Configuring the PDC and Non Authenticated URLs

▼ To Configure the PDC and Non Authenticated URLs

1 **Log onto the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab and click the profile name to modify its attributes.**

3 **Select the Security tab.**

4 **Modify the following attributes:**

Attribute Name	Description
Certificate-enabled Gateway hosts	a. Add the Gateway name to the Certificate-enabled Gateway hosts. Add the Gateway in the format `host1.sesta.com`. b. Click Add.
Non-authenticated URLs	You can specify that some URLs do not need authentication. These are normally directories that contain images. In the Non-Authenticated URLs field, enter the required folder path in the format `folder/subfolder`. URLs that are not fully-qualified (for example, /images) are treated as portal URLs. To add a non-portal URL, fully qualify the URL, click Add to add this entry to the Non-Authenticated URLs list.
Trusted SSL Domains	In the Trusted SSL Domains field, enter the domain names and click Add.

Configuring the TLS and SSL Options

▼ To Configure the TLS and SSL Options

1. **Log onto the Portal Server administration console as administrator.**

2. **Select the Secure Remote Access tab and click the profile name to modify its attributes.**

3. **Select the Security tab.**

4. **Modify the following attributes:**

Attribute Name	Description
40-bit Encryption	Select this option if you want to allow 40-bit (weak) Secure Sockets Layer (SSL) connections. If you do not select this option, only 128-bit connections are supported. If you disable this option, the user needs to ensure that the browser is configured to support the required connection type. **Note –** The user needs to do the following in the case of Netscape Navigator 4.7x: a. Select Security Info under Tools in the Communicator menu. b. Click the Navigator link in the left pane. c. Click Configure SSL v2 or Configure SSL v3 under Advanced Security (SSL) Configuration. d. Enable the required ciphers.
Null Ciphers	Select the Enable Null Ciphers checkbox to enable null ciphers.
SSL Cipher Selection	Secure Remote Access supports a number of standard ciphers. You have the option of supporting all the pre-packaged ciphers, or selecting the required ciphers individually. You can select specific SSL ciphers for each Gateway instance. If any of the selected ciphers is present at the client site, the SSL handshake occurs successfully.
SSL Version 2.0	Select the Enable SSL Version 2.0 checkbox to enable version 2.0. This option is enabled by default. You can enable or disable SSL version 2.0. Disabling SSL 2.0 means that browsers that support only the older SSL 2.0 cannot authenticate to Secure Remote Access. This ensures a greater level of security.
SSL2 Ciphers	Select the Enable SSL Cipher Selection checkbox option. You can select the required ciphers from the list of SSL ciphers.
SSL Version 3.0	You can enable or disable SSL version 3.0. Disabling SSL 3.0 means that browsers that support only the SSL 3.0 cannot authenticate to SRA software. This ensures a greater level of security. Select the Enable SSL Version 3.0 checkbox to enable version 3.0.
SSL3 Ciphers	Select the Enable SSL Cipher Selection checkbox option. You can select the required ciphers from the list of SSL3 ciphers.
TLS Ciphers	Select the Enable SSL Cipher Selection checkbox option. You can select the required ciphers from the list of TLS ciphers.

Configuring the Performance Options

Configuring the Timeouts and Retries

▼ To Configure the Timeouts and Retries

1. **Log onto the Portal Server administration console as administrator.**

2. **Select the Secure Remote Access tab and click the profile name to modify its attributes.**

3. **Select the Performance tab.**

4. **Modify the following attributes:**

Attribute Name	Description
Server Retry Interval (seconds)	Specify the time interval in seconds between requests to try to start the Portal Server, Rewriter proxy, or Netlet proxy if it becomes unavailable (such as a crash or it was brought down).
Gateway Timeout (seconds)	Specify the time interval in seconds after which the Gateway times out its connection with the browser. In the Gateway Timeout field, specify the interval required in seconds.
Cached Socket Timeout (seconds)	Specify the time interval in seconds after which the Gateway times out its connection with the Portal Server.

Configuring the HTTP Options

▼ To Configure the HTTP Options

1. **Log onto the Portal Server administration console as administrator.**

2. **Select the Secure Remote Access tab and click the profile name to modify its attributes.**

3. **Select the Performance tab.**

4. **Modify the following attributes:**

Attribute Name	Description
Maximum Thread Pool Size	Specify the required number of threads. You can specify the maximum number of threads that can be pre-created in the Gateway thread pool.
Persistent HTTP Connections	Select the Enable Persistent HTTP Connections checkbox to enable HTTP connections. You can enable HTTP persistent connections at the Gateway to prevent sockets being opened for every object (such as images and style sheets) in the Web pages.
Maximum Number of Requests per Peristent Connection	Enter the maximum number of requests.
Timeout for Persistent Socket Connections (seconds)	Enter the required timeout in seconds.
Grace Timeout to Account for Turnaround Time (seconds)	Enter the required grace timeout in seconds. This is the round-trip time for the network traffic between the client (browser) and the Gateway. ■ Time taken for the request to reach the gateway after the browser has sent it ■ Time between gateway sending the response and the browser actually receiving it This is dependent on factors such as network conditions and the client's connection speed.
Maximum Connection Queue Length	Specify the maximum concurrent connections that the Gateway should accept. Specify the required number of connections.

Monitoring the Secure Remote Access Performance

Monitoring allows administrators to assess the performance of different components of the Secure Remote Access.

▼ To Monitor Secure Remote Access Performance

1. **Log in to the Portal Server management console.**

2. **Select the Secure Remote Access tab, and click Monitoring in the submenu.**

3. **In the Monitoring page, select a proxy instance from the drop-down menu.**

4. **Select an attribute in the MBeans table to view performance values.**

Configuring the Rewriter Options

Configuring the Basic Options

▼ To Configure the Basic Options

1. **Log onto the Portal Server administration console as administrator.**
2. **Select the Secure Remote Access tab and click the profile name to modify its attributes.**
3. **Select the Rewriter tab.**
4. **Modify the following attributes:**

Attribute Name	Description
Rewriting of All URIs	Select the Enable Rewriting of All URIs checkbox to enable the Gateway to rewrite all URLs. If you enable the Enable Rewriting of All URIs option in the Gateway service, Rewriter rewrites any URL without checking against the entries in the Proxies for Domains and Subdomains list. Entries in the Proxies for Domains and Subdomains list are ignored.
URIs Not to Rewrite	Add the URI in the edit box. **Note –** Adding #* to this list allows URIs to be rewritten, even when the href rule is part of the ruleset.

Configuring the Map URIs to RuleSets

Rulesets are created in the Rewriter service under Portal Server Configuration in the Portal Server management console. See the *Portal Server Administration Guide* for details.

After the ruleset is created, you associate a domain with the ruleset using the Map URIs to RuleSets field. The following two entries are added by default to the Map URIs to RuleSets field:

- *://*.Sun.COM/portal/*|default_gateway_ruleset

  where sun.com is the install domain of the portal and /portal is the portal install context
- *|generic_ruleset

This means that for all pages from the default domain, the default Gateway ruleset is applied. For all other pages, the generic ruleset is applied. The default Gateway ruleset and the generic ruleset are pre-packaged rulesets.

Note – For all the content appearing on the desktop, the ruleset for the default domain is used, irrespective of where the content is fetched from.

For example, assume that the desktop is configured to scrape the content from the URL `yahoo.com`. The Portal Server is in `sesta.com`. The ruleset for `sesta.com` is applied to the fetched content.

Note – The domain for which you specify a ruleset must be listed in the Proxies for Domains and Subdomains list.

▼ To Configure the Map URIs to RuleSets

1. **Log onto the Portal Server administration console as administrator.**

2. **Select the Secure Remote Access tab, and click the profile name to modify its attributes.**

3. **Select the Rewriter tab.**

4. **Modify the following attributes:**

Attribute Name	Description

URI	Enter the required domain or host name and the ruleset in the Map URIs to RuleSets field and click Add.
	The entry is added to the Map URIs to RuleSets field.
	The format for specifying the domain or host name and the ruleset is as follows:
	`domain-name|ruleset-name`
	For example:
	`eng.sesta.com|default`
	Note – `The order of priority for applying the ruleset is hostname-subdomain-domain.`
	An example of entries in the Domain-based rulesets list is:
	`sesta.com|ruleset1`
	`eng.sesta.com|ruleset2`
	`host1.eng.sesta.com|ruleset3`
	■ `ruleset3` is applied for all pages on `host1`.
	■ `ruleset2` is applied for all pages in the `eng` subdomain, except for pages retrieved from `host1`.
	■ `ruleset1` is applied for all pages in the `sesta.com` domain, except for pages retrieved from the `eng` subdomain, and from `host1`.

Configuring the Map Parser to MIME Types

Rewriter has four different parsers to parse the web pages based on the content type - HTML, JAVASCRIPT, CSS and XML. Common MIME types are associated with these parsers by default. You can associate new MIME types with these parsers in the Map Parser to MIME Types field of the Gateway service. This extends Rewriter functionality to other MIME types.

Separate multiple entries with a semicolon or a comma (";" or ",".)

For example:

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

means any content with these MIMEs are sent to the HTML Rewriter and HTML Rules would be applied to rewrite the URLs.

Tip – Removing unnecessary parsers from the MIME mappings list can increase the speed of operation. For example, if you are sure that the content from a certain intranet does not have any JavaScript, you can remove the JAVASCRIPT entry from the MIME mappings list.

▼ To Configure the Map Parser to MIME Types

1 Log onto the Portal Server administration console as administrator.

2 Select the Secure Remote Access tab and click the profile name to modify its attributes.

3 Select the Rewriter tab.

4 Modify the following attributes:

Attribute Name	Description
Parsers	a. In the Map Parser to MIME Types field, add the required MIME type in the Edit box. Use a semicolon or comma to separate multiple entries. Specify the entry in the format `HTML=text/html;text/htm` b. Click Add to add the required entry to the list.

Configuring Personal Digital Certificate Authentication

PDCs are issued by a Certification Authority (CA) and signed with the CA's private key. The CA validates the identity of a requesting body before issuing a certificate. Thus the presence of a PDC is a powerful authentication mechanism.

PDCs contain the owner's public key, the owner's name, an expiration date, the name of the Certification Authority that issued the Digital Certificate, a serial number, and maybe some other information.

Users can use PDCs and encoded devices such as Smart Cards and Java Cards for authentication in the Portal Server. The encoded devices carry an electronic equivalent of a PDC stored on the card. If a user logs in using one of these mechanisms, no Log in screen displays and no authentication screen is displayed.

The PDC authentication process involves several steps:

1. From a browser, the user types a connection request, say `https://my.sesta.com`.

   The response to this request depends on whether the Gateway to `my.sesta.com` has been configured to accept certificates.

Note – When a Gateway is configured to accept certificates, it accepts only logins with certificates, not any other kind of login.

The Gateway checks that the certificate has been issued by a known Certificate Authority, has not expired, and has not been tampered with. If the certificate is valid, the Gateway lets the user proceed to the next step in the authentication process.

2. The Gateway passes the certificate to the PDC authentication module in the server.

▼ To Configure PDCs and Encoded Devices

1 **Add the following line in the** `/etc/opt/SUNWam/config/AMConfig.properties` **file on the Portal Server machine:** `com.iplanet.authentication.modules.cert.gwAuthEnable=yes.`

2 **Import the Required Certificates into the certificate database of the Gateway that you want PDC-enabled. To configure the certificates, see "To import the Root CA certificate on the gateway machine" on page 179**

3 **Log into the Access Manager administration console as administrator, do the following:**

   a. **Select the Identity Management tab and then select an Organization.**

   b. **Click Services for the Organization from the View drop down menu.**

   c. **Click Add to register the certificate.**

4 **From the Access Manager administration console, do the following:**

   a. **Select the required organization and click the arrow next to Certificate.**

   b. **In the Trusted Remote Host list box, highlight none and click Remove.**

   c. **Enter any in the text field and click Add.**

   d. **Click Save.**

5 **From the Access Manager administration console, do the following:**

   a. **Choose the required organization and then select Services from the View drop-down menu.**

   The list of services is displayed.

b. **Click the arrow next to the Authentication Configuration core service and then click New.**

   The New Service Instance page is displayed.

c. **Enter the service instance name as** `gatewaypdc`**.**

d. **Click Submit.**

   The gatewaypdc Service Instance List is displayed.

e. **Click gatewaypdc to edit the service.**

   The gatewaypdc show properties page is displayed.

f. **Click Edit link next to Authentication Configuration and then click Add.**

   The Add Module page is displayed.

g. **Choose Cert from the Module Name field and REQUIRED for Enforcement criteria, and then click OK.**

h. **Click OK to complete.**

6 **From the Access Manager administration console, do the following:**

a. **Click the arrow next to Core.**

b. **In the Organization Authentication modules list box, select gatewaypdc.**

c. **Choose Dynamic from the User Profile drop-down menu.**

d. **Click Save to complete.**

7 **Log into the Portal Server administration console as administrator and do the following:**

a. **Select the Secure Remote Access tab and select the appropriate gateway profile.**

b. **Select the Security tab.**

c. **In the Certificate-enabled Gateway hosts list box, add the Gateway name.**

d. **Click Save.**

8 **Restart the gateway profile from a terminal window:**

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

9 **Install the client certificate issued from CA into the browser one has to access PDC enabled gateway.**

10 **Install the client certificate into the JVM keystore. JVM control panel can be accessed as below from the windows machine Start > Setting > Control Panel > Java.**

Add the following to the Applet RunTime parameters:

- `Djavax.net.ssl.keyStore=Path to Keystore`
- `Djavax.net.ssl.keyStorePassword=password`
- `Djavax.net.ssl.keyStoreType=type`

11 **Access your gateway profile and organization:**

https://gateway:instance-port/YourOrganization

You should be logged in without any prompt for Username and Password with the name of the certificate.

▼ To import the Root CA certificate on the gateway machine

1 **Import the Root CA certificate on the gateway machine.**

a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`

Certadmin menu is listed.

b. **Select option 3. Enter the path for the certificates.**

For more information, see the Chapter 10, "Working with Certificates."

2 **Generate a Certificate Signing Request for submitting to the CA.**

a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`

Certadmin menu is listed.

b. **Select option 2. Enter appropriate information.**

c. **Save the file.**

3 **Submit the Certificate Signing Request to a CA and get it approved. Save the certificate response after CA signing.**

4 **Import the Server Certificate after getting approved by CA.**

a. `<Gateway-Install-Dir>/SUNWportal/bin/certadmin -n <gw-profile-name>`

Certadmin menu is listed.

b. **Select option 4.**

c. **Specify the location of the file containing the Server Certificate.**

5 **Import the Root CA certificate on the Portal Server machine.**

Configuring Gateway Attributes Using the Command Line Options

This section provides the command line options to configure Gateway attributes from the terminal window for the following tasks:

- "To Manage Storage of External Server Cookies" on page 180
- "To Enable Marking Cookies as Secure" on page 181
- "To Create List of URLs for Proxies Not to be Used" on page 182
- "To Manage RuleSet to URI Mapping" on page 183
- "To Specify the Default Domain" on page 184
- "To Manage MIME Guessing" on page 184
- "To Create a List of URI Mappings to Parse" on page 185
- "To Manage Masking" on page 186
- "To Specify the masking Seed String" on page 186
- "To Create a List of URIs Not to Mask" on page 187
- "To Make a Gateway Protocol the Same as the Original URI Protocol" on page 188

▼ To Manage Storage of External Server Cookies

When the Store External Server Cookies option is enabled, Gateway stores and manages cookies for any third party application or server that is accessed through the Gateway. Although the application or server cannot service cookieless devices or depends on cookies for state management, Gateway transparently masks the application or server from knowing that the Gateway is servicing a cookieless device.

