Sun Java System Portal Server 7 2005Q4 Release Note Supplement

Beta

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

Part No: 819–2403

051014@13215

Contents

Preface

The Sun Java™ System Portal Server Release Note Supplement explains in detail how to install or upgrade to this version of the software and post installation configuration, discusses the new `psadmin` command line utilities that can be used to perform the basic duties of administrating the Portal Server software, describes the new inter-portlet feature, and includes reference material for the administration tag library.

Who Should Use This Book

This book includes information including new features and enhancements in the Portal Server software and is meant for administrators and other individuals installing and using this version of the product.

Before You Read This Book

Before you read this book, see the *Sun Java System Portal Server 7 2005Q4 Release Notes*.

How This Book Is Organized

Chapters 1 through 4 describe how to install, upgrade, configure, and uninstall the software.

Chapters 5 through 12 include some essential post-installation tasks.

Chapter 13 describes how to create a new portal and populate the portal with sample data.

Chapters 14 through 18 provides instructions for installing certain collaboration portlets on the Communities Sample.

Chapters 19 through 35 describes the `psadmin` command-line utility subcommands.

Chapter 36 discusses the new inter-portlet communication support.

Chapter 37 describes the administration tag library.

Chapters 38 and 39 describe file and operations format for the PAR subcommands.

Default Paths and File Names

The following table describes the default paths and file names used in this Release Note Supplement.

TABLE P–1 Default Paths and File Names

Term	Description
PortalServer-base	Represents the base installation directory for a previous version of Portal Server. The software default base installation and product directory depends on your specific platform: Solaris™ systems `/opt`
PortalServer7–base	Represents the base installation directory for this version of Portal Server. The software default base installation and product directory depends on your specific platform: Solaris™ systems `/opt`
AccessManager-base	Represents the base installation directory for Sun Java System Access Manager. The Access Manager default base installation and product directory depends on your specific platform: Solaris systems: `/opt/SUNWam`
DirectoryServer-base	Represents the base installation directory for Sun Java System Directory Server. Refer to the product documentation for the specific path name.
ApplicationServer-base	Represents the base installation directory for Sun Java System Application Server. Refer to the product documentation for the specific path name.
WebServer-base	Represents the base installation directory for Sun Java System Web Server, or BEA WebLogic Server 8.1 SP2. Refer to the product documentation for the specific path name.

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

Sun Function	URL	Description
Documentation	`http://www.sun.com/documentation/`	Download PDF and HTML documents, and order printed documents
Support and Training	`http://www.sun.com/supportraining/`	Obtain technical support, download patches, and learn about Sun courses

Typographic Conventions

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

TABLE P–2 Typographic Conventions

Typeface or Symbol	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*.

TABLE P–2 Typographic Conventions *(Continued)*

Typeface or Symbol	Meaning	Example
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. Perform a *patch analysis*. Do *not* save the file. [Note that some emphasized items appear bold online.]

Shell Prompts in Command Examples

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

TABLE P–3 Shell Prompts

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

CHAPTER 1

Installing the Sun Java System Portal Server 7 2005Q4 Software

This supplement contains the following sections

Introduction

The Sun Java System Portal Server 7 2005Q4 software uses the Sun Java Enterprise System 4 install wizard to install the Portal Server software and associated components.

With this release, the Secure Remote Access component has to be installed along portal in the same session. Some post-installation configuration is required for the Secure Remote Access component.

Pre-Installation Requirements

Hardware and Operating System Requirements

TABLE 1–1 Hardware and Operating System Requirements

Component	Platform Requirement
Supported Platforms	Sun Ultra™ 60 or Sun Blade comparable or better workstation or server
Operating System	Solaris™ 8 or Solaris™ 9 U6 or Solaris™ 10 on SPARC
	Solaris™ 9 or Solaris™ 10 on x86
	Red Hat Enterprise Linux 2.1 or 3.0 Update 3 on x86

TABLE 1–1 Hardware and Operating System Requirements *(Continued)*

Component	Platform Requirement
RAM	1024 Mbytes of RAM for evaluation install
	1.5 Gbytes of RAM for regular deployment on Sun Java System Web Server and BEA WebLogic Server
	2.0 Gbytes of RAM for regular deployment on Sun Java System Application Server
Disk Space	1 Gbyte of disk space for Portal Server and associated applications
Swap Space	The swap space of the machine should be twice of its physical memory. For example, in case of a 2.0 Gbytes RAM, the swap space should be 4.0 Gbytes.

Note – The Sun Java Enterprise System 4 Directory Server installation fails on Red Hat Enterprise Linux 3.0 Update 2 leading to Portal Server installation failure.

Software Requirements

The Portal Server software requires the following stack components:

- Sun Java System Directory Server 5.2 P4
- Sun Java System Access Manager 7 2005Q4 installed in legacy mode.
- Sun Java System Web Server 6.1 SP5 or Sun Java System Application Server 8.1 (including the patches) or BEA WebLogic Server 8.1 SP4. Note that third party web container must be installed and running prior to installing the Portal Server software.

The sun-soarsdk rpm (registry server sdk rpm) is not relocatable; so, do not choose non-default locations for registry on Linux.

For detailed instructions for installing the stack components, see the *Sun Java Enterprise System 2005Q4 Installation Reference*.

Before You Begin

This sections includes the following:

- "Miscellaneous Checks" on page 24
- "Installing On Linux" on page 25
- "Installing on an Application Server" on page 25

Miscellaneous Checks

1. If the install system does not have direct connectivity to the internet, an HTTP proxy needs to be specified. For example, for Sun Java System Application Server, specify the following in the `domain.xml` file:

```
<jvm-options>-Dhttp.proxyHost=Proxy-Host</jvm-options>
<jvm-options>-Dhttp.proxyPort=Proxy-Port</jvm-options>
<jvm-options>-Dhttp.nonProxyHosts="PortalServer-Host"</jvm-options>
```

Here, *Proxy-Host* is the fully-qualified domain name of the proxy host, *Proxy-Port* is the port on which the proxy is run, and *PortalServer-Host* is the fully-qualified domain name of the Portal Server software host.

2. Execute the command `prtconf | grep Memory` to check RAM.
3. Use the command `df -lk` to see how much swap space your machine has. To temporarily increase your swap space by 4 Gbytes, you can use the following instructions:

```
> mkfile 4g /fourGigXtraSwap
> swap -a /fourGigXtraSwap
```

Installing On Linux

- Remove the link `/usr/share/bdb/db.jar` before installation, if it exists.
- Check if a version of `ant` below 1.5.4 exists on the system by running the following command:

```
rpm -qa | grep ant
```

The `ant` version 1.5.2 interferes with portal configuration. If an earlier version of `ant` is found installed, remove it by running the following command:

```
rpm -e ant-1.5.2-23 ant-libs-1.5.2-23
```

Installing on an Application Server

If you are configuring the Sun Java System Application Server or BEA WebLogic Server for session failover (see *Sun Java System Application Server Enterprise Edition 8.1 2005Q2 High Availability Administration Guide* and Replication and Failover for Servlets and JSPs respectively), note that the Portal Server software must be installed on a non-default server. To install on a non-default server, see instructions below.

Installing and Verifying the Installation of the Portal Server Software

This section contains the following:

▼ To Install the Portal Server Software

Steps

1. **Go to the *PS_Install_Image*`/orion/`*OS-arch* directory, where *OS-arch* can be `Solaris_sparc`, or `Solaris_x86`, or `Linux_x86`.**

2. **Type `./installer` to invoke the wizard to install the software.**

3. **Select Next (at the Welcome Screen), and Accept License.**

4. **Select the language support you want to install and the components you wish to install on this system.**

   To install Portal Server software, select Sun Java System Portal Server 7 2005Q4.

5. **Specify the installation directory for the following software.**

Directory Preparation Tool	By default, this is installed in `/opt/SUNWcomds`
Access Manager	By default, this is installed in `/opt`
Web Server	By default, this is installed in `/opt/SUNWwbsvr`
Portal Server	By default, this is installed in `/opt` You will be prompted for the installation directory for Portal Server Secure Remote Access instead of Portal Server only (bug id 6333347). Note that Secure Remote Access is installed in the *PortalServer7–base* directory.

6. **Specify whether or not you wish to configure now.**

   The installer only supports adding one portal and one instance; for any other configuration, the configure later option must be selected. If the configure now option is selected, after the packages are installed, the configuration is immediately started; otherwise, the configuration can be done by selecting the configure later option. See supplement Chapter 3 for more information.

   If you selected configure now option, proceed to the next step; otherwise, skip to step 24.

7. **Specify the following common server settings:**

   Host Name, DNS Domain Name, Host IP Address
   Host name, domain, and IP address of the system

   Administrator User ID and Password
   User ID and password of the administrator

   System User and Group
   System user name and group ID

Note that the values you enter here will appear as default values during the rest of the installation.

8. **Specify the Web Server settings for administration and Default Web Server Instance.**

   For more information, see the "Web Server Configuration Information" in *Sun Java Enterprise System 2005Q4 Installation Reference* in the *Sun Java Enterprise System 2005Q4 Installation Reference*.

9. **Specify the Directory Server administration settings, server settings, configuration information, data storage location, populate data information.**

   For more information, see the "Directory Server Configuration Information" in *Sun Java Enterprise System 2005Q4 Installation Reference* in the *Sun Java Enterprise System 2005Q4 Installation Reference*.

10. **Specify the Access Manager administration information, web container information, services information, directory server information, and provisioned directory information.**

    For more information, see the "Access Manager Configuration Information" in *Sun Java Enterprise System 2005Q4 Installation Reference* in the *Sun Java Enterprise System 2005Q4 Installation Reference*.

11. **Select a web container for Portal Server software.**

    You can select one of the following:

    - Sun Java System Web Server
    - Sun Java System Application Server
    - BEA WebLogic Server

      If you are installing on a third party container, after installing the web container, install Directory Server, Access Manager, Portal Server, and Sun Java System Web Server (for the Portal Server administration console) in configure later mode and execute the Directory Server configurator, Access Manager configurator (`amconfig`), and Portal Server configurator (`psconfig`) with the correct file.

12. **Specify the web container information.**

    Note that based on your selection in Step 8, the information you provide will vary. For:

    - Sun Java System Web Server, specify the following:

Installation Directory	By default, this is installed in `/opt/SUNWwbsvr`.
Server Instance and Server Instance Port	By default, the server instance is the fully qualified host name of the system and 80 is the instance port.

Server Document Root	By default, `/opt/SUNWwbsvr/https-`*hostname.domain*`/docs` is the document root.
Secure Server Instance Port	Port number of the secure server instance

- Sun Java System Application Server, specify the following:

Installation Directory	By default, this is `/opt/SUNWappserver/appserver`.
Domain Name	By default, `domain1`.
Server Instance Directory and Port	By default, the instance directory is `/var/opt/SUNWappserver/domains/domain1` and port is 8080.
Document Root Directory	By default, `/var/opt/SUNWappserver/domains/domain1/docroot`.
Administration Port	By default, 4849.
Administrator User ID and Password	By default, `admin`.
Secure Server Instance Port	Port number of the secure server instance
Secure Administration Server Port	Port number of the secure administration server

- BEA WebLogic Server, specify the following:

Installation Directory	By default, `/usr/local/bea/weblogic81`.
Instance Directory	By default, `/usr/local/bea/user_projects`.
JDK Home Directory	By default, `/usr/local/bea/jdk142_05`.
Server/Cluster Host, Port, and Protocol	By default, the host is the hostname, Port is 7001, and protocol can be HTTP or HTTPS.
Administrator User ID and Password	By default, user ID is `weblogic`.
Administrator Host, Port, and Protocol	By default, host is the hostname, port is 7001, and protocol can be HTTP or HTTPS.
Managed Servers	WebLogic supports two type of servers: Admin Server and Managed Server. Installation on an

Admin Server instance requires the Admin Server to be up and running. To install on a Managed Server Instance:

- The Admin Server and the Managed Server must be running.
- The Node manager must be created and running and the Managed Server must be assigned to the Node manager.

13. Specify the following to deploy in to the web container:

Portal Access URL	By default, this is *protocol*`://`*hostname*`.`*domain*`:`*port*`/portal`.
Portal ID	By default, this is `portal1`.
Search ID	By default, this is `search1`.
Deployment URI	By default, this is `/portal`.

14. Specify whether or not you wish to configure all sample portals.

This version of the portal server includes three sample portals. For more information on these, see *Sun Java System Portal Server 7 2005Q4 Release Notes*. You can select one or more sample portals to configure.

15. To install Secure Remote Access also, specify the following; otherwise, skip to step 24.

Protocol	This can be HTTP or HTTPS.
Host and Port	By default, host is the *hostname*`.`*domain* and port is 80.
Deployment URI	By default, `/portal`.

16. Specify the following information to install the Gateway:

Gateway Protocol
By default, this is HTTPS.

Portal Server Domain
By default, this is the domain name of the system.

Gateway Domain and Port
By default, the domain is the default domain of the system and port is 443.

Gateway Profile Name
By default, this is `default`.

Log User Password
Log user password

Host Name, Subdomain, and Domain
By default, this is the host name, sub domain, and domain of the system on which you are installing the Gateway.

Host IP Address and Access Port
By default, this is the IP address of the system on which you are installing the Gateway and the port is, by default, 443.

Gateway Profile Name and Log User Password
By default, the gateway profile name is `default`.

17. Specify whether or not you wish to start Gateway after installation.

18. Specify the following information for the Netlet Proxy.

Host Name, Subdomain, and Domain
By default, the system values are used.

Host IP Address and Access Port
By default, the IP address is the IP address of the system and port is 10555

Gateway Profile Name and Log User Password
By default, the gateway profile name is `default`.

19. Specify whether or not you wish to start Netlet Proxy after installation.

20. Specify the following information to install the Rewriter Proxy.

Host Name, Subdomain, and Domain
By default, the system values are used.

Host IP Address and Access Port
By default, the IP address is the IP address of the system and port is 10443.

Gateway Profile Name and Log User Password
By default, the gateway profile name is `default`.

21. Specify whether or not you wish to start Rewriter Proxy after installation.

22. Specify the following proxy information for the Secure Remote Access software.

Note – Based on whether or not you wish to work with Portal Server software on another host, the ability to edit this page will vary. If the proxy server being installed are to work with an instance of Portal Server installed on a different host, select the option to Work with Portal Server on another host, and specify the following information.

Portal Server Protocol, Host, Port, and Deployment URI
By default, host is the *hostname*.*domain* of the system where Portal Server is installed, port is 81, and URI is `/portal`.

Organization DN
Organization distinguished node.

Access Manager Service URI and Encryption Key
By default, the URI is `/amserver`.

23. Specify the following certificate information for the Secure Remote Access software:

Organization, Division, City/Locality, State/Province
Specify your organization name, division, city, and state information.

Country Code
Use the two character format.

Certificate Database Password
This must be at least eight characters.

24. Specify whether or not you are ready to install by selecting the Next.

25. Specify whether or not you wish to open the registration window during installation and select Install to install the software.

▼ Completing Installation on BEA WebLogic Server

If you installed the Portal Server software on BEA WebLogic Server, you must install the Portal Server administration console (`psconsole`) on Sun Java System Web Server to administer the Portal Server via the console. Also, you must manually deploy WAR files in `/var/opt/SUNWportal/portals/`*portal-ID*`/war` directory. You must perform the tasks in this section on the host where the Portal Server software is installed.

Steps

1. Deploy the WAR files in `/var/opt/SUNWportal/portals/`*portal-ID*`/war` directory. For example, type:

```
./usr/local/bea/jdk142_05/jre/bin/java -classpath /usr/local/bea/weblogic81/server/lib/we
./usr/local/bea/jdk142_05/jre/bin/java -classpath /usr/local/bea/weblogic81/server/lib/we
```

- **To deploy the WAR files in `/var/opt/SUNWportal/portals/`*portal-ID*`/war` directory through the WebLogic administration console:**

  a. Log in to the WebLogic administration console and select the Admin Server from the domain it was created.

  b. Select the Web Applications Modules and click on Deploy a new Web Application Module.

  c. Follow the instruction to choose the war file on the file system of the Admin Server machine and deploy them.

2. **Restart the WebLogic Server after completing the configuration.**
3. **Install the Sun Java System Web Server software.**
4. **Deploy the `psconsole.war` file using the web server `wdeploy` command.**

▼ To Install on a Second Machine

The GUI installer cannot be used for multi portal installation. Follow the instructions in this section to install Portal Server on a second machine.

Steps

1. **Install Access Manager SDK and Web Server first.**
2. **Start the web container and then invoke the Portal Server software GUI installer to install the software in configure later mode.**
3. **Copy the `/etc/opt/SUNWcacao/security` directory from the first machine to the second machine and then start the CACAO server.**
4. **Complete the Portal Server installation by running the `psconfig --–config` command.**

▼ Post-Installation Configuration for Secure Remote Access

Steps

1. **Specify the complete protocol and fully qualified domain name for Non Authenticated URL list in *PortalServer7–base*`/export/request/enableSRAForPortal.xml` file using the following `amadmin` command:**

   ```
   ./amadmin --runasdn ADMIN_DN --password ampassword --verbose
   --continue --data file
   ```

2. **Log in to the Portal Server administration console and navigate to Secure Remote Access Rewrite Rulesets.**
3. **Execute the following commands if you do not see the `default_gateway_ruleset`, `exchange_2000sp3_owa_ruleset`, `exchange_2003_owa_ruleset`, `inotes_ruleset`, `iplanet_mail_ruleset`, `sap_portal_ruleset` and `wml_ruleset` entries.**

   ```
   ./psadmin create-rewriter-ruleset -u amadmin -f /tmp/passwd -F PortalServer7-base/SUNWportal/export/
   ./psadmin create-rewriter-ruleset -u amadmin -f /tmp/passwd -F PortalServer7-base/SUNWportal/export/
   ./psadmin create-rewriter-ruleset -u amadmin -f /tmp/passwd -F PortalServer7-base/SUNWportal/export/
   ./psadmin create-rewriter-ruleset -u amadmin -f /tmp/passwd -F PortalServer7-base/SUNWportal/export/
   ```

```
./psadmin create-rewriter-ruleset -u amadmin -f /tmp/passwd -F PortalServer7-base/SUNWportal/e
./psadmin create-rewriter-ruleset -u amadmin -f /tmp/passwd -F PortalServer7-base/SUNWportal/e
./psadmin create-rewriter-ruleset -u amadmin -f /tmp/passwd -F PortalServer7-base/SUNWportal/e
```

4. **Do the following if Gateway is configured:**

```
cd /etc/opt/SUNWportal/default
chmod -R 755 *
```

5. **To enable access to the Portal Server via the Gateway, see supplement 4 Enabling Access to the Portal Server Via the Gateway.**

▼ To Verify the Installation

Verify the Portal Server software installation by:

- Accessing the of the sample portals.
- Accessing the Portal Server software administration console.
- (Optional) Verifying the Gateway port and running the Portal Server in secure mode.

Steps

1. **Type** *protocol*`://`*fully-qualified-hostname*`:`*port*`/`*portal-URI* **in the browser.**

   When you type the URL, the welcome page, a short description of Portal server and links to sample portals that you selected for installation is displayed. Click on one of the links and access the anonymous portal desktop for the sample portal. If the sample Portal desktop displays without any exception, then your Portal Server installation was successful.

2. **Type** *protocol*`://`*fully-qualified-hostname*`:`*port*`/psconsole` **in the browser.**

3. **Run the following command to check if the gateway is running on the specified port (the default port is 443):**

   ```
   netstat -an | grep port-number
   ```

   If the gateway is not running, use the following command to start the gateway:

   PortalServer7-base`/SUNWportal/bin/psadmin start-sra-instance -u amadmin -f` *amadmin-password-file* `--instance-type gateway --instance-name` *GatewayInstanceName*

Note – Create a file and add `amadmin` password in plaint text and pass it as an input to the `-f` option above.

Also view the log files. The log file name is picked up from the property called `debug.com.sun.portal.handler.java.util.logging.FileHandler.pattern` in the `platform.conf` file.

4. **Run the Portal Server in secure mode by typing the gateway URL in your browser:**

   `https://`*gateway-machine-name*`:`*portnumber*

   If you have chosen the default port (443) during installation, you need not specify the port number.

CHAPTER 2

Upgrading to Sun Java System Portal Server 7 2005Q4

This supplement contains the following:

Pre-Upgrade Requirements

This section includes the following:

Hardware and Operating System Requirements

The following hardware and software are required to upgrade from Portal Server 6 2005Q1 software on Solaris to this release.

TABLE 2–1 Hardware and Operating System Requirements for Upgrade

Component	Platform Requirements
Supported Platforms	Sun Ultra™ 60, or Sun Blade, or better workstation or server
Operating System	Solaris 8, 9 U6, or 10
RAM	1024 Mbytes of RAM for evaluation install
	1.2 Gbytes of RAM for regular deployment
Disk space	1.5 Gbyte of disk space for Portal Server software and associated applications

Software Requirements

This section contains the following:

■ "Sun Java System Access Manager Software" on page 36

Note – The stack components have to be upgraded to their JES 4 versions prior to executing the `psupgrade` script.

Web Containers

Portal Server 7 2005Q4 software supports the following web containers:

Java Enterprise System Application Server 8.1
Java Enterprise System Web Server 6.1
BEA Weblogic Server 8.1 sp2
BEA Weblogic Server 8.1 sp4
IBM Websphere Server 5.1

Sun Java System Access Manager Software

Portal Server software can be upgraded on machines with:

Access Manager software previously installed and configured on the same physical machine using JES3.
Access Manager software previously installed and configured on a separate machine using JES3. The JES 3 Access Manager software SDK must be installed on the Portal Server software host.
JES 3 Access Manager software SDK installed on the machine were Portal Server Gateway is previously installed.