For information on cookieless devices and client detection, see the *Access Manager Customization and API Guide.*

- **Type the following command and press Enter to manage storage of external server cookies.**
  - **To enable:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a CookieManagement true
    ```

  - **To disable:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a CookieManagement false
    ```

  - **To get attribute value:**

    ```
    PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a CookieManagement
    ```

More Information See also

“psadmin set-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference* and “psadmin get-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference*

▼ To Enable Marking Cookies as Secure

When a cookie is marked as secure, the browser treats the cookie with additional security. The implementation of security depends on the browser. The Enable Cookie Management attribute must be enabled for this to work.

- **Type the following command and press Enter to mark cookies as secure.**
  - **To enable:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a MarkCookiesSecure true
    ```

  - **To disable:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a MarkCookiesSecure false
    ```

  - **To get the attribute value:**

    ```
    PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a MarkCookiesSecure
    ```

More Information See also

"psadmin set-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference* and "psadmin get-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference*

▼ To Create List of URLs for Proxies Not to be Used

The Gateway tries to connect directly to the URLs listed in the Do Not Use Webproxy URLs list. A webproxy is not used to connect to these URLs.

● **Type the following command and press Enter to manage URLs for proxies not to be used.**

Note – Separate each URL with a blank space where there are more than one URL.

- **To specify URLs not to be used:**

  ```
  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
  gateway --gateway-profile PROFILE_NAME -a DontUseWebProxyURL -A
  "LIST_OF_URLS"
  ```

- **To add to the existing list of URLs:**

  ```
  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
  gateway --gateway-profile PROFILE_NAME -a DontUseWebProxyURL -A
  "LIST_OF_URLS"
  ```

- **To remove from the existing list of URLs:**

  ```
  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
  gateway --gateway-profile PROFILE_NAME -a DontUseWebProxyURL -E
  "LIST_OF_URLS"
  ```

- **To get the existing list of URLs:**

  ```
  PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f PASSWORD_FILE -m
  gateway --gateway-profile PROFILE_NAME -a DontUseWebProxyURL
  ```

More Information See also

"psadmin set-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference* and "psadmin get-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference*

▼ To Manage RuleSet to URI Mapping

Secure Remote Access supports Microsoft Exchange 2000 SP3 installation and MS Exchange 2003 of Outlook Web Access (OWA).

1 **To add a URI to the existing list:**

```
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 **To remove a URI from the existing list:**

```
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 **To get the existing list:**

```
PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f PASSWORD_FILE -m
gateway --gateway-profile PROFILE_NAME -a DomainsAndRulesets
```

4 **Type the following command and press Enter to manage RuleSet for Outlook Web Access.**

- **To add a RuleSet**

  ```
  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"
  ```

- **To remove a RuleSet:**

  ```
  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"
  ```

- **To set a list of URIs to RuleSet mappings:**

  ```
  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"
  ```

More Information

See also

“psadmin set-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference* and “psadmin get-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference*

▼ To Specify the Default Domain

The default domains are useful when URLs contain only the host names without the domain and subdomain. In this case, the Gateway assumes that the host names are in the default domain list, and proceeds accordingly.

For example, if the host name in the URL is `host1`, and the default domain and subdomain are specified as `red.sesta.com`, the host name is resolved as `host1.red.sesta.com`.

- **Type the following command and press Enter to specify the default domains.**

  - **To set default domain:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a DefaultDomainsAndSubdomains
    "DOMAIN_NAME"
    ```

  - **To get the default domain:**

    ```
    PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a DefaultDomainsAndSubdomains
    ```

More Information See also

“psadmin set-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference* and “psadmin get-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference*

▼ To Manage MIME Guessing

Rewriter depends on the MIME type of the page to choose the parser. Some web servers such as WebLogic and Oracle do not send MIME types. To work around this, you can enable the MIME guessing feature by adding data to the Map Parser to URIs list box.

- **Type the following command and press Enter to manage MIME guessing.**

  - **To enable MIME guessing:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a EnableMIMEGuessing true
    ```

  - **To disable MIME guessing:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a EnableMIMEGuessing false
    ```

- **To get value:**

  PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a EnableMIMEGuessing

More Information See also

“psadmin set-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference* and “psadmin get-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference*

▼ To Create a List of URI Mappings to Parse

If the MIME Guessing checkbox is enabled and the server has not sent a MIME type, use this list box to map the parser to the URI.

Multiple URIs are separated by a semicolon.

For example HTML=*.html; *.htm;*Servlet. This means that the HTML Rewriter is used to rewrite the content for any page with a html, htm, or Servlet extension.

● **Type the following command and press Enter to create a list of URI mappings to parse.**

- **To set a list of URI mappings to parse:**

  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MIMEMap

- **To add to the existing list:**

  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MIMEMap -A *LIST*

- **To remove from the existing list:**

  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME* -a MIMEMap -E *LIST*

- **To get the existing list:**

  PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f *PASSWORD_FILE* -m gateway --gateway-profile *PROFILE_NAME*-a MIMEMap

More Information See also

“psadmin set-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference*

To Manage Masking

Masking allows Rewriter to rewrite a URI so that the intranet URL of a page is not seen.

- **Type the following command and press Enter to manage masking.**

  - **To enable masking:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a EnableObfuscation true
    ```

  - **To disable masking:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a EnableObfuscation false
    ```

  - **To get value:**

    ```
    PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a EnableObfuscation
    ```

More Information

See also

“psadmin set-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference* and “psadmin get-attribute” in *Sun Java System Portal Server 7.2 Command-Line Reference*

To Specify the masking Seed String

A seed string is used for masking a URI. A masking algorithm generates the string.

Note – Book marking of an masked URI may not work if this seed string has been changed or if the Gateway is restarted.

- **Type the following command and press Enter to specify the masking seed string.**

  - **To set the masking seed string:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a ObfuscationSecretKey
    SECRET_KEY
    ```

  - **To get the value:**

    ```
    PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a ObfuscationSecretKey
    ```

More Information See also

"psadmin set-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference* and "psadmin get-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference*

▼ To Create a List of URIs Not to Mask

Some applications (such as an applet) require an Internet URI and cannot be masked. To specify those applications, add the URI to the list box.

For example if you added `*/Applet/Param*` to the list box, the URL would not be masked if the content URI `http://abc.com/Applet/Param1.html` is matched in the RuleSet rule.

Note – Separate each URI with a blank space where there are more than one URI.

● **Type the following command and press Enter to create a list of URIs not to mask.**

- **To set a list of URIs not to mask:**

  ```
  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
  gateway --gateway-profile PROFILE_NAME -a NotToObscureURIList
  LIST_OF_URI
  ```

- **To add to the existing list:**

  ```
  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
  gateway --gateway-profile PROFILE_NAME -a NotToObscureURIList -A
  LIST_OF_URI
  ```

- **To remove from the existing list:**

  ```
  PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
  gateway --gateway-profile PROFILE_NAME -a NotToObscureURIList -E
  LIST_OF_URI
  ```

- **To get the existing values:**

  ```
  PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f PASSWORD_FILE -m
  gateway --gateway-profile PROFILE_NAME -a NotToObscureURIList
  ```

More Information See also

"psadmin set-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference* and "psadmin get-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference*

▼ To Make a Gateway Protocol the Same as the Original URI Protocol

When a Gateway runs in both HTTP and HTTPS mode, you can enable Rewriter to use a consistent protocol to access the referred resources in the HTML content.

For example, if the original URL is `http://intranet.com/Public.html` then the http Gateway is added. If the original URL is `https://intranet.com/Public.html` then the https Gateway is added.

Note – This applies only to static URIs and not to dynamic URIs generated in Javascript.

- **Type the following command and press Enter to make a Gateway protocol the same as the original URI protocol.**

  - **To enable:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a UseConsistentProtocolForGateway
    true
    ```

  - **To disable:**

    ```
    PS_INSTALL_DIR/bin/psadmin set-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a UseConsistentProtocolForGateway
    false
    ```

  - **To get the value:**

    ```
    PS_INSTALL_DIR/bin/psadmin get-attribute -u amadmin -f PASSWORD_FILE -m
    gateway --gateway-profile PROFILE_NAME -a UseConsistentProtocolForGateway
    ```

More Information

See also

"psadmin set-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference* and "psadmin get-attribute" in *Sun Java System Portal Server 7.2 Command-Line Reference*

CHAPTER 9

Configuring Rewriter in the Gateway Service

highlights content here.

This chapter has the following sections:

For more information on rewriter rules, see "Defining Language Based Rules" on page 70

For more information on Rewriter problems, see "Troubleshooting Using Debug Logs" on page 94.

For Rewriter examples, see "Working Samples" on page 97.

Creating a List of URIs to RuleSet Mappings

After the ruleset is created, associate a domain with the ruleset using the Map URIs to RuleSets field. The following two entries are added by default to the Map URIs to RuleSets field:

- `*://*.Sun.COM/portal/*|default_gateway_ruleset`

  where `sun.com` is the install domain of the portal and /portal is the portal install context
- `*|generic_ruleset`

This means that for all pages from portal directory with the domain `sun.com`, the `default_gateway_ruleset` is applied. For all other pages, the generic ruleset is applied. The `default_gateway_ruleset` and the `generic_ruleset` are pre-packaged rulesets.

Note – For all the content appearing on the standard Portal Desktop, the ruleset for the `default_gateway_ruleset` is used, irrespective of where the content is fetched from.

For example, assume that the standard Portal Desktop is configured to scrape the content from the URL `yahoo.com`. The Portal Server is in `sesta.com`. The ruleset for `sesta.com` is applied to the fetched content.

Note – The domain for which you specify a ruleset must be listed in the Proxies for Domains and Subdomains list.

Using Wildcards Within the Syntax

You can map a fully qualified URI or a partial URI by using an asterisk in the ruleset.

For example, you could apply the `java_index_page_ruleset` to an `index.html` page as follows:

```
www.sun.com/java/index.html/java_index_page_ruleset
```

or you could apply all pages in the java directory to the `java_directory_ruleset`, as follows:

```
www.sun.com/java/* /java_directory_ruleset
```

Configuring Rewriter in the Gateway Service

Using the Gateway service, under the Rewriter tab, you can perform the following tasks within two categories, Basic and Advanced:

Basic Tasks

- "To Enable the Gateway to Rewrite All URLs" on page 190
- "To Specify the URIs Not to Rewrite" on page 191
- "To Map a URI to a RuleSet" on page 191
- "To Specify MIME Mappings" on page 192
- "To Specify the Default Domains" on page 193

▼ To Enable the Gateway to Rewrite All URLs

If you enable the Enable Rewriting of All URIs option in the Gateway service, Rewriter rewrites any URL without checking against the entries in the Proxies for Domains and Subdomains list. Entries in the Proxies for Domains and Subdomains list are ignored.

1 **Log into the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab, and select the gateway profile for which you want to modify the attributes.**

3 **Select the Rewriter tab.**

4 **Under Basic Options, select the Enable Rewriting of All URIs checkbox to enable the Gateway to rewrite all URLs.**

5 **Click Save to complete.**

6 **Restart the Gateway from a terminal window:**

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

▼ To Specify the URIs Not to Rewrite

1 **Log into the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab, and select the gateway profile for which you want to set the attribute.**

3 **Select the Rewriter tab.**

4 **Under Basic Option, enter the URI in the Add text field and then click Add.**

The URI values is displayed in the URIs Not To Rewrite box.

Note – Adding #* to this list allows URIs to be rewritten, even when the href rule is part of the ruleset.

5 **Click Save to complete.**

6 **Restart the Gateway from a terminal window:**

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

▼ To Map a URI to a RuleSet

1 **Log into the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab, and select the gateway profile for which you want to set the attribute.**

3 **Select the Rewriter tab.**

4 **Under Rewriter Options, click Map URI to Rulesets, and click Add Row.**

5 **Enter the required domain or host name in the URI field and the enter appropriate ruleset for the domain in the Rule Set field.**

The entry is added to the Map URIs to RuleSets list. The format for specifying the domain or host name and the ruleset is as follows:

```
domain name|ruleset name
```

For example:

```
eng.sesta.com|default
```

6 **Click Save to Complete.**

7 **Restart the Gateway from a terminal window:**

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

▼ To Specify MIME Mappings

Rewriter has four different parsers to parse the web pages based on the content type: HTML, JAVASCRIPT, CSS and XML. Common MIME types are associated with these parsers by default. You can associate new MIME types with these parsers in the Map Parser to MIME Types field of the Gateway service. This extends the Rewriter functionality to other MIME types.

Separate multiple entries with a semicolon or a comma (";" or ",".) For example:

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

means any content with these MIMEs are sent to the HTML Rewriter and HTML rules would be applied to rewrite the URLs.

Tip – Removing unnecessary parsers from the MIME mappings list can increase the speed of operation. For example, if you are sure that the content from a particular intranet will not have any JavaScript, you can remove the JAVASCRIPT entry from the MIME mappings list.

---

1 **Log into the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab, and select the gateway profile for which you want to set the attribute.**

3 **Select the Rewriter tab.**

4 **Under Rewriter Option, click Map Parser to Map MIME Types .**

Specify the entry in the format `HTML=text/html;text/htm`

5 **Click Add Row to add the entry to the list. Enter the parser value and corresponding MIME value to map to in the MIME Type filed.**

6 **Click Save to complete.**

7 **Restart the Gateway from a terminal window:**

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

▼ To Specify the Default Domains

The default domain and subdomain are useful when URLs contain only the host names without the domain and subdomain. In this case, the Gateway assumes that the host names are in the default domain and subdomain, and proceeds accordingly.

For example, if the host name in the URL is `host1`, and the default domain and subdomain are specified as `red.sesta.com`, the host name is resolved as `host1.red.sesta.com`.

1 **Log into the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab, and select the gateway profile for which you want to set the attribute.**

3 **Select Deployment Tab.**

4 **In the Proxies for Domains and Subdomains field, type the required domain name with out proxy.**

5 **Click Save to complete.**

6 **Restart the Gateway from a terminal window:**

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

CHAPTER 10

Working with Certificates

This chapter describes certificate management and explains how to install self-signed certificates and certificates from a Certificate Authority.

This chapter explains the following topics:

Introduction to SSL Certificates

The Sun Java System Portal Server Secure Remote Access software provides certificate-based authentication for remote users. SRA uses Secure Sockets Layer (SSL) to enable secure communication. The SSL protocol enables secure communication between two machines.

A SSL certificate provides encryption and decryption capabilities using a public and private key pair.

The two types of certificates are:

- Self-signed certificates (also called root CA certificate)

- Certificates issued by Certificate Authority (CA)

By default, a self-signed certificate is generated and installed when you install the Gateway.

You can generate, obtain, or replace a certificate anytime after installation.

SRA also supports client authentication with Personal Digital Certificates (PDCs). PDCs are a mechanism to authenticate a user through SSL client authentication. With SSL client authentication, the SSL handshake ends at the Gateway. The Gateway extracts the user's PDC and passes it to the authenticated server. This server uses the PDC to authenticate the user. To configure PDCs along with Authentication Chaining, see "Using Authentication Chaining" on page 56.

SRA provides a tool named `certadmin` that you can use to manage the SSL certificates. See "The certadmin Script" on page 201.

Note – Certificate pop up windows are common in SSL applications. Advise users to accept the warning and proceed.

Certificate Files

Certificate related files are located in `/etc/opt/SUNWportal/cert/`*gateway-profile-name*. This directory contains 5 files by default.

"Certificate Files" on page 196 lists these files and their descriptions.

TABLE 10–1 Certificate Files

File Name	Type	Description
`cert8.db`, `key3.db`, `secmod.db`	Binary	Contains the data for certificates, keys, and cryptographic modules. Can be manipulated using the `certadmin` script. If necessary, these files can be shared between the Portal Server host and gateway components or the Gateway.
`.jsspass`	hidden text file	Contains the encrypted password for the SRA key database.

TABLE 10–1 Certificate Files *(Continued)*

File Name	Type	Description
`.nickname`	hidden text file	Stores the names of the token and certificate that the Gateway needs to use in the format *token-name:certificate-name*. If you are using the default token (the token on the default internal software encryption module), omit the token name. In most cases, the `.nickname` file stores only the certificate name. As an administrator, you can modify the certificate name in this file. The certificate that you specify is now used by the Gateway.

Certificate Trust Attributes

The trust attributes of a certificate indicate the following information:

- Whether the certificate (in the case of client or server certificate) was issued by a Trusted CA.
- Whether the certificate (in the case of a root certificate) can be trusted as the issuer of a server or client certificate.

The three available trust categories for each certificate are expressed in this order: “SSL, email, object signing”. Only the first category is useful for the Gateway. In each category position, zero or more trust attribute codes are used.

The attribute codes for the categories are separated by commas, and the entire set of attributes is enclosed by quotation marks. For example, the self-signed certificate generated and installed during the Gateway installation is marked "u,u,u" which means the certificate is a server certificate (user certificate) and not a root CA certificate.

“Certificate Trust Attributes” on page 197 lists the possible attribute values and the meaning of each value.

TABLE 10–2 Certificate Trust Attributes

Attribute	Description
p	Valid peer
P	Trusted peer (implies p)
c	Valid CA
T	Trusted CA to issue client certificates (implies c)

TABLE 10–2 Certificate Trust Attributes *(Continued)*

Attribute	Description
C	Trusted CA to issue server certificates (SSL only) (implies c)
u	Certificate can be used for authentication or signing
w	Send warning (use with other attributes to include a warning when the certificate is used in that context)

CA Trust Attributes

Most well-known public CAs are included in the certificate database. See "Modifying the Trust Attributes of a Certificate" on page 207 for information on modifying the trust attributes of a public CA.

"CA Trust Attributes" on page 198 lists the most common Certificate Authorities with the trust attributes.

TABLE 10–3 Public Certificate Authorities

Certificate Authority Name	Trust Attribute
Verisign/RSA Secure Server CA	CPp,CPp,CPp
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

TABLE 10–3 Public Certificate Authorities *(Continued)*

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
TC 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
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

TABLE 10–3 Public Certificate Authorities *(Continued)*

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
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

TABLE 10–3 Public Certificate Authorities *(Continued)*

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
E-Certify RA	CPp,CPp,CPp
Entrust.net Global Secure Server CA	CPp,CPp,CPp
Entrust.net Global Secure Personal CA	CPp,CPp,CPp

The certadmin Script

You can use the `certadmin` script to do the following certificate administration tasks:

- "Generating Self-Signed Certificates" on page 202
- "Generating a Certificate Signing Request (CSR)" on page 203
- "Adding a Root CA Certificate" on page 204
- "Installing a Certificate from a CA" on page 206
- "Deleting a Certificate" on page 207
- "Modifying the Trust Attributes of a Certificate" on page 207
- "Listing Root CA Certificates" on page 208
- "Listing All Certificates" on page 209
- "Printing a Certificate" on page 210

Generating Self-Signed Certificates

You need to generate certificates for SSL communication between each server and Gateway.

▼ To Generate a Self-Signed Certificate After Installation

1 **As root, run the `certadmin` script on the Gateway machine for which you want to generate a certificate:**

portal-server-install-root`/SUNWportal/bin/certadmin -n` *gateway-profile-name*

The certificate administration menu is displayed.

```
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 **Choose option 1 on the certificate administration menu.**

The certificate administration script asks you if you want to keep the existing database files.

3 **Enter organization-specific information, token name, and the certificate name.**

Note – For a wild card certificate, specify a * in the fully-qualified DNS name of the host. For example, if the fully-qualified DNS name of the host is `abc.sesta.com`, specify it as `*.sesta.com`. The certificate that is generated is now valid for all host names in the `sesta.com` domain.

```
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.
```

The token name (default being empty) and certificate name are stored in the `.nickname` file under `/etc/opt/SUNWportal/cert/`*gateway-profile-name*.

4 **Restart the Gateway for the certificate to take effect:**

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

Generating a Certificate Signing Request (CSR)

Before you can order a certificate from a CA, you need to generate a certificate signing request which contains the information that is required by the CA.

▼ To Generate a CSR

1 **As root, run the `certadmin` script:**

```
portal-server-install-root/SUNWportal/bin/certadmin -n gateway-profile-name
```

The certificate administration menu is displayed.

```
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 **Choose option 2 on the certificate administration menu.**

The script prompts you for organization-specific information, token name, and web master's email and phone number.

Ensure that you specify the fully-qualified DNS name of the host.

```
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 **Type all the required information.**

Note – Do not leave the web master's email and phone number blank. The information is necessary for obtaining a valid CSR.

A CSR is generated and stored in the file *portal-server-install-root*/`SUNWportal/bin/csr.hostname.datetimestamp`. The CSR is also printed on the screen. You can directly copy and paste the CSR when you order a certificate from a CA.

Adding a Root CA Certificate

If a client site presents a certificate signed by a CA that is unknown to the Gateway certificate database, the SSL handshake fails.

To prevent this, you need to add a root CA certificate to the certificate database. This ensures that the CA becomes known to the Gateway.

Browse to the CA's website and obtain the root certificate for that CA. When you use the `certadmin` script, specify the file name and path of the root CA certificate.

▼ To Add a Root CA Certificate

1 **As root, run the `certadmin` script.**

portal-server-install-root/`SUNWportal/bin/certadmin -n` *gateway-profile-name*

The certificate administration menu is displayed.

```
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 **Choose option 3 on the certificate administration menu.**

3 **Enter the name of the file that contains the root certificate and enter the name of the certificate.**

The root CA certificate is added to the certificate database.

Installing SSL Certificates From the Certificate Authority

During the installation of the Gateway, a self-signed certificate is created and installed by default. At any point after installation, you can install SSL certificates signed by vendors who provide official certificate authority (CA) services, or by your corporate CA.

The three steps involved in this task are:

- "Generating a Certificate Signing Request (CSR)" on page 203
- "Ordering a Certificate from a CA" on page 205
- "Installing a Certificate from a CA" on page 206

Ordering a Certificate from a CA

After generating a certificate signing request (CSR), you need to order the certificate from the CA using a CSR.

To Order a Certificate From a CA

1 **Go to the Certificate Authority's web site and order your certificate.**

2 **Provide the CSR as requested by the CA. Provide other information if requested by the CA.**

You will receive your certificate from the CA. Save it in a file. Include the "BEGIN CERTIFICATE" and "END CERTIFICATE" lines with the certificate in the file.

The following example omits the actual certificate data.

```
-----BEGIN CERTIFICATE-----
The certificate contents...
----END CERTIFICATE-----
```

Installing a Certificate from a CA

Using the `certadmin` script, install the certificate obtained from the CA in your local database files in `/etc/opt/SUNWportal/cert/`*gateway-profile-name*.

▼ To Install a Certificate From a CA

1 **As root, run the `certadmin` script.**

portal-server-install-root`/SUNWportal/bin/certadmin -n` *gateway-profile-name*

The certificate administration menu is displayed.

```
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 **Choose option 4 on the certificate administration menu.**

The script asks you to enter the certificate file name, certificate name, and the token name.

```
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 **Supply all the required information.**

The certificate is installed in `/etc/opt/SUNWportal/cert/`*gateway-profile-name*, and the screen prompt returns.

4 **Restart the Gateway for the certificate to take effect:**

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

Deleting a Certificate

You can delete a certificate by using the certificate administration script.

▼ To Delete a Certificate

1 **As root, run the `certadmin` script.**

portal-server-install-root`/SUNWportal/bin/certadmin -n`

where *gateway-profile-name* is the name of the Gateway instance.

The certificate administration menu is displayed.

```
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 **Choose option 5 on the certificate administration menu.**

3 **Enter the name of the certificate to be deleted.**

Modifying the Trust Attributes of a Certificate

One case in which the trust attributes of a certificate needs to be modified is if client authentication is used with the Gateway. An example of client authentication is PDC (Personal Digital Certificate). The CA that issues the PDCs must be trusted by the Gateway, and the CA certificate must be marked "T" for SSL.

If the Gateway is set up to communicate with an HTTPS site, the CA of the HTTPS site server certificate must be trusted by the Gateway, and the CA certificate must be marked "C" for SSL.

▼ To Modify the Trust Attributes for a Certificate

1 **As root, run the `certadmin` script.**

gateway-install-root`/SUNWportal/bin/certadmin -n`
gateway-profile-name

where *gateway-profile-name* is the name of the Gateway instance.

The certificate administration menu is displayed.

```
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 **Choose option 6 on the certificate administration menu.**

3 **Enter the name of the certificate. For example, Thawte Personal Freemail CA.**

```
Please enter the name of the certificate?
Thawte Personal Freemail CA
```

4 **Enter the trust attribute for the certificate.**

```
Please enter the trust attribute you want the
certificate to have [CT,CT,CT]
```

The certificate trust attribute will be changed.

Listing Root CA Certificates

You can view all root CA certificates by using the certificate administration script.

To View the List of Root CAs

1 **As root, run the `certadmin` script.**

portal-server-install-root`/SUNWportal/bin/certadmin -n`
gateway-profile-name

where *gateway-profile-name* is the name of the Gateway instance.

The certificate administration menu is displayed.

```
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 **Choose option 7 on the certificate administration menu.**

All root CA certificates are displayed.

Listing All Certificates

You can view all certificates and their corresponding trust attributes by using the certificate administration script.

To List All the Certificates

1 **As root, run the `certadmin` script.**

```
portal-server-install-root
/SUNWportal/bin/certadmin -n
gateway-profile-name
```

where *gateway-profile-name* is the name of the Gateway instance.

The certificate administration menu is displayed.

```
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 **Choose option 8 on the certificate administration menu.**

All CA certificates are displayed.

Printing a Certificate

You can print a certificate by using the certificate administration script.

▼ To Print a Certificates

1 **As root, run the `certadmin` script.**

```
portal-server-install-root/SUNWportal/bin/certadmin -n
  gateway-profile-name
```

where *gateway-profile-name* is the name of the Gateway instance.