Before You Begin

Before upgrading to Portal Server 7 2005Q4 software:

1. For:
   - Web Server 6.1, in the `server.xml` file, replace `<JVMOPTIONS>-Xmx256m</JVMOPTIONS>` with:

     ```
     <JVMOPTIONS>-Xms512m -Xmx512m -Xss128k</JVMOPTIONS>
     <JVMOPTIONS>-XX:NewSize=256M</JVMOPTIONS>
     <JVMOPTIONS>-XX:MaxNewSize=256M</JVMOPTIONS>
     <JVMOPTIONS>-XX:PermSize=64M</JVMOPTIONS>
     <JVMOPTIONS>-XX:MaxPermSize=256M</JVMOPTIONS>
     ```

   - Application Server 8.1, in the `domain.xml` file, replace `<jvm-options>-Xmx512m</jvm-options>` with:

     ```
     <jvm-options>-Xms1024m</jvm-options>
     <jvm-options>-Xmx1024m</jvm-options>
     <jvm-options>-Xss128k</jvm-options>
     <jvm-options>-XX:NewSize=256M</jvm-options>
     ```

```
<jvm-options>-XX:MaxNewSize=256M</jvm-options>
<jvm-options>-XX:PermSize=64M</jvm-options>
<jvm-options>-XX:MaxPermSize=256M</jvm-options>
```

2. Restart the web container.
3. Check the version of the following Solaris packages on your system:

```
SUNWjaxp
SUNWxrgrt
SUNWxrpcrt
SUNWxsrt
SUNWaclg
```

   If the version is 7.x. remove them manually. For example, type **`pkgrm`** *SUNWpkg*.

4. Install the `SUNWsoar-sdk` package on the machine with a previous version of the Portal Server software.
5. Install the patches 118671-03 and 119793-01 for Solaris on SPARC and x86.
6. Stop Gateway if Gateway was installed and running on this machine.

Instructions to Upgrade and Verify Upgrade

This section contains the following:

▼ To Upgrade to Portal Server 7 2005Q4 Software

Steps

1. **Go to the directory where you have downloaded the 2005Q4 software and unzip the `java_es_05Q4_portal_server_beta_sparc_solaris8.zip` file.**
2. **Type the following on the command line:**

   ```
   cd PS_Install_Image/portalserver/
   ./psinstall
   ```

   This will install this version of the Portal Server software.

3. **Go to** *PortalServer7-base***`/portal_svr/Tools/upgrade/bin` directory and type `./psupgrade`.**

   The upgrade script will require you to provide the following:

   - Access Manager server administrator's password.
   - Access Manager `ldapuser` password
   - Directory Server Directory Manager password.

- Web container administrator's password.
- Web container Master Password in case of Application Server 8.1 for Portal Server software installation.
- Web container certificate database password if Secure Remote Access was previously installed on this machine.
- Secure Remote Access log user password if Secure Remote Access or its constituents were previously installed on this machine.

When you upgrade, the upgraded Portal Server installation will be at *PortalServer7-base*/`SUNWportal` directory.

4. **Install theAccess Manager software patches, 119465-02 and 119466-02 (for Solaris on SPARC and x86), on the machine where Access Manager is installed and restart the Access Manager server.**
5. **Restart the Portal Server web container and Gateway (if Gateway was installed and upgraded).**

▼ To Ensure Upgrade to Portal Server 7 2005Q4 was Successful

Steps

1. **Access the Access Manager software administration console from your browser. To access, type `http://`*hostname*`/amconsole`.**
2. **Access the Portal Server Desktop. To access, type** *protocol*`://`*fully-qualified-hostname*`:`*port*`/`*portal-URI*.
   If the sample Portal desktop displays without any exception, then your upgrade was successful. Try logging in as a user to ensure that the sample Portal desktop displays without errors.
3. **Access the Portal Server software administration console. To access, type** *protocol*`://`*fully-qualified-hostname*`:`*port*`/psconsole`.
   Verify that it displays information about Portal Server7 2005Q4 software. Ensure that a Portal with portal-id `Upgraded` is created.

CHAPTER 3

Configuring After the Installation

This supplement contains the following:

Overview

The Sun Java System Portal Server 7 2005Q4 software can be installed using the installer in one of the two modes: the `config now` mode, where installation and configuration take place simultaneously, or the `config later` mode, which requires you to run the *PortalServer7-base*`/bin/psconfig --config` *config-xml-file* command after installing the software.

This supplement describes how to use the sample configuration XML file or construct a configuration XML file for the desired portal set up. A basic understanding of the structure of XML is required to construct a custom configuration file.

Using the Sample Configuration XML File

The Sun Java System Portal Server software includes twenty sample configuration XML files at:

`/opt/SUNWportal/samples/psconfig` directory for SPARC and x86
`/opt/sun/portal/samples/psconfig` directory for Linux

Note – If portal is installed in a non-default location, these locations will relatively change.

The *PortalServer7-base*/SUNWportal/samples/psconfig/ReadMe.txt describe each example file. Read through this file to see which configuration example best suits your set up and replace the @TAGS@ (marked by @. . .@) after reviewing the default values specified in the example file. Create the required configuration XML file for the desired portal setup by modifying a selected configuration example.

Any of the configuration examples for the Web Server container can be adapted for Sun Java System Application Server or BEA WebLogic Application Server by replacing the <WebContainerProperties> element section and the @TAGS@ tokens after reviewing the default values. For example:

Examples 1, 3 to 9, and 13 are common configurations for the Sun Java System Web Server container.
Example 14 is a configuration for the Sun Java System Application Server 8.1 container
Example 15 is a configuration for the BEA WebLogic container

Multi portal configurations (see example 15) can be customized by:

1. Including multiples instances of <PortalServer>, <Instance>, and <SearchServer> elements.
2. Replacing the @TAGS@ tokens after reviewing the default values.

Constructing a Configuration XML File

If the sample configuration file does not suit your desired setup and if a custom configuration XML file is to be constructed, follow the instructions in this section. In order to set up your custom configuration file, you must:

1. Begin by constructing the "Required Configuration" on page 41.

   This basic configuration is required to make the portal psadmin command useable.
2. Construct the <ComponentsToConfigure> element depending on which components are to be configured on this host. See "Portal Server Configuration" on page 44 for more information.
3. Construct the following configuration information based on the components to configure on this host:

   "Basic Portal Configuration" on page 44
   "Sample Portal Configuration" on page 45
   "Web Container Configuration" on page 45
   "Search Server Configuration" on page 49
   "Secure Remote Access Configuration" on page 50
   "Gateway Configuration" on page 50
   "Netlet Proxy Configuration" on page 51
   "Rewriter Proxy Configuration" on page 52
4. Run the ./psconfig --config *configfile.xml* command.

Required Configuration

This section describes the overall Portal Server, header/footer, shared components, and the Access Manager elements in the configuration file. See `example2.xml` file.

For Solaris on SPARC and x86

```
<?xml version = "1.0" encoding = "UTF-8"?>
<PortalServerConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="file:///opt/SUNWportal/lib/psconfig.xsd" SchemaVersion="1
        <Configure ConfigurationHostName="@HOST.DOMAIN@" SystemUser="root" SystemGroup="othe
            <SharedComponents
                JavaHome="/usr/jdk/entsys-j2se"
                CacaoProdDir="/opt/SUNWcacao"
                CacaoConfigDir="/etc/opt/SUNWcacao"
                SharedLibDir="/usr/share/lib"
                PrivateLibDir="/usr/share/lib"
                JDMKLibDir="/opt/SUNWjdmk/5.1/lib"
                NSSLibDir="/usr/lib/mps/secv1"
                JSSJarDir="/usr/share/lib/mps/secv1"
                WebNFSLibDir="/opt/SUNWebnfs"
                DerbyLibDir="/usr/share/lib/Derby"
                JCIFSLibDir=""
                JChardet=""
                AntLibDir="/usr/sfw/lib/ant"
                AntHomeDir="/usr/sfw"
         RegistryLibDir="/opt/SUNWsoar/lib"
            />
            <AccessManager>
                <InstallationDirectory
                    ProdDir="/opt/SUNWam"
                    DataDir="/var/opt/SUNWam"
                    ConfigDir="/etc/opt/SUNWam/config"
                    ConfigFile="AMConfig.properties"
                />
                <UserCredentials
                    AdministratorUID="amadmin"
                    AdministratorUserPassword="@AMADMIN.PASSWORD@"
                    LDAPUserId="amldapuser"
                    LDAPUserIdPassword="@AMLDAPUSER.PASSWORD@"
                    DirectoryManagerDn="cn=Directory Manager"
                DirectoryManagerPassword="@DIRMGR.PASSWORD@"/>
            </AccessManager>
            <PortalConfiguration>
                <InstallationDirectory
                    ProdDir="/opt/SUNWportal"
                    DataDir="/var/opt/SUNWportal"
                ConfigDir="/etc/opt/SUNWportal"/>
                <ComponentsToConfigure>
                    .
                    .
                    .
                </ComponentsToConfigure>
                .
                .
```

```
                .
            </PortalConfiguration>
            .
            .
            .
        </Configure>
</PortalServerConfiguration>
```

For Linux

```
<?xml version = "1.0" encoding = "UTF-8"?>
<PortalServerConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="file:///opt/sun/portal/lib/psconfig.xsd" SchemaVersion="1.0">
        <Configure  ConfigurationHostName="@HOST.DOMAIN@" SystemUser="root" SystemGroup="other" Va
            <SharedComponents
                JavaHome="/usr/jdk/entsys-j2se"
                CacaoProdDir="/opt/sun/cacao"
                CacaoConfigDir="/etc/opt/sun/cacao"
                SharedLibDir="/opt/sun/share/lib"
                PrivateLibDir="/opt/sun/private/share/lib"
                JDMKLibDir="/opt/sun/jdmk/5.1/lib"
                NSSLibDir="/opt/sun/private/lib"
                JSSJarDir="/opt/sun/private/share/lib"
                WebNFSLibDir="/opt/sun/webnfs"
                DerbyLibDir="/opt/sun/share/lib/Derby"
                JCIFSLibDir=""
                JChardet=""
                AntHomeDir="/opt/sun/share"
                AntLibDir="/opt/sun/share/lib"
             RegistryLibDir="/opt/sun/SUNWsoar/lib"
                />
                <AccessManager>
                    <InstallationDirectory
                        ProdDir="/opt/sun/identity"
                        DataDir="/var/opt/sun/identity"
                        ConfigDir="/etc/opt/sun/identity/config"
                        ConfigFile="AMConfig.properties"
                    />
                    <UserCredentials
                        AdministratorUID="amadmin"
                        AdministratorUserPassword="@AMADMIN.PASSWORD@"
                        LDAPUserId="amldapuser"
                        LDAPUserIdPassword="@AMLDAPUSER.PASSWORD@"
                        DirectoryManagerDn="cn=Directory Manager"
                    DirectoryManagerPassword="@DIRMGR.PASSWORD@"/>
                </AccessManager>
                <PortalConfiguration>
                    <InstallationDirectory
                        ProdDir="/opt/sun/portal"
                        DataDir="/var/opt/sun/portal"
                    ConfigDir="/etc/opt/sun/portal"/>
                    <ComponentsToConfigure>
                        .
                        .
                        .
```

```
                </ComponentsToConfigure>
                .
                .
                .
            </PortalConfiguration>
            .
            .
            .
        </Configure>
</PortalServerConfiguration>
```

Tokens to Replace

`@HOST.DOMAIN@`	The host and domain name of the machine on which configuration is occurring.
`@AMADMIN.PASSWORD@`	Administrator's password for the Access Manager instance with which Portal is to be configured.
`@AMLDAPUSER.PASSWORD@`	Internal LDAP User password for the Access Manager instance with which Portal is to be configured.
`@DIRMGR.PASSWORD@`	Administrator's password of the Directory Server with which Portal is to be configured.

Values to Modify

`PortalServerConfiguration xsi:noNamespaceSchemaLocation`
If portal is installed in a non-default location, then this location needs to be changed accordingly.