The certificate administration menu is displayed.

```
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 **Choose option 9 on the certificate administration menu.**

3 **Enter the name of the certificate.**

CHAPTER 11

Configuring the Netlet

This chapter describes configuring the Netlet attributes from the Sun Java System Portal Server administration console. All the attributes that can be configured at the organization level can also be configured at the user level. For more information on organization, role and user level attributes, see the *Access Manager Administration Guide*.

This chapter has the following sections:

- "Configuring the Netlet Attributes" on page 211
- "Proxy Configuration for Netlet" on page 215

Configuring the Netlet Attributes

You can perform the following tasks to configure the Netlet:

- "To Configure the Basic Attributes" on page 211
- "Configuring the Advanced Attributes" on page 212
- "To Create, Modify, or Delete a Netlet Rule" on page 214

▼ To Configure the Basic Attributes

1 **Log onto the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab and select the Netlet tab.**

3 **Select a DN for a user or an organization from Select DN list or add a DN.**

4 **Modify the following attributes:**

Attribute Name	Description
COS Priority	Specify value that is used to determine the inheritance of the attribute values. For more information on this attribute, see the *Sun Java System Directory Server Administration Guide*.
Launch Netlet Using	Select the mode either the Java Webstart or Applet option to start the Netlet service.
Default Loopback Port	Specify the port to be used on the local machine when applets are downloaded through Netlet. The default value of 58000 is used unless the value is overridden in the Netlet rules. Enter the required port number.
Keep Alive Interval (seconds)	If the client is connecting to the Gateway through a web proxy, then idle Netlet connections are disconnected due to proxy time out. To prevent this, enter a value less than the proxy time-out.

5. **Click Save to complete.**

▼ Configuring the Advanced Attributes

1. **Log onto the Portal Server administration console as administrator.**
2. **Select the Secure Remote Access tab and select the Netlet tab.**
3. **Select a DN for a user or an organization from Select DN list or add a DN.**
4. **Modify the following attributes:**

Attribute Name	Description
Terminate Netlet at Portal Logout	Select Yes to ensure that all connections are terminated when a user logs out of the Portal Server. This ensures greater security. By default, this option is selected. Select No to ensure that live Netlet connections are operational even after the user has logged out of the Portal Server desktop. **Note –** When the No option is selected, users are not allowed to make new Netlet connections after logging out of the Portal Server. Only existing connections are preserved.
Re-authenticate for Connections	Select Yes to specify the port to be used on the local machine when applets are downloaded through Netlet. The default value of 58000 unless the value is overridden in the Netlet rules. By default, the No option is selected.

Attribute Name	Description
Display Warning Popup for Connections	Select Yes to display a warning popup dialog box on the user's desktop when other users are trying to connect to Netlet through the listen port and the user is running an application using Netlet. By default, the Yes option is selected.
Display Checkbox in Port Warning Dialog	Select Yes to display a warning popup dialog box on the users desktop when Netlet tries to connect to the destination host through an available port on the local machine, if its enabled in the administration console. By default, the Yes option is selected.
Netlet Rules	Create Netlet rules at a global level. These rules are inherited by any new organization that you create. For more information on creating, modifying, and deleting Netlet rules, see "To Create, Modify, or Delete a Netlet Rule" on page 214
Default Native VM Cipher	Select from the drop down box the default cipher for the Netlet rules. This is useful when using existing rules that did not include the cipher as a part of the rule. For more information, see the "Backward Compatibility" on page 143 section.
Default Java Plugin Cipher	Select from the drop down box the default Java Plugin cipher. See "Supported Ciphers" on page 142 for a list of supported ciphers.
Allowed/Denied Hosts	Select the host address check box and select host to either allow access based on the user or organization type and select either the Allow or Deny option from the drop-down box. To add a new host: a. Click Add Row. b. Enter the specify the fully qualified host address, for example: `abc`, type `abc.sesta.com`. **Note –** To delete an existing host: From the Host list, select the host and click Delete. You can define access or deny to certain hosts to specific hosts for certain organizations, roles, or users. For example, you can set up the Allow list with five hosts to which the user can telnet. You can deny access to specific hosts within an organization. Specify a unique `local port for` each rule. **Note –** An asterisk (*) in this field indicates that all the hosts in the specified domain are accessible. For example, if you specify `*.sesta.com`, all the Netlet targets within the `sesta.com` domain can be executed by the user. You can also specify a wild card IP address such as `xxx.xxx.xxx.*`.
Access/Deny Netlet Rules	Select the Nelet rule and select either the Allow or Deny option from the drop-down box. You can define access to specific Netlet rules for certain organizations, roles or users. You can deny access to specific Netlet rules for certain organizations, roles or users. **Note –** An asterisk (*) in this field indicates that all the defined Netlet rules are available for the selected organization.

5 Click Save to complete.

▼ To Create, Modify, or Delete a Netlet Rule

You can also create new rules or modify existing rules at the organization, role, or user levels. These rules are inherited by any new organization that you create.

1. **Log onto the Portal Server administration console as administrator.**

2. **Select the Secure Remote Access tab and select the Netlet tab.**

3. **Select a DN for a user or an organization from Select DN list or add a DN.**

4. **Under Advanced > Netlet Rules, click New Rule.**

   - **To delete a rule, select a rule and click Delete.**

   - **To modify a rule, click the rule name.**

     In the Netlet page, modify the parameters as explained the steps below.

5. **Enter the rule name in the Rule Name field.**

6. **Select Other choose from the list of available ciphers and under Encryption Ciphers list, select one or more encryption cipher or select Default to retain the default encryption cipher.**

   This is useful when using existing rules that did not include the cipher as a part of the rule. For information, see the Backward Compatibility section. For more information on ciphers, see Specify the Default Encryption Cipher.

7. **Enter the URL to the application to be invoked in the Remote Application URL field.**

8. **Select the Client Port checkbox if an applet needs to be downloaded. Enter client port number, server host address, and server port number in the Client Port, Server Host, and Server Port field. Specify a unique `local port for` each rule.**

   By default, the Enable Download Applet box is disabled. Specify the applet details only if the applet needs to be downloaded from a host other than the Portal Server host. For more information, see “Downloading an Applet From a Remote Host” on page 136.

9. **Select the Enable Extend Session checkbox to ensure that the Portal Server session time is extended while the Netlet session corresponding to this rule is running.**

10. **Under Map Local Port to Destination Server Port, do the following:**

    a. **Enter the local port on which Netlet listens in the Local Port field.**

       For an FTP rule, the local port value must be 30021.

b. **Enter an entry in the Destination Hosts field.**

For a static rule, enter the host name of the target machine for the Netlet connection. For a dynamic rule, enter "TARGET".

c. **Enter the port on the target host in the Destination Port field.**

11 **Click Save to complete.**

The rule name is displayed in the Netlet home page.

Proxy Configuration for Netlet

The following attributes can be configured at the user level:

- Browser proxy type
- Browser proxy host
- Browser proxy port
- Browser proxy override list

If you do not specify these values in the administration console and Netlet is unable to determine the browser proxy setting, the user is asked for this information when a connection is being established through Netlet for the first time. This information is stored and used for future connections by the user.

Netlet fails to determine the browser proxy setting in the following scenarios:

- The user has Internet Explorer 4.x, 5.x or 6.x with Java plug-in (version less then 1.4.0), has enabled the "Use Browser Settings" option in the Proxies tab of the Java Plug-in Control Panel, and has specified an add-on product or INS file in the "Use automatic configuration script" field in the Local Area Network Settings dialog of Internet Explorer.
- The user has Netscape 6.2 with Java Plug-in (version 1.3.1_01 or greater) and has enabled the "Use Browser Settings" option in the Proxies tab of the Java Plug-in Control Panel.

In both these cases, Netlet may not be able to determine the browser settings, and hence the user is asked to supply the following information:

- Browser proxy type

  This attribute can take the values DIRECT or MANUAL. If the user chooses DIRECT from the drop-down list, Netlet connects directly to the gateway host.
- Browser proxy host

  Specify the required proxy host through which Netlet needs to connect.
- Browser proxy port

  Specify the port on the proxy host through which Netlet needs to connect.
- Browser proxy override list (Comma separated)

Specify the hosts for which you do not want Netlet to connect through the proxy. This list can contain multiple comma-separated host names.

CHAPTER 12

Configuring Netlet With Private Domain Certificates

This chapter describes configuring the client browser's Java Plug–in, so that Netlet can be used with PDC.

Note – Only Virtual Machines (VMs) with JSSE support Netlet with PDC.

Configuring Netlet for PDC

Intro text should be here.

To Configure Netlet for PDC

1 **Add `com.iplanet.authentication.modules.cert.gwAuthEnable=yes` anywhere in** `/ect/opt/SUNWam/config/AMConfig.properties` **file on the Portal Server machine.**

2 **Import the Required Certificates into the certificate database of the Gateway to be PDC enabled.**

3 **Import the Root CA certificate on the gateway machine.**

4 **Add the CA certificate to your gateway profile.**

Tip – Create your own gateway profile to test PDC.

Perform the following to steps to add the certificate to your gateway profile.

a. *Gateway Install Directory*`/SUNWportal/bin/certadmin -n gateway profile name`

Certadmin menu will be listed.

b. **Select Option 3.**

c. **Provide the certificate path.**

Certificate added message will display.

5 **Generate a Certificate Signing Request for submitting to the CA.**

Perform the following steps to generate a Certificate Signing Request:

a. *Gateway Install Directory*`/SUNWportal/bin/certadmin -n gateway profile name`

Certadmin menu will be listed.

b. **Select Option 2.**

c. **Provide appropriate answers to the questions.**

d. **Save the request in a file.**

6 **Submit the Certificate Signing Request to a CA and get it approved.**

Tip – Save the certificate signing response after CA signing.

---

7 **Import the CA approved Server Certificate.**

Perform the following steps to import the Server Certificate:

a. *Gateway Install Directory*`/SUNWportal/bin/certadmin -n gateway profile name`

Certadmin menu will be listed.

b. **Select Option 4.**

c. **Provide the location of the file containing the Server Certificate.**

8 **Import the Root CA certificate to the Portal Server machine.**

- **For Application Server use the following command to add** `root-ca`**.**

```
./certutil -A -n rootca -t "TCu,TCu,TCuw" -d
/var/opt/SUNWappserver/domains/domain1/config -a -i path to root-ca
```

CHAPTER 13

Configuring Proxylet

This chapter describes configuring Proxylet from the Sun Java System Portal Server administration console.

This chapter contains the following sections:

- “Configuring the Proxylet Attributes” on page 219
- “Configuring Applications to the Portal Desktop” on page 221
- “Launching Proxylet in Java Web Start or Applet Mode” on page 222

Configuring the Proxylet Attributes

Proxylet can be configured to launch automatically when the user logs in by checking the Download Proxylet Applet Automatically checkbox under Deployment options. When the Download Proxylet Automatically checkbox is not selected, users can get Proxylet on-demand by clicking the Launch the Proxylet link in the Proxylet channel on the standard Portal Desktop.

▼ To Configure the Proxylet Attributes

1 **Log into Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab, and select the Proxylet tab.**

3 **Select an appropriate DN from the Select DN list box or add an existing DN for a specific user or an organization.**

4 **Under the Proxylet page, do the following:**

Attribute Name	Description
COS Priority	Select the class of service for Proxylet traffic from a list of options.
Download Proxylet Applet Automatically	Click Yes to automatically download the Proxylet applet to the client machine. The following are the basic requirements to download the Proxylet applet: Client machine can run a server application Client machine Java version is 1.4 and above The browser is IE 6.0 sp2 or Firefox 2.0 Correct browser permissions
Refresh Portal via Proxylet	Click Yes if you want to refresh your Portal desktop after the Proxylet is launched, and make the traffic to go through the Proxylet. "App Urls" won't be functional if both "Refresh portal after Proxylet Launch" and "Download Proxylet Applet Automatically" are enabled.
Launch Mode	Select Java Web Start or Applet.
Default Proxylet Applet Bind IP	Type the IP address where Proxylet binds and listens for requests from the browser.
Default Proxylet Applet Port	Type the port number where Proxylet listens for requests from a browser.
Automatic Proxy Configuration File Location	Type the location of the configuration file that contains proxy settings from the Proxy Auto Configuration (PAC) file or from the proxy configuration list.

5 **In the Proxylet Rules option, do the following:**

a. **Specify the rules for the application to be launched through the Proxylet service.**

b. **Click Add.**

c. **Enter the domain name, for example, `www.google.com` in the Domain field.**

d. **Enter the host and corresponding port number for domain for Proxylet to process. This ensures that the Proxylet resolves the HTTP requests and the requests are not routed through the gateway.**

6 **Click Save to complete.**

Configuring Applications to the Portal Desktop

Requests such as HTTP, FTP, and so on go through the Proxylet service. Proxylet rules allow the administrator to specify mappings based on protocol, host, or port to domains. With Proxylet rules your can specify the domain and proxy settings in the Proxy Auto Configuration (PAC) file. For example, you can create a rule so that all FTP traffic is routed through Netlet and all HTTP traffic is routed through Proxylet. You can configure predefined applications that need to be rendered through the Proxylet service. This can be done based on the user or organization preferences. Once the applications are added for Proxylet to process, the users desktop is easier to manage and provides better performance.

▼ To Configure an Application to the Portal Desktop

Before You Begin

- Ensure that the Proxylet option is enabled. For more information on enabling Proxylet, see the Gateway Profiles chapter.

1. **Log into Portal Server administration console as administrator.**

2. **Select the Portal tab, and select the portal instance to modify.**

   The Desktop page is displayed.

3. **Select an appropriate DN from the Select DN list box or add an existing DN for a specific user or an organization.**

4. **Click the Manage Containers and Channels link.**

   The Manage Containers and Channels page is displayed.

5. **From the left pane, select Proxylet.**

6. **From the right pane, select the `Appurls` link.**

7. **In the Properties wizard, enter the application name and the value. Modify the properties of the application as required. For example, enter an appropriate name for the application, and http://www.example.com.**

8. **Click Close to complete.**

   The user or at the organization level can now view the application link on the Portal Desktop.

Launching Proxylet in Java Web Start or Applet Mode

You can start Proxylet either in the Java Web Start or Applet mode from the Portal Desktop.

▼ To Launch Proxylet in Java Web Start or Applet Mode

1. **Log onto the Portal Desktop as the proxylet user.**

2. **In the Front Page, go to the Proxylet channel and click the Edit icon.**

3. **From the Launch Mode list box, select the Java Web Start or Applet option.**

4. **Click Finished.**

   To invoke Proxylet, select the application from the Proxylet Channel. This launches the application in the Java Web Start or Applet mode.

   - If the Download Automatically is selected, click the application under the Proxylet channel.
   - Based on the user preferences, the Proxylet console is displayed depending on the selection of Java Web Start or Applet mode. Accept all certificates and continue to work on the application.

CHAPTER 14

Configuring NetFile

This chapter describes configuring NetFile from the Sun Java System Portal Server administration console.

This chapter contains the following section:

- "Configuration Tasks for NetFile" on page 223

Configuration Tasks for NetFile

This section has the following tasks:

- "To Configure the Basic Options" on page 223
- "To Configure the Access Privileges" on page 225
- "To Configure the Host Preferences" on page 225
- "To Configure the Operation Preferences" on page 226
- "To Configure the Operation Preferences" on page 226

▼ To Configure the Basic Options

1. **Log onto the Portal Server administration console as administrator.**

2. **Select the Secure Remote Access tab and select the Netfile tab.**

3. **Select a DN for a user or an organization from Select DN list or add a DN.**

4. **Modify the following attributes:**

Attribute Name	Description
COS Priority	Specify value that is used to determine the inheritance of the attribute values. For more information on this attribute, see the *Sun Java System Directory Server Administration Guide*.
Domain/Host Preferences	Enter the default domain that NetFile requires to contact allowed hosts. This default domain value is applicable only if the user does not specify a fully qualified host name while adding a host using NetFile. **Note –** Ensure that the Default Domain field is not blank, and that it contains a valid domain name.
Default WINS/DNS Server	Enter the WINS/DNS server host address that NetFile uses to access Microsoft Windows hosts. **Note –** A user can override this value by specifying a different value while adding a machine.
Host Detection Order	Use the Move Up and Move Down button to specify the host detention order.
Common Hosts	Enter either the host name or the fully qualified name and click Add. If the host name that you have provided matches the host name configured by the user, the two sets of information are merged and the user-specified values override the values that you specified. Configure a list of hosts to be available through NetFile to all remote NetFile users.

Note – For example, suppose you have configured 4 common hosts - `sesta`, `siroe`, `florizon`, and `abc`. A user configures 3 hosts out of which 2 are `sesta` and `siroe`. User-specified values override administrator-specified values in such conflict situations. `florizon` and `abc` are also listed in the user's NetFile, and the user can carry out various operations on those hosts. In case you have listed `florizon` in the Denied Hosts List, `florizon` is listed in the user's NetFile, but no operation can be carried out on `florizon`.

Host Type—If the user has already added a host that is listed in the Common Hosts list, the user setting takes precedence. If a conflict in the type exists, the shares added by the administrator are not added for that user. If the user and the administrator add the same share, the share is added, but the password set by the user takes precedence.

5 **Click Save to complete.**

▼ To Configure the Access Privileges

1 **Log onto the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab and select the Netfile tab.**

3 **Select a DN for a user or an organization from Select DN list or add a DN.**

4 **Click Access Privilege and modify the following attributes:**

Attribute Name	Description
Access to Windows Hosts	Select the Allow checkbox to ensure that users have access to Windows Hosts. By default, the Allow checkbox is selected.
Access to FTP Hosts	Select the Allow checkbox to ensure that users have access to FTP Hosts.
Access to NFS Hosts	Select the Allow checkbox to ensure that users have access to NFS Hosts.
Access to Netware Hosts	Select the Allow checkbox to ensure that users have access to Netware Hosts.

5 **Click Save to complete.**

▼ To Configure the Host Preferences

1 **Log onto the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab and select the Netfile tab.**

3 **Select a DN for a user or an organization from Select DN list or add a DN.**

4 **By default, users are allowed to access all the hosts through NetFile because of the * entry in the Allow/Deny hosts list. If you want to change that, remove the * entry and specify only those hosts to which users need to have access through NetFile, in this list. Alternatively, you can keep the * entry here, and specify the hosts to which you want to deny access in the Denied Hosts list. In that case, all the hosts except the ones specified in the Denied Hosts list are allowed access.**

Note – If you deny access to a host, and a user has already added this host in the NetFile window, the denied host continues to be displayed in the NetFile window of the user. But the user is not be able to carry out any operations on the host. In NetFile Java2, denied hosts, if displayed in the application, are marked with a red cross to indicate that they are inaccessible. If both the Allowed Hosts and Denied Hosts lists are blank, access is not allowed to any host.

5 **Click Save to complete.**

▼ To Configure the Operation Preferences

1 **Log onto the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab and select the Netfile tab.**

3 **Select a DN for a user or an organization from Select DN list or add a DN.**

4 **Modify the following attributes:**

Attribute Name	Description
Default Compression Type	Select ZIP or GZ from the drop down box as the default file compression format.
Default Compression Level	Select the default compression level from the drop down box. The default is 6.
Temporary Directory Location	Enter the location for the temporary files. The specified temporary directory is created if it does not exist on the server. A temporary directory is required some file operations such as mailing files. The default temporary directory is `/tmp`. The temporary files are deleted after the required operation has completed. **Note –** Ensure that the ID with which the web server is running (such as `nobody` or `noaccess`) has `rwx` permissions for the specified directory. Also ensure that the ID has `rx` permissions for the entire path to the required temporary directory. **Tip –** You may want to create a separate temporary directory for NetFile. If you specify a temporary directory that is common to all modules of the Portal Server, the disk may quickly run out of space. A few operations in NetFile, such as mailing files, do not work if the temporary directory has no space.

Attribute Name	Description
File Upload Limit (MB)	Enter the maximum size of the files that can be uploaded in this field. The default value is 5MB. When the size of the file being uploaded exceeds the limit specified here, an error message is displayed and the file is not uploaded. If you enter an invalid value, NetFile resets the value to the default value. You can specify different file upload size limits for different users.
Search Directories Limit	Enter the maximum number of directories that can be searched in a single search operation. This limit helps reduce network clogging and increases the speed of access if a number of users are logged in simultaneously. The default value is 100. Suppose a user has a directory called *A*. Assume that *A* has 100 subdirectories. If you specify the maximum directories to be searched as 100, the search operation goes through directory A and stops. The search does not proceed through the other directories in the users machine since the limit of 100 was reached with directory *A*. The search results accumulated until the search limit is reached are displayed to the user along with an error message stating that the search exceeded its limit. To continue the search, the user must manually restart the search at the next directory. The search operation is carried out in a depth-first manner. This means that the search operation is carried out in all the subdirectories of the directory that the user selected, before moving on to the next directory.

5 **Click Save to complete.**

▼ To Configure the Operation Privileges

You can allow or deny permission for users to perform the following tasks from remote hosts.

1 **Log onto the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab and select the Netfile tab.**

3 **Select a DN for a user or an organization from Select DN list or add a DN.**

4 **Modify the following attributes:**

Attribute Name	Description
File Rename	Select the Allow checkbox to enable users to rename files. This option is selected by default.
File/Folder Deletion	Select the Allow checkbox to enable users to delete files and directories. This option is selected by default.
File Upload	Select the Allow checkbox to enable users to upload files. This option is selected by default.
File/Folder Download	Select the Allow checkbox to enable users to download files or directories. This option is selected by default.
File Search	Select the Allow checkbox to enable users to perform file search operations. This option is selected by default.
File Mail	Select the Allow checkbox to enable users to access to mail. This option is selected by default.
File Compression	Select the Allow checkbox to enable users to choose the compression type. This option is selected by default.
Changing User Id	Select the Allow checkbox to enable users to change their user ID. Users can use different IDs to connect to hosts using NetFile. In a large organization, users may have multiple user IDs. You may want to restrict users to use a single user ID. In that case, you can disable the Allow Changing User ID option. This prevents all the users in the specific organization from changing their user ID, and limits them to using a single ID (the desktop login ID) to connect to hosts using NetFile. In another situation, a user may have different login IDs on different machines, in which case, you may want to allow the user to change the ID as required.
Changing Microsoft Windows Domains	Select the Allow checkbox to enable users to change the default Microsoft Windows domain host. This option is selected by default. When the user specifies a domain name, the username and password for that domain also needs to be specified. If the username and password for the host needs to be used, the user needs to remove the domain from the User Domain name field.

Note – When the any of the above options are not selected, the changes takes effect only after the user logs onto Portal Server desktop again.

---

5. **Click Save to complete.**

CHAPTER 15

Configuring Secure Socket Layer Accelerators

This chapter describes configuring various accelerators for Sun Java System Portal Server Secure Remote Access.

This chapter contains the following sections:

Introduction to Accelerators

External accelerators are dedicated hardware co-processors that off-load the Secure Socket Layer (SSL) functions from a server\qs CPU, thereby freeing the CPU to perform other tasks and increasing the processing speed for SSL transactions.

Sun Crypto Accelerator 1000

The Sun™ Crypto Accelerator 1000 (Sun CA1000) board is a short PCI board that functions as a cryptographic co-processor to accelerate public key and symmetric cryptography. This product has no external interfaces. The board communicates with the host through the internal PCI bus interface. The purpose of this board is to accelerate a variety of computationally intensive cryptographic algorithms for security protocols in eCommerce applications.

Many critical cryptographic functions, such as RSA [7] and Triple-DES (3DES) [8], can be off-loaded from an application to the Sun CA1000 and performed in parallel. This frees the CPU to perform other tasks, increasing the processing speed for SSL transactions.

See "To Configure Crypto Accelerator 1000" on page 230 for steps.

Enable Crypto Accelerator 1000

Ensure that Portal Server Secure Remote Access has been installed, and a gateway server certificate (self-signed or issued by any CA) has been installed. See the Chapter 10, “Working with Certificates,” for details.

“Enable Crypto Accelerator 1000” on page 230 is a checklist to help you keep track of the required information before installing the SSL Accelerator.lists the Crypto Accelerator 1000 parameters and values.

TABLE 15–1 Crypto Accelerator 1000 Installation Checklist

Parameter	Value
SRA installation base directory	/opt
SRA certificate database path	/etc/opt/SUNWportal/cert/default
SRA server certificate nickname	server-cert
Realm	sra-keystore
Realm user	crypta

▼ To Configure Crypto Accelerator 1000

1. **Follow the instructions in the user's guide to install the hardware. See:**

   `http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-2450-11.pdf`

2. **Install the following packages from the CD.**

   `SUNWcrypm, SUNWcrypu, SUNWcrysu, SUNWdcar, SUNWcrypr, SUNWcrysl, SUNWdcamn, SUNWdcav`

3. **Install the following patches. (You can get them from the `http://sunsolve.sun.com`)**

   110383-01, 108528-05, 112438-01

4. **Make sure you have the tools `pk12util` and `modutil`.**

   These tools are installed under `/usr/sfw/bin`. If the tools are not available in the `/usf/sfw/bin` directory, you need to manually add the SUNWtlsu package from the Sun Java System distribution media:

   ```
   Solaris_[sparc/x86]/Product/shared_components/
   ```

5. **Create the slots file:**

   ```
   vi /etc/opt/SUNWconn/crypto/slots
   ```

   and put "crypta@sra" as the first and only line in the file.

6 Create and set a realm.

a. **Login as root.**

b. Type these commands:

```
cd /opt/SUNWconn/bin/secadm
```

```
secadm> create realm=sra
```

Realm sra created successfully.

7 Create a user:

a. **Type and respond to these commands:**

```
secadm> set realm=sra
```

```
secadm{srap}> su
```

```
secadm{root@sra}>create user=crypta
```

Initial password:

Confirm password:

User crypta created successfully.

8 Login as the user you created.

```
secadm{root@sra}> login user=crypta
```

Password:

```
secadm{crypta@sra}> show key
```

No keys exist for this user.

9 Load the Sun Crypto module.

The environment variable `LD_LIBRARY_PATH` must point to `/usr/lib/mps/secv1/`

Type:

```
modutil -dbdir /etc/opt/SUNWportal/cert/default -add "Sun Crypto Module"
-libfile /opt/SUNWconn/crypto/lib/libpkcs11.so
```

Use the following command to verify that this module is loaded:

```
modutil -list -dbdir /etc/opt/SUNWportal/cert /default
```

10 Export the gateway certificate and the key to the "Sun Crypto Module".

The environment variable `LD_LIBRARY_PATH` must point to `/usr/lib/mps/secv1/`

Type:

```
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"
```

Now run the show key command:

secadm{crypta@sra}> `show key`

You should see two keys for this user.

11 **Change the nickname in the /etc/opt/SUNWportal/cert/default/.nickname file.**

```
vi /etc/opt/SUNWportal/cert/default/.nickname
```

replace the `server-cert` with `crypta@sra:server-cert`

12 **Enable ciphers for acceleration.**

SUN CA1000 accelerates RSA functions but supports acceleration only for DES and 3DES ciphers.

13 **Modify the `/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name* to enable the accelerator:**

```
gateway.enable.accelerator=true
```

14 **From a terminal window, restart the gateway:**

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

Note – Gateway binds to a plain ServerSocket (non SSL) on the port mentioned as https port in the gateway profile.

No SSL encryption or decryption is done on the incoming client traffic. This is done by the accelerator.

PDC is not be functional in this mode.

Sun Crypto Accelerator 4000

The Sun™ Crypto Accelerator 4000 board is a Gigabit Ethernet-based network interface card that supports cryptographic hardware acceleration for IPsec and SSL (both symmetric and asymmetric) on Sun servers.

In addition to operating as a standard Gigabit Ethernet network interface card for unencrypted network traffic, the board contains cryptographic hardware to support a higher throughput for encrypted IPsec traffic.

The Crypto Accelerator 4000 board accelerates cryptographic algorithms in both hardware and software. It also supports bulk encryption for ciphers DES and 3DES.

See “To Configure Crypto Accelerator 4000” on page 233 for steps.

Enable Crypto Accelerator 4000

Ensure that SRA has been installed and a gateway server certificate (self-signed or issued by any CA) has been installed. The following checklist helps you keep track of the required information before installing the SSL Accelerator.

“Enable Crypto Accelerator 1000” on page 230 lists the Crypto Accelerator 4000 parameters and values.

TABLE 15–2 Crypto Accelerator 4000 Installation Checklist

Parameter	Value
Portal Server Secure Remote Access installation base directory	/opt
SRA instance	default
SRA certificate database path	/etc/opt/SUNWportal/cert/default
SRA server certificate nickname	server-cert
CA4000 keystore	srap
CA4000 keystore user	crypta

▼ To Configure Crypto Accelerator 4000

1. **Follow the instructions in the user\qs guide to install the hardware and the software packages. See:**

   http://www.sun.com/products-n-solutions/hardware/docs/pdf/816-2450-11.pdf

2. **Install the following patch. (You can get them from the http://sunsolve.sun.com):** `114795`

3. **Make sure that you have the tools `certutil`, `pk12util` and `modutil`.**

   These tools are installed under `/usr/sfw/bin`

   If the tools are not available in the `/usf/sfw/`bin directory, you need

   to manually add the SUNWtlsu package from the Sun Java System distribution media:

   `Solaris_[sparc/x86]/Product/shared_components/`

4. **Initialize the board.**

   Run the `/opt/SUNWconn/bin/vcadm` tool to initialize the crypto board and set the following values.

Initial Security Officer Name: `sec_officer`

Keystore name: `sra-keystore`

Run in FIPS 140-2 Mode: `No`

5 **Create a user.**

vcaadm{vca0@localhost, sec_officer}> `create user`

New user name: `crypta`

Enter new user password:

Confirm password:

User crypta created successfully.

6 **Map token to the key store.**

```
vi /opt/SUNWconn/cryptov2/tokens
```

and append `sra-keystore` to the file.

7 **Enable bulk encryption.**

```
touch /opt/SUNWconn/cryptov2/sslreg
```

8 **Load the Sun Crypto module.**

The environment variable `LD_LIBRARY_PATH` must point to `/usr/lib/mps/secv1/`

Type:

```
modutil -dbdir /etc/opt/SUNWportal/cert/default -add "Sun Crypto Module"
-libfile /opt/SUNWconn/cryptov2/lib/libvpkcs11.so
```

You can verify that this module is loaded using the following command:

```
modutil -list -dbdir /etc/opt/SUNWportal/cert/default
```

9 **Export the gateway certificate and the key to the "**`Sun Crypto Module`**".**

The environment variable `LD_LIBRARY_PATH` must point to `/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"
```

You can verify that the key has been exported using the following command:

```
certutil -K -h "sra-keystore" -d /etc/opt/SUNWportal/cert/default
```

10 **Change the nickname in the** `/etc/opt/SUNWportal/cert/default/.nickname` **file:**

```
vi /etc/opt/SUNWportal/cert/default/.nickname
```

replace the `server-cert` with `sra-keystore:server-cert`

11 **Enable the ciphers for acceleration.**

12 **From a terminal window, restart the gateway:**

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

The Gateway prompts you to enter the keystore password.

Enter Password or Pin for `"sra-keystore":crypta:crytpa-password`

Note – Gateway binds to a plain ServerSocket (non SSL) on the port mentioned as https port in the gateway profile.

No SSL encryption or decryption is done on the incoming client traffic. This is done by the accelerator.

PDC is not be functional in this mode.

External SSL Device and Proxy Accelerators

An external SSL device can run in front of Portal Server Secure Remote Access (SRA) in open mode. It provides the SSL link between the client and SRA.

The following tasks can be performed:

- "To Enable an External SSL Device Accelerator" on page 235
- "To Configure External SSL Device Accelerators" on page 236

▼ To Enable an External SSL Device Accelerator

1 **Ensure that SRA has been installed and a gateway is running in open mode (HTTP mode).**

2 **Enable an HTTP Connection.**

The table lists the external SSL device and proxy accelerator parameters and values.

Parameter	Value
SRA instance	default
Gateway Mode	http

Parameter	Value
Gateway Port	880
External Device/Proxy Port	443

▼ To Configure External SSL Device Accelerators

1 **Follow the instructions in the user guide to install the hardware and software packages.**

2 **Install the required patches, if any.**

3 **Configure a gateway instance to use HTTP.**

4 **Enter the following values in the `platform.conf` file:**

`gateway.enable.customurl=true`

`gateway.enable.accelerator=true`

`gateway.httpurl=https://`*external-device-URL:port-number*

5 **Gateway notification can be configured in two ways:**

- When the Access Manager can contact the gateway machine at port 880 (Session notifications are in HTTP), enter values in the `platform.conf` file.

  ```
  vi /etc/opt/SUNWportal/platform.conf.default
  ```

  ```
  gateway.protocol=http
  ```

  ```
  gateway.port=880
  ```

  - When the Access Manager can contact the external device/proxy at port 443 (Session notifications are be in HTTPS), enter values in the `platform.conf` file.