`SharedComponents JCIFSLibDir`
JCIFS is an optional 3rd party component that is required only by the Netfile component. Install the JCIFS package and specify the `lib` location here.

`SharedComponents JChardet`
JChardet is an optional 3rd party component that is required only by the Netfile component. Install the JChardet package and specify the `lib` location here.

`AccessManager InstallationDirectory ProdDir, DataDir, ConfigDir`
Specify the installation location of Access Manager software here if it was not installed in the default location.

`PortalConfiguration InstallationDirectory ProdDir, DataDir, ConfigDir`
Specify the installation location of Portal Server software here if it was not installed in the default location.

Portal Server Configuration

Different Portal Server components that can be installed and configured across different nodes include the core Portal Server, Secure Remote Access (SRA), Gateway, Netlet Proxy, and Rewriter Proxy. Depending on which components are configured on this host, the `<ComponentsToConfigure>` element can be constructed.

If all components are to be configured this host, include the following:

```
<ComponentsToConfigure>
        <component>portalserver</component>
        <component>sracore</component>
        <component>gateway</component>
        <component>netletproxy</component>
        <component>rewriterproxy</component>
</ComponentsToConfigure>
```

To exclude components, remove the corresponding `<component>` element.

Basic Portal Configuration

XML Fragment

```
<PortalConfiguration>
        <PortalServer PortalAccessURL="http://@HOST.DOMAIN@:@PORT@/portal"
            PortalID="myPortal"
            PortalWebappURI="/portal"
            SearchServerID="mySearch">
            <Instance InstanceID="myInstance">
                <WebContainerProperties
                    .
                    .
                    .
                />
            </Instance>
        </PortalServer>
</PortalConfiguration>
```

Tokens to Replace

`@HOST.DOMAIN@`	The host and domain name of the machine on which portal is to be configured.
`@PORT@`	Web container port at which portal has to be deployed

Values to Modify

`PortalConfiguration PortalServer PortalAccessURL` (optional)
If the DEPLOY URI is non-default, change `/portal` to the changed URI value.

`PortalConfiguration PortalServer PortalWebappURI` (optional)
If the DEPLOY URI is non-default, change, `/portal` to the changed URI value. In case of non-default DEPLOY URI, ensure that both `PortalAccessURL` and `PortalWebappURI` are specified in the configuration XML file.

`PortalConfiguration PortalServer PortalID`
Change `myPortal` to the required portal ID, which should be unique.

`PortalConfiguration PortalServer Instance InstanceID`
Change `myInstance` to the required instance ID, which should be unique.

`PortalConfiguration PortalServer SearchServerID` (optional)
Specifies which Search Server this portal samples are configured with. This is needed only if samples are configured.

Sample Portal Configuration

Portal Server software supports three types of sample portals: the Developer Sample, Enterprise Sample, and Community Sample. Each of these samples are created under its own sub-org for ease of management. Configuring any or all of these samples is supported.

```
<PortalConfiguration>
        <PortalServer
            .
            .
            .
        >
            <SamplePortal>
                <Sample Name="DeveloperPortal"/>
                <Sample Name="EnterprisePortal"/>
                <Sample Name="CommunityPortal"/>
            </SamplePortal>
            .
            .
            .
        </PortalServer>
</PortalConfiguration>
```

Web Container Configuration

The Web container configuration varies with the container to be configured. In the configuration XML file, there is one `<WebContainerProperties>` element specified for the web container under the `<PortalServer><Instance>` element and one under the `<SearchServer>` element.

Sun Java System Web Server Configuration

Tip – See `example1.xml`, examples 3 to 9, and `example13.xml` files.

XML Fragment For Solaris on SPARC and x86

```
<WebContainerProperties
        Host="@HOST.DOMAIN@"
        Port="@PORT@"
        Scheme="http"
```

```
        WebContainerInstallDir="/opt/SUNWwbsvr"
        WebContainerInstanceName="@INSTANCENAME@"
        WebContainerInstanceDir="/opt/SUNWwbsvr/https-@INSTANCENAME@"
        WebContainerDocRoot="/opt/SUNWwbsvr/docs"
        WebContainerAdminHost=""
        WebContainerAdminPort="@ADMIN.PORT@"
        WebContainerAdminScheme="http"
        WebContainerAdminUid="admin"
        WebContainerAdminPassword="@PASSWORD@"
        WebContainerCertificateDBPassword=""
        WebContainerType="SJSWS6"
    />
```

For Linux

```
<WebContainerProperties
        Host="@HOST.DOMAIN@"
        Port="@PORT@"
        Scheme="http"
        WebContainerInstallDir="/opt/sun/webserver"
        WebContainerInstanceName="@INSTANCENAME@"
        WebContainerInstanceDir="/opt/sun/webserver/https-@INSTANCENAME@"
        WebContainerDocRoot="/opt/sun/webserver/docs"
        WebContainerAdminHost=""
        WebContainerAdminPort="@ADMIN.PORT@"
        WebContainerAdminScheme="http"
        WebContainerAdminUid="admin"
        WebContainerAdminPassword="@PASSWORD@"
        WebContainerCertificateDBPassword=""
        WebContainerType="SJSWS6"
    />
```

Tokens to Replace

`@HOST.DOMAIN@`	The host and domain name of the machine on which portal instance is to be configured
`@PORT@`	Web server port
`@INSTANCENAME@`	Web server instance name
`@ADMIN.PORT@`	Web server administration port
`@PASSWORD@`	Web server administrator's password

Values to Modify

`WebContainerInstallDir, WebContainerInstanceDir, WebContainerDocRoot`
If the web server is installed in a non-default location.

`WebContainerAdminScheme, WebContainerCertificateDBPassword`
If web server is installed in secure mode (https).

Tip – See `example14.xml` file.

XML Fragment For Solaris on SPARC and x86

```
<WebContainerProperties
        Host="@HOST.DOMAIN@"
        Port="@PORT@"
        Scheme="http"
        WebContainerInstallDir="/opt/SUNWappserver/appserver"
        WebContainerInstanceName="server"
        WebContainerDomainName="domain1"
        WebContainerInstanceDir="/var/opt/SUNWappserver/domains/domain1"
        WebContainerDocRoot="/var/opt/SUNWappserver/domains/domain1/docroot"
        WebContainerAdminHost="@HOST.DOMAIN@"
        WebContainerAdminPort="@ADMIN.PORT@"
        WebContainerAdminScheme="https"
        WebContainerAdminUid="admin"
        WebContainerAdminPassword="@PASSWORD@"
        WebContainerMasterPassword="@MASTER.PASSWORD@"
        WebContainerType="SJSAS81"
    />
```

XML Fragment For Linux

```
<WebContainerProperties
        Host="@HOST.DOMAIN@"
        Port="@PORT@"
        Scheme="http"
        WebContainerInstallDir="/opt/sun/appserver"
        WebContainerInstanceName="server"
        WebContainerDomainName="domain1"
        WebContainerInstanceDir="/var/opt/sun/appserver/domains/domain1"
        WebContainerDocRoot="/var/opt/sun/appserver/domains/domain1/docroot"
        WebContainerAdminHost="@HOST.DOMAIN@"
        WebContainerAdminPort="@ADMIN.PORT@"
        WebContainerAdminScheme="https"
        WebContainerAdminUid="admin"
        WebContainerAdminPassword="@PASSWORD@"
        WebContainerMasterPassword="@MASTER.PASSWORD@"
        WebContainerType="SJSAS81"
    />
```

Tokens to Replace

`@HOST.DOMAIN@`	The host and domain name of the machine on which portal instance is to be configured
`@PORT@`	Application server port
`@ADMIN.PORT@`	Application server administration port
`@PASSWORD@`	Application server administrator's password

`@MASTER.PASSWORD@` Application server Master Password if specified

Values to Modify

`WebContainerInstallDir`
If application server is installed at a non-default location

`WebContainerDomainName, WebContainerInstanceDir, WebContainerDocRoot`
If deploying to a non-default application server domain

`WebContainerInstanceName`
Instance name within the Application Server domain. The `server` is the name of the first instance which is created by default at the same time the Application Server 8.1 domain is created. This can be changed to the name of any other created instance within that domain.

BEA WebLogic Configuration

Tip – See `example15.xml` file.

XML Fragment for Solaris on SPARC and x86

```
<WebContainerProperties
        Host="@HOST.DOMAIN@"
        Port="@PORT@"
        Scheme="http"
        WebContainerInstallDir="/usr/local/bea/weblogic81"
        WebContainerInstanceName="myserver"
        WebContainerInstanceDir="/usr/local/bea/user_projects/domains/mydomain"
        WebContainerDocRoot=""
        WebContainerAdminHost="@HOST.DOMAIN@"
        WebContainerAdminPort="@ADMIN.PORT@"
        WebContainerAdminScheme="http"
        WebContainerAdminUid="weblogic"
        WebContainerAdminPassword="@PASSWORD@"
        WebContainerJDKDir="/usr/local/bea/jdk141_05"
        WebContainerManagedServer="false"
        WebContainerType="BEAWL8"
    />
```

XML Fragment for Linux

```
<WebContainerProperties
        Host="@HOST.DOMAIN@"
        Port="@PORT@"
        Scheme="http"
        WebContainerInstallDir="/root/bea/weblogic81"
        WebContainerInstanceName="myserver"
        WebContainerInstanceDir="/root/bea/user_projects/domains/mydomain"
        WebContainerDocRoot=""
        WebContainerAdminHost="@HOST.DOMAIN@"
```

```
        WebContainerAdminPort="@ADMIN.PORT@"
        WebContainerAdminScheme="http"
        WebContainerAdminUid="weblogic"
        WebContainerAdminPassword="@PASSWORD@"
        WebContainerJDKDir="/root/bea/jdk141_05"
        WebContainerManagedServer="false"
        WebContainerType="BEAWL8"
    />
```

Tokens to Replace

`@HOST.DOMAIN@`	The host and domain name of the machine on which portal instance is to be configured
`@PORT@`	Application server port
`@ADMIN.PORT@`	Application server administration port
`@PASSWORD@`	Application server administrator's password

Values to Modify

`WebContainerInstallDir, WebContainerInstanceDir, WebContainerJDKDir`
If application server is installed at a non-default location

`WebContainerInstanceDir, WebContainerInstanceName`
If deploying to a non-default domain and server.

`WebContainerManagedServer`
If Managed Server option is true

Search Server Configuration

The Search Server is deployed to a specific web container instance which is defined by a `<WebContainerProperties>` element. Multiple Search servers can be specified by having multiple `<SearchServer>` elements within a `<PortalConfiguration>` section, each with a unique ID. A Portal may be associated with a specific search server by specifying the `SearchServerID` attribute within the `<PortalServer>` element.

```
<PortalConfiguration>
        <SearchServer SearchServerID="mySearch">
            <WebContainerProperties
                .
                .
                .
            />
        </SearchServer>
        <PortalServer
            SearchServerID="mySearch">
                .
                .
                .
        </PortalServer>
</PortalConfiguration>
```

Secure Remote Access Configuration

The SRA core component can only be installed and configured on the same node as the portal server component. Further the portal server and SRA core components have to be configured at the same time. That is, the SRA core component cannot be configured on a host that already has an existing portal server.