    ```
    vi /etc/opt/SUNWportal/platform.conf.default
    ```

    ```
    gateway.host=External Device/Proxy Host Name
    ```

    ```
    gateway.protocol=https
    ```

    ```
    gateway.port=443
    ```

6 **Make sure that the SSL device/proxy is up and running and configured to tunnel the traffic to the gateway port.**

7 **From a terminal window, restart the gateway:**

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

PART III

Managing the Secure Remote Access Server

The Secure Remote Access server has two interfaces for administration:

- Portal Server management console
- The Chapter 1, “psadmin Utility,” in *Sun Java System Portal Server 7.2 Command-Line Reference* command line utility

Most administration tasks are performed through the web-based Portal Server management console which can be accessed locally or remotely using a web browser. For more information, see “Using the Portal Server Management Console” in *Sun Java System Portal Server 7.2 Administration Guide*.

However, tasks such as file modification must be administered through the UNIX command-line interface.

CHAPTER 16

Managing the Gateway

highlights here

Tasks to Manage the Gateway

This section has the following tasks to manage the portal server gateway:

- "To Create a Gateway Profile" on page 239
- "To Create Gateway Instances Using the Same LDAP" on page 240

▼ To Create a Gateway Profile

1 **Log into the Portal Server administration console as administrator.**

2 **Click the Secure Remote Access tab and click New Profile.**

The New Profile page is displayed.

3 **Enter the name of the new gateway profile.**

4 **Select the profile to use for creating the new profile from the drop-down list.**

By default, any new profile that you create is based on the pre-packaged Default profile. If you have created a custom profile, you can select that profile from the drop-down list. The new profile inherits all the attributes of the selected profile.

The existing profile that is copied for the new one, copies the same port. Change the port for the new profile so that it does not conflict with the existing one.

5 **Click OK.**

The new profile is created and listed in the Profiles page.

Caution – Ensure that you change the port of the instance so that it does not clash with any existing port in use.

6 **Telnet to the machine where the instance needs to be created. The default gateway instance is up and running at this machine.**

7 **Install AM-SDK in configure now mode.**

8 **Install Gateway using UI installer in configure now mode or select configure later mode.**

9 **Copy the** `/opt/SUNWportal/template/sra/GWConfig.properties.template` **file to a temporary location . For example,** `/tmp`**.**

10 **Modify the values as required.**

Note – The values should match the port numbers in the gateway instance for the new profile.

11 **Once complete, run the following command:**

```
./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file
location>.template -t gateway
```

12 **Restart the Gateway with this gateway profile name to ensure the changes to take effect:**

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

For more information on starting and stopping the Gateway, see “To Start the Gateway Instances” on page 241. To configure the Gateway, see Chapter 8, “Configuring the Secure Remote Access Gateway”

▼ To Create Gateway Instances Using the Same LDAP

1 **Replace the key that is used to encrypt and decrypt passwords with the same string used for the first Gateway.**

```
am.encryption.pwd= string_key_specified_in gateway-install
```

2 **Replace the key that is the shared secret for application authentication module:**

```
com.iplanet.am.service.secret= string_key_specified_in gateway-install
```

3 **In** `/etc/opt/SUNWam/config/ums` **modify the following areas in** `serverconfig.xml` **to be consistent with the first installed instance of Portal Server:**

```
<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 **Restart Access Manager services.**

▼ To Start the Gateway Instances

By default, the Gateway starts as user `noaccess`.

1 **After installing the Gateway and creating the required profile, run the following command to start the Gateway:**

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

`default` — is the default gateway profile that is created during installation. You can create your own profiles later, and restart the Gateway with the new profile. See "Creating a Gateway Profile" on page 32.

Note – Replace the *<profile name>* with an appropriate profile name to start other instances of the Gateway.

Restarting the server (the machine on which the Gateway instances are configured) restarts all instances of the Gateway.

Ensure that no backed up profiles are present in the `/etc/opt/SUNWportal` directory.

---

2 **Run the following command to check if the Gateway is running on the specified port:**

`netstat -an | grep` *port-number*

The default Gateway port is 443.

▼ To Stop the Gateway

1 **Use the following command to stop the Gateway:**

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

Note – Replace the <*profile name*> with an appropriate profile name to start other instances of the Gateway.

2. **Run the following command to verify if any of the Gateway processes are still running:**

   ```
   /usr/bin/ps -ef | grep entsys
   ```

▼ To Start and Stop Gateway Using Management Console

1. **"To Login to the Management Console" in** *Sun Java System Portal Server 7.2 Administration Guide*

2. **Select the Secure Remote Access tab.**

3. **Click the Manage Instances submenu.**

4. **Under SRA Proxy instances, select an instance.**

   - **Click Start to start an instance.**

   - **Click Stop to stop an instance.**

▼ To Restart the Gateway with a Different Profile

● **Restart the Gateway:**

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

▼ To Restart the Gateway

● **In a terminal window, connect as root and do one of the following:**

- Start the watchdog process:

  ```
  ./psadmin sra-watchdog -u uid -f password-filename -t instance-type on
  ```

`[--adminuser | -u] uid`	Specifies the administrator's distinguished name (DN) or user ID.
`[-passwordfile | -f] password-filename`	Specifies the administrator's password in the password file.
`[--type | -t] instance-type`	Specifies the type of the Secure Remote Access instance. Enter: gateway, nlproxy, or rwproxy.

For information on watchdog command, see the *Sun Java System Portal Server Command Line Reference Guide*.

This creates an entry in the crontab utility and the watchdog process is now active. The watchdog monitors all running instances of a Gateway on a particular machine and Gateway port and restarts the Gateway if it goes down.

▼ To Specify a Virtual Host

1 **Login as root and edit the `platform.conf`file of the required Gateway instance:**

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

2 **Add the following entries:**

`gateway.virtualhost=`*fully-qualified-gateway-host gateway-ip-address fully-qualified-reverse-proxyhost*

`gateway.enable.customurl=true` (This value is set to false by default.)

3 **Restart the Gateway:**

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

If these values are not specified, the Gateway defaults to normal behavior.

▼ To Specify a Proxy

1 **From the command-line, edit the following file:**

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

2 **Add the following entries:**

```
http.proxyHost=proxy-host
http.proxyPort=proxy-port
http.proxySet=true
```

3 **Restart the Gateway to use the specified proxy for requests made to the server:**

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

To create a Netlet Proxy instance

1 **Telnet to the machine where the instance needs to be created. The default gateway instance is up and running at this machine.**

2 **Copy the** `/opt/SUNWportal/template/sra/NLPConfig.properties.template` **file to a temporary location . For example,** `/tmp`**.**

3 **Modify the values as required in the file for the new profile.**

4 **Once complete, run the following command:**

```
./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file
location>.template -t nlproxy
```

5 **Start the new instance of the Netlet proxy with the required gateway profile name to ensure that the changes take effect:**

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

To Restart a Netlet Proxy

● **In a terminal window, connect as root and do one of the following:**

- **Start the watchdog process:**

  ```
  psadmin sra-watchdog -u uid -f password-filename -t instance-type on
  ```

  Enter `nlproxy` in place of the *instance-type*. For more information on this command, see the *Sun Java Portal Server Command Line Reference Guide*.

  This creates an entry in the crontab utility and the watchdog process is now active. The watchdog monitors the Netlet proxy port and brings up the proxy if it goes down.

- **Start a Netlet proxy manually:**

  ```
  psadmin start-sra-instance -u uid -f password-filename -N sra-instance-name
  -t instance-type
  ```

Enter `nlproxy` in place of the *instance-type*. This the profile name corresponding to the required Netlet Proxy instance. For more information on this command, see the *Sun Java Portal Server Command Line Reference Guide*.

▼ To Create a Rewriter Proxy Instance

1. **Telnet to the machine where the instance needs to be created. The default gateway instance is up and running at this machine.**

2. **Copy the** `/opt/SUNWportal/template/sra/GWConfig.properties.template` **file to a temporary location . For example,** `/tmp`**.**

3. **Modify the values as required in the file for the new profile.**

4. **Once complete, run the following command:**

   ```
   ./psadmin create-sra-instance -u amadmin -f <passwordfile> -S <template file
   location>.template -t rwproxy
   ```

5. **Start the new instance of the Rewirter Proxy with the required gateway profile name to ensure that the changes take effect:**

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

▼ To Restart a Rewriter Proxy

● **In a terminal window, connect as root and do one of the following:**

- **Start the watchdog process:**

  ```
  psadmin sra-watchdog -u uid -f password-filename -t instance-type on
  ```

  Enter `rwproxy` in place of the *instance-type*. For more information on this command, see the *Sun Java Portal Server Command Line Reference Guide*.

  This creates an entry in the crontab utility and the watchdog process is now active. The watchdog monitors the Rewriter Proxy port and brings up the proxy if it goes down.

- **Start a Rewriter Proxy manually:**

  ```
  start-sra-instance -u uid -f password-filename -N sra-instance-name -t
  instance-type
  ```

Enter `rwproxy` in place of the *instance-type*. This the profile name corresponding to the required Rewritter Proxy instance. For more information on this command, see the *Sun Java Portal Server Command Line Reference Guide*.

▼ To Enable a Reverse Proxy

1 **Log in as root and edit the `platform.conf` file of the required Gateway instance:**

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

2 **Add the following entries:**

`gateway.virtualhost=`*fully-qualified-gateway-host gateway-ip-address fully-qualified-reverse-proxyhost*

`gateway.enable.customurl=true` (This value is set to false by default.)

`gateway.httpurl=`*http reverse-proxy-URL*

`gateway.httpsurl=`*https reverse-proxy-URL*

`gateway.httpurl` is used to rewrite the response for the request received at the port which is listed as HTTP port in the gateway profile.

`gateway.httpsurl` is used to rewrite the response for the request received at the port which is listed as HTTPS port in the gateway profile.

3 **Restart the Gateway:**

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

If these values are not specified, the Gateway defaults to normal behavior.

▼ To Add Authentication Modules to an Existing PDC Instance

1 **Login to the Access Manager administration console as administrator.**

2 **Select the required organization.**

3 **Select Services from the View drop-down box.**

The services are displayed.

4 **Click Authentication Configuration.**

The Service Instance List is displayed.

5 **Click Gatewaypdc.**

The Gatewaypdc properties page is displayed.

6 **Click Edit.**

The Add Module page is displayed.

7 **Select Module Name and set Flag to Required.**

8 **Click OK.**

9 **Click Save after adding one or more modules.**

10 **Click Save in the `gatewaypdc` properties page.**

11 **Restart the Gateway for the changes to take effect:**

gateway-install-location`/SUNWportal/bin/psadmin start-sra-instance –u amadmin – f` *<password file>* `–N` *<profile name>*`– t` *<gateway>*

▼ To Disable Browser Caching

1 **Login as root and edit the `platform.conf` file of the required Gateway instance:**

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

2 **Edit the following line:**

```
gateway.allow.client.caching=true
```

This value is set to `true` by default. Change the value to `false` to disable browser caching at the client side.

3 **Restart the Gateway:**

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

▼ To Share LDAP Directories

1 **Modify the following areas in `AMConfig.properties` to synchronize with the first installed instance of Portal Server and Access Manager servers:**

# The key that will be used to encrypt and decrypt passwords.
`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 **In `/etc/opt/SUNWam/config/ums` modify the following areas in `serverconfig.xml` to be insync with the first installed instance of Portal Server and Access Manager server:**

```
<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 **Restart the Access Manager services.**

CHAPTER 17

Federation Management Scenarios

This chapter describes .... The following topics are discussed:

- "Using Federation Management" on page 249
- "Federation Management Scenario" on page 250
- "Configuring Federation Management Resources" on page 250

Using Federation Management

Federation Management enables users to aggregate their local identities so that they have one network identity. Federation Management uses the network identity to allow users to login at one service provider's site and access other service providers' sites without having to re-authenticate their identity. This is referred to as single sign-on.

Federation management can be configured in open mode and secure mode on the Portal Server. The Portal Server Administration Guide describes how to configure federation management in open mode. Before configuring Federation management in secure mode, using Portal Server Secure Remote Access server, ensure that it works in open mode. If you want your users to use Federation Management from the same browser in both open and secure mode, they must clear the cookies and cache from the browser.

For detailed information on Federation Management, see the *Access Manager Federation Management Guide*.

Federation Management Scenario

A user authenticates to an initial service provider. Service providers are commercial or not-for-profit organizations that offer web-based services. This broad category can include internet portals, retailers, transportation providers, financial institutions, entertainment companies, libraries, universities, and governmental agencies.

The service provider uses a cookie to store the user's session information in the client browser. The cookie also includes the user's identity provider.

Identity providers are service providers that specialize in providing authentication services. As the administrating service for authentication, they also maintain and manage identity information. Authentication accomplished by an identity provider is honored by all service providers with whom they are affiliated.

When the user attempts to access a service that is not affiliated with the identity provider, the identity provider forwards the cookie to the unaffiliated service provider. This service provider can then access the identity provider called out in the cookie.

However, cookies cannot be read across different DNS domains. Therefore a Common Domain Cookie Service is used to redirect the service provider to the correct identity provider thus enabling single sign-on for the user.

Configuring Federation Management Resources

The Federation resources, the service providers, identity providers, and the Common Domain Cookie Service (CDCS), are configured in the gateway profile based on where they reside. This section describes how to configure three scenarios:

▼ To Configure Federation Management Resources

1. **When all resources are inside the corporate intranet**

2. **When all resources are not inside the corporate intranet or the identity provider resides in the Internet**

3. **When all resources are not inside the corporate intranet or the service provider is a third party residing in the Internet while the identity provider is protected by the Gateway.**

Configuration 1

In this configuration the service providers, identity providers and the Common Domain Cookie Service are deployed in the same corporate intranet and the identity providers are not published in the Internet Domain Name Server (DNS). The CDCS is optional.

In this configuration the Gateway points to the service provider, which is the Portal Server. This configuration is valid for multiple instances of the Portal Server.

▼ To Configure Gateway to a Service Provider (Portal Server)

1. **Log into the Portal Server administration console as administrator.**

2. **Select the Secure Remote Access tab and select the appropriate gateway profile to modify its attributes.**

   The Edit Gateway Profile page is displayed.

3. **Select the Core tab.**

4. **Select the Enable Cookie Management checkbox to enable cookie management.**

5. **Select the Security tab.**

6. **In the Portal Servers field, enter Portal Server names to use the relative URLs such as: `/amserver` or `/portal/dt` listed in the Non-Authenticated URLs list. For example:**

   `http://`*idp-host:port*`/amserver/js`

   `http://`*idp-host:por*`t/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. **In the Portal Servers field, enter the Portal Server name. For example, `/amserver`.**

8. **Click Save.**

9. **Select the Security tab.**

10 **In the Non-Authenticated URLs list, add the federation resources. For example:**

```
/amserver/config/federation

/amserver/IntersiteTransferService

/amserver/AssertionConsumerservice

/amserver/fed_images

/amserver/preLogin

/portal/dt
```

11 **Click Add.**

12 **Click Save.**

13 **If web proxies are needed to reach the URLs listed in the Non-authenticated URLs list, select the Deployment tab.**

14 **In the Proxies for Domains and Subdomains field, enter the necessary web proxies.**

15 **Click Add.**

16 **Click Save.**

17 **From a terminal window, restart the Gateway:**

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

Configuration 2

In this configuration the identity providers, identity providers and the Common Domain Cookie Provider (CDCP) are not deployed in the corporate intranet or the identity provider is a third party provider residing the in Internet.