XML Fragment

Secure remote access support can be added to portal be adding the `<component>sracore</component>` to the `<ComponentsToConfigure>` section. In addition add the following section to the `<PortalConfiguration>` section:

```
<PortalConfiguration>
        .
        .
        .
        <SecureRemoteAccessCore
            GatewayProtocol="https"
            PortalServerDomain="@DOMAIN@"
            GatewayPort="@GATEWAY.PORT@"
            GatewayProfileName="default"
        LogUserPassword="@SRA.LOGUSER.PASSWORD@"/>
</PortalConfiguration>
```

Tokens to Replace

`@DOMAIN@`	Domain name of the machine on which portal is to be configured
`@GATEWAY.PORT@`	Port on which Gateway is to run
`@SRA.LOGUSER.PASSWORD@`	SRA log user password

Values to Modify

`GatewayProfileName`	Change this if the default profile is not to be used

Gateway Configuration

Tip – See `example10.xml` file.

XML Fragment

```
<ComponentsToConfigure>
        <component>gateway</component>
</ComponentsToConfigure>
<PortalServer PortalAccessURL="http://@PSHOST.DOMAIN@:@PORT@/portal">
</PortalServer>
<Gateway Profile="default">
        <SRAInstance
            Protocol="https"
```

```
            Host="@HOST.DOMAIN@"
            Port="@GATEWAY.PORT@"
            IPAddress="@IPADDRESS@"
            LogUserPassword="@SRA.LOGUSER.PASSWORD@"
        StartInstance="true"/>
</Gateway>
<CertificateInformation
        Organization="Sun Microsystems"
        Division="Software"
        CityOrLocality="Santa Clara"
        StateProvince="CA"
        CountryCode="US"
    CertificateDatabasePassword="@SRA.CERTDB.PASSWORD@"/>
```

Tokens to Replace

`@HOST.DOMAIN@`	The host and domain name of the machine on which gateway is to be configured
`@GATEWAY.PORT@`	Port on which Gateway is to run
`@IPADDRESS@`	IP address of the machine on which Gateway is run
`@PSHOST.DOMAIN@`	The host and domain name of the machine on which portal instance is to be configured.
`@PORT@`	Port on which portal instance is to run
`@SRA.LOGUSER.PASSWORD@`	SRA log user password
`@SRA.CERTDB.PASSWORD@`	SRA Certificate database password

Values to Modify

`Gateway Profile`
Change this if the default profile is not to be used

`Gateway SRAInstance StartInstance`
Change if start on install is not required

`CertificateInformation`
Change attributes in this section accordingly

Netlet Proxy Configuration

Tip – See `example11.xml` file.

XML Fragment

```
<ComponentsToConfigure>
        <component>netletproxycomponent>netletproxy>
</ComponentsToConfigure>
<PortalServer PortalAccessURL="http://@PSHOST.DOMAIN@:@PORT@/portal">
```

```
</PortalServer>
<NetletProxy Profile="default">
        <SRAInstance
            Protocol="https"
            Host="@HOST.DOMAIN@"
            Port="@NETLET.PROXY.PORT@"
            IPAddress="@IPADDRESS@"
            LogUserPassword="@SRA.LOGUSER.PASSWORD@"
        StartInstance="true"/>
</NetletProxy>
<CertificateInformation
        Organization="Sun Microsystems"
        Division="Software"
        CityOrLocality="Santa Clara"
        StateProvince="CA"
        CountryCode="US"
    CertificateDatabasePassword="@SRA.CERTDB.PASSWORD@"/>
```

Tokens to Replace

`@HOST.DOMAIN@`	The host and domain name of the machine on which Netlet Proxy is to be configured
`@NETLET.PROXY.PORT@`	Port on which Netlet Proxy is to run
`@IPADDRESS@`	IP address of the machine on which Netlet Proxy is to run
`@PSHOST.DOMAIN@`	The host and domain name of the machine on which portal instance is to be configured.
`@PORT@`	Port on which portal instance is to run
`@SRA.LOGUSER.PASSWORD@`	SRA log user password
`@SRA.CERTDB.PASSWORD@`	SRA Certificate database password

Values to Modify

`NetletProxy Profile`
Change this if the default profile is not to be used

`NetletProxy SRAInstance StartInstance`
Change if start on install is not required

`CertificateInformation`
Change attributes in this section accordingly

Rewriter Proxy Configuration

Tip – See `example12.xml` file.

XML Fragment

```
<ComponentsToConfigure>
        <component>rewriterproxycomponent>rewriterproxy>
</ComponentsToConfigure>
<PortalServer PortalAccessURL="http://@PSHOST.DOMAIN@:@PORT@/portal">
</PortalServer>
<RewriterProxy Profile="default">
        <SRAInstance
            Protocol="https"
            Host="@HOST.DOMAIN@"
            Port="@REWRITER.PROXY.PORT@"
            IPAddress="@IPADDRESS@"
            LogUserPassword="@SRA.LOGUSER.PASSWORD@"
        StartInstance="true"/>
</RewriterProxy>
<CertificateInformation
        Organization="Sun Microsystems"
        Division="Software"
        CityOrLocality="Santa Clara"
        StateProvince="CA"
        CountryCode="US"
    CertificateDatabasePassword="@SRA.CERTDB.PASSWORD@"/>
```

Tokens to Replace

`@HOST.DOMAIN@`	The host and domain name of the machine on which Rewriter Proxy is to be configured
`@REWRITER.PROXY.PORT@`	Port on which Rewriter Proxy is to run
`@IPADDRESS@`	IP address of the machine on which Rewriter Proxy is to run
`@PSHOST.DOMAIN@`	The host and domain name of the machine on which portal instance is to be configured.
`@PORT@`	Port on which portal instance is to run
`@SRA.LOGUSER.PASSWORD@`	SRA log user password
`@SRA.CERTDB.PASSWORD@`	SRA Certificate database password

Values to Modify

`RewriterProxy Profile`
Change this if the default profile is not to be used

`RewriterProxy SRAInstance StartInstance`
Change if start on install is not required

`CertificateInformation`
Change attributes in this section accordingly

CHAPTER 4

Un-installing Sun Java System Portal Server 7 2005Q4 Software

This supplement includes instructions for uninstalling the Portal Server software.

Uninstalling the Software

▼ To Uninstall the Portal Server Software

Steps

1. **Log in to the machine running the Portal Server software and become root.**
2. **Change directories to `/var/sadm/prod/SUNWps7_0-entsys/`.**
3. **Type `./uninstall` to uninstall the Portal Server software.**
   The Sun Java Enterprise System Uninstall Wizard is displayed.
4. **Select the components to uninstall and select Next.**
   If you are uninstalling the Secure Remote Access component, you will be asked to provide the portal administrator, Access Manager administrator, and LDAP passwords.
5. **Select Uninstall to uninstall the software.**

CHAPTER 5

Enabling Access to the Portal Server Via the Gateway

This supplement includes instructions to enable access to the Portal Server via the Gateway.

Enabling Access to the Portal Server

▼ To Enable Access to the Portal Server

Steps

1. **Modify the following tokens in the *PortalServer7-base*`/SUNWportal/export/request/enableSRAForPortal.xml` file to suit your deployment.**

   `%INST_GWNAME%`
   Gateway Profile you are modifying

   `%FULLY_QUALIFIED_PORTAL_SERVER_URI%`
   Fully qualified portal URL

   `%PORTAL_SERVER_DOMAIN%`
   Domain in which the portal server resides

   `%DEPLOY_URI%`
   Deploy URL for the portal web application

2. **Save the file after making the changes.**

3. **Load the file into the directory server using the Sun Java System Access Manager's `amadmin` command as follows:**

   AccessManager-base`/bin/amadmin -u` *amadmin* `-w` *amadmin-pwd* `-t enableSRAForPortal.xml`

CHAPTER 6

Release Notes Supplement

This supplement contains the following:

Pre-install steps for Mobile Access on JES 3

▼ To Install Mobile Access on JES 3

Perform the following pre-installation steps manually before installing Mobile Access packages. These steps must be performed before installing JES3.

On a machine without Java Enterprise Server or Portal Server 7.0 packages, replace some JES3 installation packages with Portal Server 7.0 installation packages.

Steps

1. **In the JES3 installer location, replace `SUNWamma` and `SUNWammae` packages with SUNWamma and SUNWammae packages from Portal Server 7.0 installer. The path to the folder is**
   JES3-bits-location`/Solaris_sparc/Product/identity_svr/Packages/.`

2. **Similarly, Replace JES3 `SUNWma` and `SUNWmae` packages with Portal Server 7.0 SUNWma and SUNWmae packages in the location**
   JES3-location`/Solaris_sparc/Product/shared_components/Packages/.`

   After performing the preceding manual replacement of packages, you can proceed with the installation of JES 3 components followed by Portal Server 7.0 installation and configuration.

Manual Steps to Enable Mobile Access in Portal Server 7.0

▼ To enable Mobile Access functionality in Portal Server 7.0

You must perform the following manual steps to enable Mobile Access functionality in Portal Server 7.0.

Steps

1. **In the Portal Server console, click Portals**
2. **Click portal1 in the list of portals.**
3. **In the Current Location drop-down menu, select DeveloperSample.**

Note – If the user creates own organization, then select the user-created organization.

---

4. **In the Desktop Attributes section, change the default value of Parent Container to WiresessDesktopDispatcher.**
5. **Ensure the value of Edit Container is set to JSPEditContainer.**
6. **Ensure the value of Desktop Type is set to developer_sample.**
7. **Click Save.**

Pre-install steps for Mobile Access on JES4

▼ To Install Mobile Access on JES4

You must perform the following pre-installation steps manually before installing Mobile Access packages. These steps must be performed before installing JES4. On a machine without any Java Enterprise Server or Portal Server 7.0 packages, replace some JES4 installation packages with Portal Server 7.0 installation packages.

Step

- **In the JES4 installer location, replace `SUNWamma` and `SUNWammae` packages with `SUNWamma` and `SUNWammae` packages from Portal Server 7.0 installer. The path to the folder is**
  JES4-location`/Solaris_sparc/Product/identity_svr/Packages/.`

  After performing the preceding manual replacement of packages, you can proceed with installation of JES4 components followed by Portal Server 7.0 installation and configuration.

Manual Steps to Add Mobile Comms Channels

▼ To manually add Mobile Comms Channels

You must perform the following steps to enable JSPComms Channels to work with Mobile Access application.

Steps

1. **From the Portal Server console, click Manage Channels and Containers.**
2. **In the pop-up window, select DeveloperSample from Select Directory drop-down menu.**
3. **From View Type, select JSPRenderingContainer or JSPNativeContainer.**
4. **From Container Task select MobileAddressBook, MobileCalendar, and MobileMail channels and click Add.**
5. **On the portal desktop, configure the Communications Express Address Book, Communications Express Calendar, and Communications Express Mail channels, which are available by default.**

   After you add the JSPRenderingContainer or JSPNativeContainer, they can be accessed from a mobile device.

CHAPTER 7

Installing Enterprise Sample Portal Communications Tab

This supplement contains the following:

Pre-Installation Requirements

The Sun Java System Portal Server Enterprise Sample Portal Communications tab setup is dependent on the Delegated Administrator software (a Sun Java Enterprise System supported mechanism for granting domain and user level mail and calendar privileges), specifically the `commadmin` command line utility.