In this configuration the Gateway points to the service provider, which is the Portal Server. This configuration is valid for multiple instances of the Portal Server.

▼ To Configure Gateway to a Service Provider (Portal Server)

1 **Log into the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab and select the appropriate gateway profile to modify its attributes.**

3 **Select the Core tab.**

4 **Select the Enable Cookie Management checkbox to enable cookie management.**

5 **In the Portal Servers field, enter portal server names of the service provider to use the relative URLs such as:** `/amserver` **or** `/portal/dt` **listed in the Non-Authenticated URLs list.**

`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 **Click Save.**

7 **Click the Security tab.**

8 **In the Non-Authenticated URLs list, add the Federation resources. For example:**

```
/amserver/config/federation

/amserver/IntersiteTransferService

/amserver/AssertionConsumerservice

/amserver/fed_images

/amserver/preLogin

/portal/dt
```

9 **Click Add.**

10 **Click Save.**

11 **If web proxies are needed to reach the URLs listed in the Non-authenticated URLs list, select the Deployment tab.**

12 **In the Proxies for Domains and Subdomains field, enter information about the web proxies.**

13 **Click Add.**

14 **Click Save.**

15 **From a terminal window, restart the Gateway:**

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

Configuration 3

In this configuration the identity providers, identity providers and the Common Domain Cookie Provider (CDCP) are not deployed in the corporate intranet or the service provider is a third party provider residing the in Internet and the identity provider is protected by the Gateway.

In this configuration the Gateway points to the identity provider, which is the Portal Server.

This configuration is valid for multiple instances of the Portal Server. This configuration is unlikely on the Internet, however, some corporate networks may have such a configuration within their intranet, that is the identity provider may reside in a subnet this is protected by a firewall and the service providers are directly accessible from within the corporate network.

▼ To Configure Gateway to an Identity Provider (Portal Server)

1 **Log into the Portal Server administration console as administrator.**

2 **Select the Secure Remote Access tab and select the appropriate gateway profile to modify its attributes.**

3 **Select the Core tab.**

4 **Select the Enable Cookie Management checkbox to enable cookie management.**

5 **In the Portal Servers field, enter the portal server name of the identity provider to use the relative URLs such as:** `/amserver` **or** `/portal/dt` **listed in the Non-Authenticated URLs list.**

`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 **Click Save.**

7 **Select the Security tab.**

8 **In the Non-authenticated URLs list, add the federation resources. For example:**

```
/amserver/config/federation

/amserver/IntersiteTransferService

/amserver/AssertionConsumerservice

/amserver/fed_images

/amserver/preLogin

/portal/dt
```

9 **Click Add.**

10 **Click Save.**

11 **If web proxies are needed to reach the URLs listed in the Non-authenticated URLs list, select the Deployment tab.**

12 **In the Proxies for Domains and Subdomains field, enter information about the web proxies.**

13 **Click Add.**

14 **Click Save.**

15 **From a terminal window, restart the Gateway:**

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

APPENDIX A

Configuration Attributes

This appendix describes attributes that you can configure for Sun Java System Portal Server Secure Remote Access through the Portal Server administration console for each Portal Server Secure Remote Access component:

- “Access Control Service” on page 257
- “Gateway Service” on page 258
- “NetFile Service” on page 264
- “Netlet Service” on page 269
- “Proxylet Service” on page 270

Access Control Service

“Access Control Service” on page 257 lists the Access Control service attributes.

TABLE A–1 Access Control Service Attributes

Attribute	Default Value	Description
Denied URLs		List of URLs that end-users cannot access through Gateway.
Allowed URLs	*	List of URLs that end-users can access through Gateway.
Single Sign On Disabled Hosts		Disables single sign-on for a list of hosts.
Enable Single Sign On per Session		Enables single sign-on for a session.

TABLE A–1 Access Control Service Attributes *(Continued)*

Attribute	Default Value	Description
Allowed Authorization Levels	*	Indicates how much to trust an authentication. Use an asterisk to allow all authentication levels. For information on authentication levels, see the *Access Manager Administration Guide*.

Gateway Service

When you click the Gateway service, the right pane displays a button to create a new profile and a list of any gateway profiles that have been created.

If you click New, the next pane prompts you to enter the new gateway profile name. You have the option to use the default template or a previously created gateway profile as the template.

If you click one of the listed gateway profile names, a list of tabs are presented. They are:

- "Core" on page 258
- "Proxies" on page 260
- "Security" on page 261
- "Rewriter" on page 262

Core

"Core" on page 258 lists the Gateway service core attributes.

TABLE A–2 Gateway Service Core Attributes

Attribute	Default Value	Description
Enable HTTPS Connections		Enables HTTPS connections.
HTTPS Port	443	Specifies the HTTPS port.
Enable HTTP Connections	*	Enables HTTP connections.
HTTP Port	80	Specifies the HTTP port.
Enable Rewriter Proxy	*	Enables secure HTTP traffic between Gateway and the intranet. Rewriter proxy and Gateway use the same gateway profile.
Rewriter Proxy List		List of Rewriter proxies. For multiple instances of Rewriter proxies enter the details for each in the form *host-name:port*

TABLE A–2 Gateway Service Core Attributes *(Continued)*

Attribute	Default Value	Description
Enable Netlet	Checked	Enables security for TCP/IP (such as Telnet and SMTP), HTTP applications, and fixed port applications.
Enable Proxylet	Checked	Enables the download of Proxylet on a client machine.
Enable Netlet Proxy		Enhances security for Netlet traffic between Gateway and the intranet by extending the secure tunnel from the client, through Gateway to Netlet proxy residing on the intranet. Disable if you do not want to use applications with Portal Server.
Netlet Proxy Hosts		Lists Netlet proxy hosts, in the format: hostname:port
Enable Cookie Management		Tracks and manages user sessions for all web sites that the user is permitted to access. (Does not apply to the cookies used by Portal Server to track Portal Server user sessions).
Enable Persistent HTTP Connections	Checked	Enables HTTP persistent connections at Gateway to prevent sockets being opened for every object (such as images and style sheets) in the web pages.
Maximum Number of Requests per Persistent Connection	10	Specifies the number of requests per persistent connection.
Timeout for Persistent Socket Connections	50	Specifies the amount of time that needs to lapse before sockets are closed.
Grace Timeout to Account for Turnaround Time	20	Specifies the grace amount of time for the request to reach Gateway after the browser has sent i and the time between gateway sending the response and the browser actually receiving it.
URLs to which User Session Cookie is Forwarded		Enables servlets and CGIs to receive Portal Server'ss cookie and use the APIs to identify the user.
Maximum Connection Queue Length	50	Specifies the maximum concurrent connections that Gateway can accept.
Gateway Timeout (seconds)	120	Specifies the time interval in seconds before Gateway times out its connection with the browser.

TABLE A–2 Gateway Service Core Attributes *(Continued)*

Attribute	Default Value	Description
Maximum Thread Pool Size	200	Specifies the maximum number of threads that can be pre-created in the Gateway thread pool.
Cached Socket Timeout	200	Specifies the time interval in seconds before Gateway times out its connection with Portal Server.
Portal Servers		Specifies Portal Servers in the format `http://`*portal server name:port -number*`.` Gateway tries to contact each of the Portal Servers listed in a round robin manner to service the requests.
Server Retry Interval (seconds)	120	Specifies the time interval between requests to try to start Portal Server, Rewriter proxy or Netlet proxy after it becomes unavailable (such as a crash or it was brought down).
Store External Server Cookies		Allows Gateway to store and manage cookies for any third party application or server that is accessed through Gateway.
Obtain Session Information from URL		Encodes session information as part of the URL, whether cookies are supported or not. Gateway uses this session information found in the URL for validation rather than using the session cookie that is sent from the client's browser.

Proxies

"Proxies" on page 260 lists the Gateway service proxies attributes.

TABLE A–3 Gateway Service Proxies Attributes

Attribute	Default Value	Description
Use Proxy		Enables usage of web proxies.
Use Webproxy URLs		Lists the URLs that Gateway needs to contact only through the webproxies listed in the Proxies for Domains and Subdomains list, even if the Use Proxy option is disabled.
Do Not Use Webproxy URLs		Lists URLs that Gateway can connect directly to.

TABLE A–3 Gateway Service Proxies Attributes *(Continued)*

Attribute	Default Value	Description
Proxies for Domains and Subdomains	iportal.com sun.com	Specifies which proxy to use to contact specific subdomains in specific domains.
Proxy Password List		Specifies the server name, user name and password required for Gateway to authenticate to a specified proxy server, if the proxy server requires authentication to access some or all the sites.
Enable Automatic Proxy Configuration Support		Specifies that the information provided in the Proxies for Domains and Subdomains field is to be ignored.
Automatic Proxy Configuration File location		Specifies the location of files to be used for PAC support.
Enable Netlet Tunneling via Web Proxy		Extends the secure tunnel from the client, through Gateway to the web proxy that resides in the intranet.

Security

“Security” on page 261 lists the Gateway service security attributes.

TABLE A–4 Gateway Service Security Attributes

Attribute	Default Value	Description
Enable HTTP Basic Authentication	Checked	Saves the username and password so that users need not re-enter their credentials when they revisit BASIC-protected web sites.
Non-authenticated URLs	/portal/desktop/images /amserver/login_images /portal/desktop/css /amserver/jss /amconsole/console/css /portal/searchadmin/console/js /amconsole/console/js /amserver/css	Specifies URLs that do not need any authentication, such as directories that contain images.

TABLE A–4 Gateway Service Security Attributes *(Continued)*

Attribute	Default Value	Description
Certificate-enabled Gateway hosts		Lists the certificate-enabled Gateway hosts.
Allow 40-bit Encryption		Allows 40-bit (weak) Secure Sockets Layer (SSL) connections. If you do not select this option, only 128-bit connections are supported.
Enable SSL Version 2.0	checked	Enables SSL version 2.0. Disabling SSL 2.0 means that browsers that support only the older SSL 2.0 cannot authenticate to SRA.This ensures a greater level of security.
Enable SSL Cipher Selection		Enables SSL cipher selection. You have the option of to support all the pre-packaged ciphers, or you can select the required ciphers individually. You can select specific SSL ciphers for each Gateway instance.
SSL2 Ciphers		Lists the SSL version 2 ciphers you can choose.
SSL3 Ciphers		Lists the SSL version 3 ciphers you can choose.
TLS Ciphers		Lists the TLS ciphers.
Enable SSL Version 3.0	checked	Enables SSL version 3.0. Disabling SSL 3.0 means that browsers that support only the SSL 3.0 cannot authenticate to SRA. This ensures a greater level of security.
Enable Null Ciphers		Enables null ciphers.
Trusted SSL Domains		Lists the trusted SSL domains.
Mark Cookies as secure		Marks cookies as secure. The Enable Cookie Management option must be enabled.

Rewriter

The Rewriter tab has two subsections:

- "Basic" on page 262
- "Advanced" on page 263

Basic

"Basic" on page 262 lists the Gateway service Rewriter basic attributes.

TABLE A–5 Gateway Service Rewriter Attributes - Basic

Attribute	Default Value	Description
Enable Rewriting of All URIs		Specifies that any URI is rewritten without checking against the entries in the Proxies for Domains and Subdomains list.

Map URIs to RuleSets	`*://*.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`
		Associates a domain with the ruleset using the Map URIs to RuleSets list. Rulesets are created under Portal Server Configuration in the Access Manager administration console.

Map Parser to MIME Types	`JAVASCRIPT=application/x-java`
	`XML=text/xml`
	`HTML=text/html;text/htm;text/x-component;text/wml;text/vnd.wap.wml`
	`CSS=text/css`
		Associates new MIME types with HTML, JAVASCRIPT, CSS or XML. Separate multiple entries with a semicolon or a comma.

URIs Not to Rewrite		Lists the URIs not to rewrite. Note: Adding #* to this list allows URIs to be rewritten, even when the href rule is part of the ruleset.
Default Domains		Resolves a host name to a default domain and subdomain. This is specified during installation

Advanced

"Advanced" on page 263 lists the Gateway service Rewriter advanced attributes.

TABLE A–6 Gateway Service Rewriter Attributes - Advanced

Attribute	Default Value	Description
Enable MIME Guessing		Enables MIME guessing when MIME is not sent. You must add data to the Map Parser to URIs list box.
Map Parser to URI Mappings		Maps a parser to the URI. Multiple URIs are separated by a semicolon. For example HTML=*.html; *.htm;*Servlet means that Rewriter is used to rewrite the content for any page with a html, htm, or Servlet extension.
Enable Masking		Allows Rewriter to rewrite a URI so that the Intranet URL of a page is not seen.
Seed String for Masking		Specifies a seed string used for masking a URI. A masking algorithm generates this random string.
URIs not to Mask		Specifies Internet URIs not to be mask. This is used when applications (such as an applet) require an Internet URI. For example if you added */Applet/Param* to the list box, the URL would not be masked if the content URI http://abc.com/Applet/Param1.html is matched in the ruleset rule.
Make Gateway protocol Same as Original URI Protocol		Enables Rewriter to use a consistent protocol to access the referred resources in the HTML content. This applies only to static URIs, not to dynamic URIs generated in Javascript.

NetFile Service

When you click the NetFile Service, the right pane displays tabs. They are:

- "Hosts" on page 265
- "Permissions" on page 266
- "View" on page 266
- "Operations" on page 267
- "General" on page 268

Hosts

The Hosts tab has two subsections:

- "Config" on page 265
- "Access" on page 265

Config

"Config" on page 265 lists the NetFile hosts configuration attributes.

TABLE A–7 NetFile Service Hosts Configuration Attributes

Attribute	Default Value	Description
OS Character Set	Unicode(UTF-8)	Specifies the character set used as the default encoding for communicating with hosts.
Host Detection Order	WIN, NETWARE, FTP, NFS	Specifies the host detection order.
Common Hosts		Specifies hosts to be available through NetFile to all remote NetFile users.
Default Domain		Specifies the default domain that NetFile needs to use to contact allowed hosts.
Default Microsoft Windows Domain/Workgroup		Specifies the default Microsoft Windows domain or workgroup which the users choose to access a Windows host.
Default WINS/DNS Server		Specifies the WINS/DNS server that NetFile uses to access windows hosts.

Access

"Access" on page 265 lists the NetFile service hosts access attributes.

TABLE A–8 NetFile Service Hosts Access Attributes

Attribute	**Default Value**	**Description**
Allow Access to Windows Hosts	Checked	Allows access to Microsoft Windows hosts.
Allow Access to FTP Hosts	Checked	Allows access to FTP hosts.
Allow Access to NFS Hosts	Checked	Allows access to NFS hosts.
Allow Access to Netware Hosts	Checked	Allows access to Netware hosts.

TABLE A–8 NetFile Service Hosts Access Attributes *(Continued)*

Attribute	Default Value	Description
Allowed Hosts	*	Specifies hosts that users can access through NetFile.
Denied Hosts		Specifies hosts that users cannot access through NetFile.

Permissions

If you disable these options after the user has started using NetFile, the change takes effect only if the user logs out of NetFile and logs in again.