The Portal Server host must contain the Sun Java Enterprise System 2005Q4 version of the following software installed and configured:

Sun Java System Access Manager
Sun Java System Web Server or Application Server
Sun Java System Delegated Administrator Utility

In addition, the following Sun Java Enterprise System 2005Q4 products must be installed and configured and may exist on a different host:

Sun Java System Directory Server
Sun Java System Messaging Server 6
Sun Java System Calendar Server 6
Sun Java System Message Queue 3 Enterprise Edition
Sun Java System Instant Messaging 7
Sun Java System Communications Express 6

The Communication Products (Messaging Server, Calendar Server, Delegated Administrator, Instant Messaging Server, and Communications Express) must be configured for:

- Access Manager Single Sign-On (SSO) support

- Hosted Domain support

Installing Enterprise Sample Portal Communications Tab

The Communications tab is installed under the My Portal tab in the Enterprise sample portal desktop.

▼ To Install the Communications Tab

Steps

1. **Copy the `input.properties.template`, `password.properties.template`, and `comms.properties.template` files in *PortalServer-base7*`/SUNWportal/samples/portals/shared` directory to a custom directory and remove the `.template` extension from the file names.**

2. **Edit the `input.properties` file.**

   Replace all the tokens that begin and end with `%` with the appropriate portal settings. When the portal is configured, the portal instance identifier must be substituted in the `input.properties` file in `/var/opt/SUNWportal/tmp` directory.

3. **Edit the `password.properties` file.**

   Replace all the tokens that begin and end with `%` with the appropriate administrator's password value.

4. **Edit the `comms.properties` file.**

   Replace all the tokens that begin and end with `%` with the appropriate Communication Server setting. Alternatively, if all Sun Java Enterprise System products have been installed on a single host, then there is a helper `ant` script in `/opt/SUNWportal/samples/portals/enterprise/singleHostGatherCommConfig.xml` file to gather the Communication Product's information and generate the `comms.properties` file. To run this script, type **`ant -buildfile`** *PortalServer7-base*`/SUNWportal/samples/portals/enterprise/singleHostGatherCommConfig.x` **`-Doutput.location=`***outputlocation* where *outputlocation* refers to the location of the generated `comms.properties` file.

5. **Run the `ant` script in `/opt/SUNWportal/samples/portals/enterprise/build.xml` file to install the communications tab. For example, type `ant config_comms -buildfile /opt/SUNWportal/samples/portals/enterprise/build.xml -Dconfig.location=`***configlocation* **`-logfile`** *logfile* **where** *configlocation* **refers to the location (directory) of the `password.properties` and `comms.properties` files and** *logfile* **refers to the location of the generated `ant` log file.**

   This step installs and configures the Communications tab in the Enterprise sample portal. You can access a fully configured Communications tab with sample data for

each channel: address book, mail, tasks, calendar, and instant messaging.

6. **Restart the web container.**

   This will allow the Communications tab to be configured on the Enterprise Sample Portal desktop under the My Portal tab.

CHAPTER 8

Enabling User Behavior Tracking

This supplement includes instructions for enabling User Behavior Tracking. For more information on this feature, see "User Behavior Tracking" in *Sun Java System Portal Server 7 2005Q4 Release Notes* in *Sun Java System Portal Server 7 2005Q4 Release Notes*.

Enabling User Behavior Tracking

▼ To Enable User Behavior Tracking

Steps

1. **Set `com.sun.portal.ubt.enable=true` in `/`*PortalData-Dir*`/portals/`*PortalID*`/config/UBTConfig.properties` file**
2. **Start accessing portal desktop and continue with normal operations on the channels.**
3. **Observe the running UBT logs in `/`*PortalData-Dir*`/portals/`*PortalID*`/logs/`*InstanceID*`/ubt.0.0.log` file.**

   Here:

PortalData-Dir	Refers to the portal data directory; for example, `/var/opt/SUNWps`
PortalID	Indicates the Portal ID; for example `myPortal`
InstanceID	Indicates the portal instance ID; for example `sprint_80`

   Use the UBT log file to run the `psadmin generate-ubt-report` command to get sample UBT reports.

CHAPTER 9

Setting Up Federated Search

The Federated Search feature enables users to submit a search query to multiple search engines concurrently and have the search results displayed in a unified format. The Federated Search feature provides a single interface for the user to post a search query to both a web meta-repository, such as google.com and an internal directory system such as a local personnel directory. The search results from these two different sites are presented to the user in a single web page.

Federated Search

▼ To Set Up Federated Search

Steps

1. **Create a Google account and download `googleapi.jar` from `http://www.google.com/apis/`**
2. **Obtain the license key for using `googleapi.jar`.**
3. **Set up sample federated databases:**
   a. **From a terminal window, log in to the host where search server is installed.**
   b. **Type the following:**
      `cd /opt/SUNWps/sdk/search`
   c. **Modify the `sampledbs.soif` file to change google clientKey value to be your downloaded license key, and modify databaseurl, providerurl, rdmserverurl, and other url values, accordingly.**
      Use the SOIF file syntax. The number in curly brackets ({ }) following the attribute is the number of characters you enter for that attribute's value.
   d. **Type the following:**
      `cd /var/opt/SUNWps/searchservers/`*search-server*

```
./run-cs-cli rdmgr -y root
/opt/SUNWps/sdk/search/sampledbs.soif
```

`./run-cs-cli rdmgr -y root -U` to verify that the soif entries containing the configurations for sample federated databases in the `sampledbs.soif` are in the root db.

4. **Add `googleapi.jar` to the web container's class path:**

   On the Application Server:

   `cp` *google-api-install-directory*`/googleapi/googleapi.jar /var/opt/SUNWappserver7/domains/domain1/server1/applications/j2ee-modules/`*search-se*

   On the Web Server:

   Edit `/opt/SUNWwbsvr/https-`*host.domain*`/config/server.xml`

   Add the path *google-api-install-directory*`/googleapi/googleapi.jar` to the serverclasspath.

5. **Restart the web container.**

▼ To Test Federated Search

Steps

1. **Use the `rdmserver` front-end by:**
   a. **Go to http://***host-name***.red.iplanet.com/***search-server***/testrdm.html**
   b. **Select "rd-request" for Type, select "search" for "Query Language."**
   c. **Enter a federated db such as "google" for "Database."**
   d. **Enter a query such as "java" for "Scope".**
   e. **Click "Submit GET."**
   f. **Verify that search results are returned.**
2. **Use the Search channel by modifying Search channel JSPs to add federated databases to the database list and view attributes for federated search results.**

CHAPTER 10

Establishing Trust Between Two Cacao Agents

This supplement includes instructions for establishing trust between two cacao agents.

Establishing Trust Between Two Cacao Agents

With this release, any `psadmin` subcommand can be remotely executed. This means that `psadmin` command can be executed from portal on one machine to a portal on another machine. To do this:

1. Stop the cacao server on the second machine. To stop, type *PortalServer7-base*`/SUNWcacao/bin/cacaoadm stop`.
2. Copy the `/etc/opt/SUNWcacao/security` directory from the first machine to the other.
3. Start the cacao server on the second machine. To start, type *PortalServer7-base*`/SUNWcacao/bin/cacaoadm start`.

Verify this by running the `psadmin list-portals` command from the first machine.

CHAPTER 11

Setting Up Registry Support for WSRP

This supplement includes instructions for setting up registry support for WSRP.

Setting Up Registry Support

▼ To Set Up Registry Support

Steps

1. **Install and configure Sun Service Registry Server from Sun Java Enterprise System 4 before installing Portal Server and the stack components.**

   If portal server is on a different node, install just `SUNWsoar-sdk` from Sun Java Enterprise System 4 on the Portal Server host.

2. **After installation, log in to the Portal Server administration console and update the value of the `sunRegistryServer` service in SSO Adapter.**

3. **Copy the keystore that has the specified alias in the properties file to `/soar/3.0/jaxr-ebxml/location-specified-in-ssoRegistryServer.config` file.**

4. **Log in to the Access Manager administration console and grant SSO Adapter service to `amadmin`.**

CHAPTER 12

Modifying Proxylet Rules

Proxylet rules specify the domain and proxy settings in the Proxy Auto Configuration (PAC) file.

The default behavior of Proxylet is changed as follows:

When a user logs into the Portal desktop, the Proxylet channel contains a list of application URLs (much like the Netlet channel containing Netlet rules). When a user clicks on a link, Proxylet is launched if it is not already running. Once Proxylet is launched, the user is redirected to the application URL page. The Proxylet channel user interface contains controls to stop and start Proxylet. Clicking on the stop button on the user interface restores the proxy settings and stops the server. If a rule contains the string proxylet-host:proxylet-port as the proxy server, then the generated PAC file replaces the string with the host and port of Proxylet. You can make a rule so that all FTP traffic is routed through Netlet and all HTTP traffic is routed through Proxylet.

Modifying Proxylet Rules

▼ To Modify the Proxylet Rules

Step

- **Enter the proxy-host and proxy-port, using the following syntax:**
  [*protocol* :] *Domain1* [, *Domain2* , ... , *n*] : *IP* or *Host* : *Port*

  The following list describes the variables you use:

protocol	Can contain `http` or `ftp` or `https`. (This field is optional).
domain	Is any domain such as sun.com. Multiple domains are separated by a comma.
IP	Speicifies the IP address of the domain.
proxy-host	Specifies the proxy server used for this domain(s).

proxy-port Specifies the proxy server port.

CHAPTER 13

Creating a New Portal

This supplement includes instructions for creating a new empty portal and deploying sample content into an empty portal.

Creating a New Portal

This sections contains the following:

▼ To Create a New Empty Portal

Steps

1. **Create a new web container instance.**

   For example, `second`.

2. **Copy**
   PortalServer7-base`/SUNWportal/template/Webcontainer.properties.JESWS6`
   to *PortalServer7-base*`/SUNWportal/bin/second.properites` **file.**

3. **Edit the following properties in the**
   PortalServer7-base`/SUNWportal/bin/second.properties` **file.**

   - `Host=`*host.domain*
   - `Port=`*port*
   - `WebContainerInstanceName=second`
   - `WebContainerInstanceDir=/opt/SUNWwbsvr/https-second`

4. **Run the following commands:**

   `psadmin create-portal -u` *amadmin* `-f` *ps_password* `-p Second --uri /portal -w second.properties`

5. **Restart the web container.**

6. **Verify that the new portal is created properly. To verify:**
   - Type `psadmin list-portals -u` *amadmin* `-f` *ps_password*
   - Login to Access Manager administration console to see new portal-centric services.
   - Access the new portal via the browser.

▼ To Deploy Sample Content (3 samples) on a New Portal

Steps

1. **Make a copy of the** *PortalServer7-base*`/SUNWportal/samples/portals/shared/input.properties` **file and edit the following properties in the file.**
   - `ps.config.location=/etc/opt/SUNWps`
   - `ps.portal.id=Second`
   - `ps.instance.id=host_port`

     Tip – You can find out the exact instance-ID from the output of `psadmin list-portals` command.

   - `ps.access.url=http://host.domain:port/portal` For example, `http://siroe.com:80/portal`
   - `ps.webapp.uri=/portal`
   - `ps.profiler.email=admin@siroe.com`
   - `ps.profiler.smtp.host=host.domain`
   - `search.access.url=http://host.domain:port/mySearch/search`
   - `search.id=search1`
   - `am.admin.dn=uid=amAdmin,ou=People,dc=siroe,dc=com`
   - `default.org.dn=dc=siroe,dc=com`

2. **Make a copy of** *PortalServer7-base*`/SUNWportal/samples/portals/shared/password.properties` **file and edit the following properties in the file to set proper passwords.**
   - `amadminPassword=%AMADMIN_PASSWORD%`
   - `amldapuserPassword=%AMLDAPUSER_PASSWORD%`
   - `userManagementPassword=%USER_MANAGEMENT_PASSWORD%` (optional; can be ignored if you are not setting up the comm channels)

3. **Remove the following files before running the sample content configuration ant script:**
   - Directory:
     - `/var/opt/sun/portal/tmp/par` on Linux
     - `/var/opt/SUNWportal/tmp/par` on Solaris
   - Files:
     - `community_sample.par`
     - `developer_sample.par`
     - `enterprise_sample.par`
     - `welcome_sample.par`
4. **Change the order of targets in the developer sample** *PortalServer7-base*`/SUNWportal/samples/portals/developer/build.xml` **file.**

   In the `run` target of developer sample `build.xml`, `config_portlets` and `config_wsrp` targets are invoked before the `par_import` target. The portlet providers are created in the `config_portlets` target and then overwritten by the display profile import in the `par_import` target. Hence the portlet providers become unavailable. Change the order of the targets in this file.
5. **Type** `/usr/sfw/bin/ant -buildfile` *PortalServer7-base*`/SUNWportal/samples/portals/build.xml`**.**

   You will be required to specify the location of the customized `input.properties` and `password.properties` file.

Note – To capture the output of the sample portal content configuration, specify a log file when invoking ant. For example, type `ant -buildfile` *PortalServer-base*`/SUNWps/samples/portals/build.xml -logfile /tmp/samplesinstall.txt`.

▼ To Deploy Sample Content (Any One Sample) on a New Portal

Steps

1. **Make a copy of the** *PortalServer7-base*`/SUNWportal/samples/portals/shared/input.properties` **file and edit the following properties in the file.**
   - `ps.config.location=/etc/opt/SUNWps`
   - `ps.portal.id=Second`
   - `ps.instance.id=host_port`

Tip – You can find out the exact instance-id from the output of `psadmin list-portals` command.

- `ps.access.url=http://host.domain:port/portal` For example, `http://siroe.com:80/portal`
- `ps.webapp.uri=/portal`
- `ps.profiler.email=admin@siroe.com`
- `ps.profiler.smtp.host=host.domain`
- `search.access.url=http://host.domain:port/mySearch/search`
- `search.id=search1`
- `am.admin.dn=uid=amAdmin,ou=People,dc=siroe,dc=com`
- `default.org.dn=dc=siroe,dc=com`

2. **Make a copy of** *PortalServer7-base*`/SUNWportal/samples/portals/shared/password.properties` **file and edit the following properties in the file to set proper passwords.**
   - `amadminPassword=%AMADMIN_PASSWORD%`
   - `amldapuserPassword=%AMLDAPUSER_PASSWORD%`
   - `userManagementPassword=%USER_MANAGEMENT_PASSWORD%` (optional; can be ignored if you are not setting up the comm channels)
3. **Remove the following files before running the sample content configuration ant script:**
   - Directory:
     - `/var/opt/sun/portal/tmp/par` on Linux
     - `/var/opt/SUNWportal/tmp/par` on Solaris
   - Files:
     - `community_sample.par`
     - `developer_sample.par`
     - `enterprise_sample.par`
     - `welcome_sample.par`
4. **Change the order of targets in the developer sample** *PortalServer7-base*`/SUNWportal/samples/portals/developer/build.xml` **file.**

   In the `run` target of developer sample `build.xml`, `config_portlets` and `config_wsrp` targets are invoked before the `par_import` target. The portlet providers are created in the `config_portlets` target and then overwritten by the display profile import in the `par_import` target. Hence the portlet providers become unavailable. Change the order of the targets in this file.

5. **Type:**

   - **`/usr/sfw/bin/ant -buildfile`** *PortalServer7-base*`/SUNWportal/samples/portals/welcome/build.xml` to deploy the Welcome page content.
   - **`/usr/sfw/bin/ant -buildfile`** *PortalServer7-base*`/SUNWportal/samples/portals/developer/build.xml` to deploy the Developer Sample Portal content.
   - **`/usr/sfw/bin/ant -buildfile`** *PortalServer7-base*`/SUNWportal/samples/portals/enterprise/build.xml` to deploy the Enterprise Sample Portal content.

CHAPTER 14

Starting and Stopping the Apache Derby Open-Source Database

This supplement describes how to start and stop the Apache Derby open source database used by the collaboration feature. For more information on this feature, see *Sun Java System Portal Server 7 2005Q4 Release Notes*.

Starting and Stopping the Apache Derby Open-Source Database

▼ To Start and Stop the Apache Derby Open-Source Database

Portal Server software uses a Apache Derby open-source database to store configuration and membership for the collaboration feature. The Portal Server software installer installs and starts the database. The portal creation process configures the database for the portal. This includes the default portal (myPortal) that is created at install time. The Derby database runs as a standalone Java application.

Step

- **The database can be stopped and started using the Derby `NetworkServerControl` class. For example, type `java org.apache.derby.drda.NetworkServerControl start -h` *your—host* `-Dderby.system.home=/var/opt/SUNWportal/derby` where `-h` argument is required, and must match the Portal Server host name chosen at install time. To run this command:**
  - The `derby.jar`, `derbytools.jar`, and `derbynet.jar` files must be in your `classpath`. By default, these JAR files are installed into `/usr/share/lib/Derby` directory.
  - The system property `derby.system.home` must be set to:
    - `/var/opt/SUNWportal/derby` for Solaris.
    - `/var/opt/sun/portal/derby` for Linux.

Portal accesses the database using a JDBC datasource configured in the web container. The datasource configuration can be modified using the web container console, or command line interface. The database URL for the Derby community database is of the form `jdbc:derby://<host>[:<port>]/communitymc_<portalID>`. When connecting to the Derby database using third-party tools, use the driver `org.apache.derby.jdbc.ClientDriver`. This driver is in the JAR file `/usr/share/lib/Derby/derbyclient.jar`.

CHAPTER 15

Configuring Surveys and Polls Portlets

This supplement contains instructions for configuring Surveys and Polls portlets for the Communities Sample.

Configuring Surveys and Polls Portlets

▼ To Configure Surveys and Polls Portlets

Before You Begin

Ensure that the Derby server instance is up and running by executing the following command:

```
netstat -a | grep DERBY-PORT
```

If the server is not running, start the server. See supplement Chapter 14 for detailed instructions on starting and stopping the server.

Steps

1. **Copy the *PortalServer7-base*`/SUNWportal/portletapps/surveys` directory to `/var/opt/SUNWportal/portals/`*PortalId*`/portletapps/surveys` directory.**

2. **Edit the following tokens (starting and ending with `%`) in `/var/opt/SUNWportal/portals/`*PortalId*`/portletapps/input.properties` file:**

   `ps.config.location=%PS_CONFIG_LOCATION%`
   Portal configuration base directory. By default, this is `/etc/opt/SUNWportal`.

   `ps.portal.id=%PORTAL_ID%`
   Portal identifier. For example, `myPortal`.

   `am.admin.dn=%AM_ADMIN_DN%`
   Sun Java System Access Manager administrator's distinguished name. For example, use the format `uid=amAdmin,ou=People,dc=siroe,dc=com`.

   `amadminPassword=%AMADMIN_PASSWORD%`
   Access Manager administrator's (`amadmin`) password.

`ps.webapp.uri=%PORTAL_WEBAPP_URI%`
Portal web application URI. By default, this is `/portal`.

3. **Edit the tokens in `/var/opt/SUNWportal/portals/`*PortalId*`/portletapps/surveys/tokens.properties` file.**

   For example, edit the following tokens (starting and ending with `%`):

   `DB_ADMIN_URL=jdbc:derby://%DERBY_HOST%:%DERBY_PORT%/surveydb_myPortal;create=tru`
   Apache Derby Open Source database host name and domain, and the port on which derby server instance is listening.

   `DB_URL=jdbc:derby://%DERBY_HOST%:%DERBY_PORT%/surveydb_%PORTAL_ID%`
   Apache Derby Open Source database host name and domain, the port on which derby server instance is listening, and portal ID (for example, `myPortal`).

4. **Run the ant command. Type `/usr/sfw/bin/ant -buildfile` *Portal-Data-Dir*`/portals/`*PortalId*`/portletapps/surveys/build.xml`.**

   To capture output in a log file, run the following command: `/usr/sfw/bin/ant -buildfile` *Portal-Data-Dir*`/portals/`*PortalId*`/portletapps/surveys/build.xml -logfile /tmp/portletappslogs.txt`.

CHAPTER 16

Configuring Back-end Services

This supplement contains information on configuring back-end services for certain collaboration portlets.

Configuring Back-end Services for Portlets

Certain service portlets in the Communities Sample require certain back-end services to be setup and configured prior to installing and configuring the portlet. These back-end services are Sun Java System Calendar Server and Sun Delegated Administration software. You can use an existing deployment or, choose to perform a fresh install and configuration of these services. The Calendar Server, Delegated Administration software, and Portal Server software must use the same Directory Server.

▼ To Configure Back-end Services for Portlets

Steps

1. **See detailed documentation for configuring Calendar Server at II Postinstallation Configuration in Sun Java System Calendar Server 6 2005Q4 Administration Guide.**
2. **See detailed documentation for configuring Delegated Administration software at Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide.**
3. **Enable Proxy authentication in Calendar Server.**
   a. **Edit `CS-config-directory/ics.conf` file (where *CS-config-file* is by default `/etc/opt/SUNWics5/config/` directory) and change the value of `service.http.allowadminproxy` property to `yes`.**
   b. **Restart the Calendar Server.**
4. **Gather the following information before deploying the portlets.**

`UGRoot Suffix`	This is the ser/Group Root suffix of the Directory Server. This information is provided while running the `commds_setup.pl` script.
`CSUGRoot Suffix`	This is the `UGRoot suffix` under which the calendar users are created by default. It is under this suffix, the calendar admin user (calmaster) is created. It is important to choose this suffix appropriately such that the calendar admin user is able to proxy-authenticate other calendar users in other organizations. It is recommended that this suffix be the same as `UGRoot Suffix`. Here, this specifies the value of Base Distinguished Node configuration input while configuring Calendar Server.
`DA Default Domain`	This is the DNS domain name of the default organization used by Delegated Administration software (for provisioning users by default). Here, this specifies the value of Default Domain configuration input while configuring Delegated Administration software.
`DA Default Organization DN`	The Organization distinguished node will be the LDAP subtree under which all the users and groups that belong to the Delegated Administration software Default Domain are located. Note that the distinguished node must be located under the `UGRoot Suffix`. Here, this specifies the value of Organization Distinguished Node for the Default domain configuration input while configuring Delegated Administration software.

CHAPTER 17

Deploying the SharedTasks Portlet

This supplement contains instructions for installing and configuring the SharedTasks portlet for the collaboration Desktop.

Installing and Configuring the SharedTasks Portlet

▼ To Install and Configure the SharedTasks Portlet

Before You Begin Configure the back-end services as outlined in supplement Chapter 16.

Steps

1. **Create the portlet WAR file.**
   To create the WAR file:

   a. **Edit the `tokens.properties` file to specify customizable parameters for the portlet you are deploying.**
      Shared Events defines the following tokens:

      `SSOADAPTER_TEMPLATE_NAME` — Default value is `CTY-SHARED-EVENTS`.