"Permissions" on page 266 lists the NetFile service permission attributes.

TABLE A–9 NetFile Service Permissions Attributes

Attribute	Default Value	Description
Allow File Rename	Checked	Allows users to rename files.
Allow File/Folder Deletion	Checked	Allows users to delete files and folders.
Allow File Upload	Checked	Allows users to upload files.
Allow File/Folder Download	Checked	Allows users to download files and folders.
Allow File Search	Checked	Allows users to search.
Allow File Mail	Checked	Allows file mailing.
Allow File Compression	Checked	Allows file compression.
Allow Changing User Id	Checked	Allows user to use a different ID.
Allow Changing Windows Domains	Checked	Allows users to change Microsoft Windows domains.

View

"View" on page 266 lists the NetFile Service view attributes.

TABLE A–10 NetFle Service View Attributes

Attribute	Default Value	Description
Window Size	700|400	Specifies the size of the NetFile window in pixels on the user's desktop. If you enter an invalid value, NetFile uses the default value.
Window Location	100|50	Specifies the location where the NetFile window displays on the user's desktop. If you enter an invalid value, NetFile uses the default value.

Operations

The Operations tab has the following subsections:

- "Traffic" on page 267
- "Search" on page 268
- "Compression" on page 268

Traffic

"Traffic" on page 267 lists the NetFile service operations traffic attributes.

TABLE A–11 NetFile Service Operations - Traffic Attributes

Attribute	Default Value	Description
Temporary Directory Location	/tmp	Specifies a temporary directory for various NetFile file operations. Ensure that the ID with which the web server is running (such as nobody or noaccess) has rwx permissions for the specified directory. Also ensure that the ID has rx permissions for the entire path to the required temporary directory. You may want to create a separate temporary directory for NetFile. If you specify a temporary directory that is common to all modules of the Portal Server, the disk may quickly run out of space. NetFile does not work if the temporary directory has no space.

TABLE A–11 NetFile Service Operations - Traffic Attributes *(Continued)*

Attribute	Default Value	Description
File Upload Limit (MB)	5	Specifies the maximum size of the files that can be uploaded. If you enter an invalid value, NetFile resets the value to the default. Ensure that you type an integer value. You can specify different file upload size limits for different users.

Search

"Search" on page 268 lists the NetFile service operations search attributes.

TABLE A–12 NetFile Service Operations - Search Attributes

Attribute	Default Value	Description
Search Directories Limit	100	Specifies the maximum number of directories that can be searched in a single search operation.

Compression

"Compression" on page 268 lists the NetFile service operations compression attributes.

TABLE A–13 NetFile Service Operations - Compression Attributes

Attribute	Default Value	Description
Default Compression Type	Zip	Specifies either Zip or Gzip compression type.
Default Compression Level	6	Specifies the compression level, a number between 1 and 9.

General

"General" on page 268 lists the Netfile service general attributes.

TABLE A–14 NetFile Service - General Attribute

Attribute	Default Value	Description
MIME-types Configuration File Location	/opt/S1PS62/SUNWportal/samples/config/netfile	Specifies the response content type to send to the client browser.

Netlet Service

"Netlet Service" on page 269 lists the Netlet service attributes.

TABLE A–15 Netlet Service Attributes

Attribute	Default Value	Description
Netlet Rules		Choose to add or delete a rule.
If you add a rule, the following nine attributes are necessary:		
--Rule Name		Specifies a unique name for the rule.
--Encryption Ciphers		Specifies the required ciphers.
--URL		Specifies the URL to the application to be invoked.
--Download Applet		Specifies if an applet needs to be downloaded. If an applet is used, the syntax in the associated edit box is: local-port:server-host:server-port
--Extend Session		Ensures that the Portal Server session time is extended while the Netlet session corresponding to this rule is running.
--Map Local Port to Destination Server Port		Specifies local port, target host and target ports. After entering those values (in the next three rows of this table), click add to make them appear in the list.
--Local Port		Specifies the local port on which Netlet listens. For an FTP rule, the local port value must be 30021.
--Destination Hosts		Static rules contain the host name of the destination machine for the Netlet connection. Dynamic rules contain the word "TARGET".
-- Destination Ports		Specifies the port on the destination host.
Default Native VM Cipher		Specifies the default cipher for the Netlet rules. This is useful when using existing rules that did not include the cipher as a part of the rule.
Default Java Plugin Cipher		Specifies the default cipher for the Netlet rules. This is useful when using existing rules that did not include the cipher as a part of the rule.

TABLE A–15 Netlet Service Attributes *(Continued)*

Attribute	Default Value	Description
Default Loopback Port	58000	Specifies the port to be used on the client when applets are downloaded through Netlet. The default value can be overridden in the Netlet rules.
Reauthenticate for Connections		Ensures that users enter the Netlet password each time a Netlet connection needs to be established.
Display Warning Popup for Connections	Checked	Displays a message when the user runs the application over Netlet, and also when an intruder tries to gain access to the desktop through the listen port.
Display Checkbox in Port Warning Dialog	Checked	Provides the user with the option to suppress the Warning Dialog Popup when Netlet tries to connect to the destination host on the user's standard Portal Desktop.
Keep Alive Interval (minutes)	0	If the client is connecting to the Gateway through a web proxy, then idle Netlet connections are disconnected due to proxy timeout. To prevent this, give a value less than the proxy timeout for this parameter.
Terminate Netlet at Portal Logout	Checked	Ensures that all connections are terminated when a user logs out of the Portal Server.
Access to Netlet Rules	*	Define access to specific Netlet rules for certain organizations, roles or users.
Deny Netlet Rules		Denies access to specific Netlet rules for certain organizations, roles or users.
Allowed Hosts	*	Defines access to specific hosts for certain organizations, roles or users.
Denied Hosts		Denies access to specific hosts within an organization.

Proxylet Service

"Proxylet Service" on page 270 lists the Proxylet service attributes.

TABLE A–16 Proxylet Service Attributes

Attribute	Default Values	Description
Download Proxylet Applet Automatically		When the checkbox is checked, Proxylet is downloaded to the client machine when the user logs on.
Default Proxylet Applet Bind IP	127.0.0.1	The IP address where the Proxylet Applet resides.
Default Proxylet Applet Port	58081	This is the port where Proxylet listens.

APPENDIX B

Log Files

The following log files are in the default /var/opt/SUNWportal/debug directory and contain debug and other types of information:

About Log Files

TABLE B–1 Informational and Debug Files

File Name	Contents
The following log files are controlled by the debug parameter in the AMConfig-*instance-name*.properties file in the default directory /etc/opt/SUNWam/debug/file:. For Linux path names, see Comparison of Solaris and Linux Path Names.	
amconsole	Netfile, Netlet and Gateway Admin files
srapNetFile	NetFile information file
srapNetlet	Netlet information file
srapProxylet	Proxylet information file
The following log files are controlled by the debug parameter gateway.debug in the platform.conf.gateway-profile-name file in the default directory /etc/opt/SUNWportal. For Linux path names, see Comparison of Solaris and Linux Path Names.	

TABLE B–1 Informational and Debug Files *(Continued)*

File Name	Contents
srapGateway.gateway-profile-name	Gateway information
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	Start and stop time of Rewriter Proxy
nlproxy.log.netlet-proxy-instance-name	Start and stop time of Netlet Proxy
gateway.log.gateway.instance.name	Start and stop time of the Gateway
The following Rewriter files are controlled by the debug parameter in the `AMConfig-`*instance-name*`.properties file` in the default directory `/var/opt/SUNWam/config/` file. See "Troubleshooting Using Debug Logs" on page 94 for more information.	
RuleSetInfo	All the rulesets which have been used for rewriting, are logged in this file.
Original Pages	Contains the page URI, resolved URI (if the resolved URI is different than the page URI), content MIME, the ruleset that has been applied to the page, parser MIME, and the original content. Specific error/warning/messages related to parsing also appear in this file. In message mode full content is logged, in warning and error mode only exception occurred during rewriting are logged.
Rewritten Pages	Contains the page URI, resolved URI (if the resolved URI is different than the page URI), content MIME, ruleset that has been applied to the page, parser MIME, and the rewritten content. This is filled when the debug mode is set to message.
Unaffected Pages	Contains a list the pages that were not modified.

TABLE B–1 Informational and Debug Files *(Continued)*

File Name	Contents
URIInfo Pages	This file contains the URLS found and translated. Details of all the pages whose content remain same as original data is logged in this file. Details logged are: Page URI, MIME and Encoding data, rulesetID used for rewriting, and Parser MIME.

APPENDIX C

Country Codes

The following table lists the two-letter country codes that you need to specify during certificate administration.

List of Country Codes

TABLE C–1 Two-letter Country Codes

ad	Andorra, Principality of
ae	United Arab Emirates
af	Afghanistan, Islamic State of
ag	Antigua and Barbuda
ai	Anguilla
al	Albania
am	Armenia
an	Netherlands Antilles
ao	Angola
aq	Antarctica
ar	Argentina
arpa	Old style Arpanet
as	American Samoa
at	Austria

TABLE C–1 Two-letter Country Codes *(Continued)*

au	Australia
aw	Aruba
az	Azerbaidjan
ba	Bosnia-Herzegovina
bb	Barbados
bd	Bangladesh
be	Belgium
bf	Burkina Faso
bg	Bulgaria
bh	Bahrain
bi	Burundi
bj	Benin
bm	Bermuda
bn	Brunei Darussalam
bo	Bolivia
br	Brazil
bs	Bahamas
bt	Bhutan
bv	Bouvet Island
bw	Botswana
by	Belarus
bz	Belize
ca	Canada
cc	Cocos (Keeling) Islands
cf	Central African Republic
cd	Congo, The Democratic Republic of the
cg	Congo
ch	Switzerland
ci	Ivory Coast (Cote D'Ivoire)

TABLE C–1 Two-letter Country Codes *(Continued)*

ck	Cook Islands
cl	Chile
cm	Cameroon
cn	China
co	Colombia
com	Commercial
cr	Costa Rica
cs	Former Czechoslovakia
cu	Cuba
cv	Cape Verde
cx	Christmas Island
cy	Cyprus
cz	Czech Republic
de	Germany
dj	Djibouti
dk	Denmark
dm	Dominica
do	Dominican Republic
dz	Algeria
ec	Ecuador
edu	Educational
ee	Estonia
eg	Egypt
eh	Western Sahara
er	Eritrea
es	Spain
et	Ethiopia
fi	Finland
fj	Fiji

TABLE C–1 Two-letter Country Codes *(Continued)*

fk	Falkland Islands
fm	Micronesia
fo	Faroe Islands
fr	France
fx	France (European Territory)
ga	Gabon
gb	Great Britain
gd	Grenada
ge	Georgia
gf	French Guyana
gh	Ghana
gi	Gibraltar
gl	Greenland
gm	Gambia
gn	Guinea
gov	USA Government
gp	Guadeloupe (French)
gq	Equatorial Guinea
gr	Greece
gs	S. Georgia and S. Sandwich Isls.
gt	Guatemala
gu	Guam (USA)
gw	Guinea Bissau
gy	Guyana
hk	Hong Kong
hm	Heard and McDonald Islands
hn	Honduras
hr	Croatia
ht	Haiti

TABLE C–1 Two-letter Country Codes *(Continued)*

hu	Hungary
id	Indonesia
ie	Ireland
il	Israel
in	India
int	International
io	British Indian Ocean Territory
iq	Iraq
ir	Iran
is	Iceland
it	Italy
jm	Jamaica
jo	Jordan
jp	Japan
ke	Kenya
kg	Kyrgyz Republic (Kyrgyzstan)
kh	Cambodia, Kingdom of
ki	Kiribati
km	Comoros
kn	Saint Kitts and Nevis Anguilla
kp	North Korea
kr	South Korea
kw	Kuwait
ky	Cayman Islands
kz	Kazakhstan
la	Laos
lb	Lebanon
lc	Saint Lucia
li	Liechtenstein

TABLE C–1 Two-letter Country Codes *(Continued)*

lk	Sri Lanka
lr	Liberia
ls	Lesotho
lt	Lithuania
lu	Luxembourg
lv	Latvia
ly	Libya
ma	Morocco
mc	Monaco
md	Moldavia
mg	Madagascar
mh	Marshall Islands
mil	USA Military
mk	Macedonia
ml	Mali
mm	Myanmar
mn	Mongolia
mo	Macau
mp	Northern Mariana Islands
mq	Martinique (French)
mr	Mauritania
ms	Montserrat
mt	Malta
mu	Mauritius
mv	Maldives
mw	Malawi
mx	Mexico
my	Malaysia
mz	Mozambique

TABLE C–1 Two-letter Country Codes *(Continued)*

na	Namibia
nato	NATO (this was purged in 1996 - see hq.nato.int)
nc	New Caledonia (French)
ne	Niger
net	Network
nf	Norfolk Island
ng	Nigeria
ni	Nicaragua
nl	Netherlands
no	Norway
np	Nepal
nr	Nauru
nt	Neutral Zone
nu	Niue
nz	New Zealand
om	Oman
org	Non-Profit Making Organizations (sic)
pa	Panama
pe	Peru
pf	Polynesia (French)
pg	Papua New Guinea
ph	Philippines
pk	Pakistan
pl	Poland
pm	Saint Pierre and Miquelon
pn	Pitcairn Island
pr	Puerto Rico
pt	Portugal
pw	Palau

TABLE C–1 Two-letter Country Codes *(Continued)*

py	Paraguay
qa	Qatar
re	Reunion (French)
ro	Romania
ru	Russian Federation
rw	Rwanda
sa	Saudi Arabia
sb	Solomon Islands
sc	Seychelles
sd	Sudan
se	Sweden
sg	Singapore
sh	Saint Helena
si	Slovenia
sj	Svalbard and Jan Mayen Islands
sk	Slovak Republic
sl	Sierra Leone
sm	San Marino
sn	Senegal
so	Somalia
sr	Suriname
st	Saint Tome (Sao Tome) and Principe
su	Former USSR
sv	El Salvador
sy	Syria
sz	Swaziland
tc	Turks and Caicos Islands
td	Chad
tf	French Southern Territories

TABLE C–1 Two-letter Country Codes *(Continued)*

tg	Togo
th	Thailand
tj	Tadjikistan
tk	Tokelau
tm	Turkmenistan
tn	Tunisia
to	Tonga
tp	East Timor
tr	Turkey
tt	Trinidad and Tobago
tv	Tuvalu
tw	Taiwan
tz	Tanzania
ua	Ukraine
ug	Uganda
uk	United Kingdom
um	USA Minor Outlying Islands
us	United States
uy	Uruguay
uz	Uzbekistan
va	Holy See (Vatican City State)
vc	Saint Vincent and Grenadines
ve	Venezuela
vg	Virgin Islands (British)
vi	Virgin Islands (USA)
vn	Vietnam
vu	Vanuatu
wf	Wallis and Futuna Islands
ws	Samoa

TABLE C–1 Two-letter Country Codes *(Continued)*

ye	Yemen
yt	Mayotte
yu	Yugoslavia
za	South Africa
zm	Zambia
zr	Zaire
zw	Zimbabwe

Index