      Specifies the name of SSO Adapter configuration template name. This template contains the required configuration information for the portlet functioning. This template is per-portlet application; that is, the name should be different for each deployed portlet instance.

      For example, `HACKERS_COMMUNITY_EVENTS`

SSOADAPTER_TEMPLATE_DESC	Default value is CTY-SHARED-EVENTS. Contains the description for SSO Adapter template. This needs to be the same as SSOADAPTER_TEMPLATE_NAME for the template to be editable from Access Manager console. For example, HACKERS_COMMUNITY_EVENTS.
SSOADAPTER_ORGANIZATION_DN	Specifies the distinguished node of organization under which the SSO Adapter template is loaded. For example, o=CommunitySample,dc=siroe,dc=example,dc=com
DA_SERVER_HOST	Specifies the fully qualified name of the system where Delegated Administration server is running. For example, daserver.blue.planet.com.
DA_SERVER_PORT	Default value is 80. Specifies the port number of the Delegated Administration server. For example, 80.
DA_DOMAIN_SEPARATOR	Default value is @. Specifies the user ID and Domain Separater character. For example, @.
DA_PROXY_ADMIN_USER_ID	Specifies the admin user for Delegated Administration that can be used to perform proxy authentication of users. For example, admin.
DA_PROXY_ADMIN_USER_PASSWORD	Specifies the password for DA_ADMIN_USER_ID. For example, abc123.

DA_DEFAULT_DOMAIN	Specifies the DNS Domain Name of the organization where the CTY-PROXY-USER would be provisioned. For example, blue.planet.com.
IS_HOSTED_DOMAIN_ENV	Default value is false. Specifies whether the portlet should use a hosted domains environment. **Note –** If the Delegated Administration server and Calendar Server are configured to use hosted domains setup, the value needs to be set to true. For example, false
CALENDAR_SERVER_HOST	Specifies the fully qualified name of the system where Calendar Server is running. For example, calendar.blue.planet.com.
CALENDAR_SERVER_PORT	Default value is 3080. Specifies the port Number of Calendar Server. For example, 3080.
CALENDAR_PROXY_ADMIN_UID	Specifies the admin user for Calendar Server that can be used to perform proxy authentication of users. For example, calmaster.
CALENDAR_PROXY_ADMIN_PASSWORD	Specifies the password for CALENDAR_PROXY_ADMIN_UID. For example, calmaster.
AUTO_DELETE_CTY_PROXY_USER_CAL	Default value is false.

	Specifies whether the portlet should delete the `CTY-PROXY-USER` when a community is deleted. The `CTY-PROXY-USER` is the community proxy user, a user created by the Shared Tasks portlet For example, `false`.
`AM_ADMINISTRATOR_DN`	Specifies the distinguished name of the administrator for Access Manager software. For example, `uid=amAdmin,ou=People,dc=blue,dc=planet,dc=c`
`AM_ADMINISTRATOR_PASSWORD`	Specifies the password for `AM_ADMINISTRATOR_DN`. For example, `abc123`.
`AMADMIN_COMMAND`	Default value is `/opt/SUNWam/bin/amadmin`. Specifies the fully qualified path name to the `amadmin` CLI of Access Manager. For example, `/opt/sun/am/bin/amadmin`.

b. Run `ant customize`.

After Portal Server software has been installed, the Shared Tasks application will be under the `/opt/SUNWportal/portletapps/sharedtasks` directory. If Portal Server has been configured, the shared tasks application will be in `/var/opt/SUNWportal/portletapps/sharedtasks` directory.

2. Load SSO Adapter Template.

Change directories to `/var/opt/SUNWportal/portletapps/sharedtasks/build/conf` directory and type **`ant -f config.xml`** to load the SSO Adapter configuration.

3. Deploy the Portlets.

a. Create a file containing the password of the amadmin user. For example, type, `echo mypassword >/tmp/ampasswd`.

b. **Type `/opt/SUNWportal/bin/psadmin deploy-portlet -u amadmin -f /tmp/ampassword -p` *myPortal* `-v -i` *myInstance* `-g` *FULL_PATH_TO_sharedevents.war_FILE* `--userinfofile` *FULL_PATH_TO_userInfoMapping.properties*.**

c. **Restart the container.**

4. **Add the Display Profile data to the templates.**

The Shared Tasks portlet must be added to the Community templates to show up in new communities (created after the application has been installed). The template files are located at `/var/opt/SUNWportal/portals/myPortal/communitytemplates/` by default. Add the DP to the `member.xml` file in the template's directory.

For Shared Tasks, do the following:

a. **Add your new channel to the available ones.**

```
<Available>
....
<Reference value="%COMMUNITY_CONTAINER%/Sharedtasks"/>
</Available>
```

b. **Add your new channel to the selected ones.**

```
<Selected>
....
<Reference value="%COMMUNITY_CONTAINER%/Sharedtasks"/>
</Selected>
```

c. **Add the channel definition.**

```
<Channels>
....
<Channel name="Sharedtasks" provider="_Portlet__sharedtasks.sharedtasks">
    <Properties>
        <String name="title" value="Community Calendar Tasks"/>
        <String name="description" value="Community Tasks Portlet"/>
        <String name="width" value="thick"/>
        <String name="__Portlet__ps.communityId" value="|DUMMY"/>
        <String name="__Portlet__community.calendar.config" value="|DUMMY"/>
        <Collection name="__Portlet__PreferenceProperties">
            <Collection name="default">
                <String name="ps.communityId" value="|DUMMY"/>
                <String name="community.calendar.config" value="|DUMMY"/>
            </Collection>
            <Collection name="isReadOnly">
                <Boolean name="ps.communityId" value="false"/>
                <Boolean name="community.calendar.config" value="false"/>
            </Collection>
        </Collection>
        <Collection  name="userInfoDescriptions">
            <Collection name="timezone">
```

```
                <String name="en-US" value="Preferred User Time Zone"/>
            </Collection>
        </Collection>
    </Properties>
</Channel>
</Channels>
```

5. **Restart the web container.**
6. **Login and verify that the service channel is displayed on your desktop.**
   a. **Go to `http://HOST:PORT/portal/dt` and select Community Samples.**
   b. **Login as test/test and create a new community. Then log out.**
   c. **Login to the community portal as test/test.**
      The Shared Tasks portlet will be displayed.

CHAPTER 18

Installing and Configuring SharedEvents Portlet

This supplement contains instructions for installing and configuring the SharedEvents portlet for the Communities Sample.

Deploying the SharedEvents Portlet

▼ To Install and Configure SharedEvents Portlet

Before You Begin Configure the back-end services as outlined in supplement Chapter 16.

Steps

1. **Create the portlet WAR file.**

   To create the WAR file:

   a. **Edit the `tokens.properties` file to specify customizable parameters for the portlet you are deploying.**

      Shared Events defines the following tokens:

      `SSOADAPTER_TEMPLATE_NAME`
      Default value is `CTY-SHARED-EVENTS`.

      Specifies the name of SSO Adapter configuration template name. This template contains the required configuration information for the portlet functioning. This template is per-portlet application; that is, the name should be different for each deployed portlet instance.

      For example, `HACKERS_COMMUNITY_EVENTS`

      `SSOADAPTER_TEMPLATE_DESC`
      Default value is `CTY-SHARED-EVENTS`.

      Contains the description for SSO Adapter template. This needs to be the same as `SSOADAPTER_TEMPLATE_NAME` for the template to be editable from Access Manager console.

For example, HACKERS_COMMUNITY_EVENTS.

SSOADAPTER_ORGANIZATION_DN
Specifies the distinguished node of organization under which the SSO Adapter template is loaded.

For example, o=CommunitySample,dc=siroe,dc=example,dc=com.

DA_SERVER_HOST
Specifies the fully qualified name of the system where Delegated Administration server is running.

For example, daserver.blue.planet.com.

DA_SERVER_PORT
Default value is 80.

Specifies the port number of the Delegated Administration server.

For example, 80.

DA_DOMAIN_SEPARATOR
Default value is @.

Specifies the user ID and Domain Separater character.

For example, @.

DA_PROXY_ADMIN_USER_ID
Specifies the admin user for Delegated Administration that can be used to perform proxy authentication of users.

For example, admin.

DA_PROXY_ADMIN_USER_PASSWORD
Specifies the password for DA_ADMIN_USER_ID.

For example, abc123.

DA_DEFAULT_DOMAIN
Specifies the DNS Domain Name of the organization where the CTY-PROXY-USER would be provisioned.

For example, blue.planet.com.

IS_HOSTED_DOMAIN_ENV
Default value is false.

Specifies whether the portlet should use a hosted domains environment.

Note – If the Delegated Administration server and Calendar Server are configured to use hosted domains setup, the value needs to be set to true.

For example, `false`

`CALENDAR_SERVER_HOST`
Specifies the fully qualified name of the system where Calendar Server is running.

For example, `calendar.blue.planet.com`.

`CALENDAR_SERVER_PORT`
Default value is `3080`.

Specifies the port Number of Calendar Server.

For example, `3080`.

`CALENDAR_PROXY_ADMIN_UID`
Specifies the admin user for Calendar Server that can be used to perform proxy authentication of users.

For example, `calmaster`.

`CALENDAR_PROXY_ADMIN_PASSWORD`
Specifies the password for `CALENDAR_PROXY_ADMIN_UID`.

For example, `calmaster`.

`AUTO_DELETE_CTY_PROXY_USER_CAL`
Default value is `false`.

Specifies whether the portlet should delete the `CTY-PROXY-USER` when a community is deleted.

For example, `false`.

`CREATE_EVENTS_IN_CTY_MEMBERS_CAL`
Default value is `true`.

Specifies whether the portlet should create the calendar event in all the member's community calendars whenever a community event is created.

For example, `true`.

`AM_ADMINISTRATOR_DN`
Specifies the distinguished name of the administrator for Access Manager software.

For example, `uid=amAdmin,ou=People,dc=blue,dc=planet,dc=com`

`AM_ADMINISTRATOR_PASSWORD`
Specifies the password for `AM_ADMINISTRATOR_DN`.

For example, `abc123`.

`AMADMIN_COMMAND`
Default value is `/opt/SUNWam/bin/amadmin`.

Specifies the fully qualified path name to the amadmin CLI of Access Manager.

For example, `/opt/sun/am/bin/amadmin`.

b. Run `ant customize`.

After Portal Server software has been installed, the Shared Events application will be under the `/opt/SUNWportal/portletapps/sharedevents` directory. If Portal Server has been configured, the shared events application will be in `/var/opt/SUNWportal/portletapps/sharedevents` directory.

2. Load SSO Adapter Template.

Change directories to `/var/opt/SUNWportal/portletapps/sharedevents/build/conf` directory and type **`ant -f config.xml`** to load the SSO Adapter configuration.

3. Deploy the Portlets.

a. Create a file containing the password of the amadmin user. For example, type, `echo mypassword >/tmp/ampasswd`.

b. Type `/opt/SUNWportal/bin/psadmin deploy-portlet -u amadmin -f /tmp/ampassword -p` *myPortal* `-v -i` *myInstance* `-g` *FULL_PATH_TO_sharedevents.war_FILE* `--userinfofile` *FULL_PATH_TO_userInfoMapping.properties*.

c. Restart the container.

4. Add the Display Profile data to the templates.

The Shared Events portlet must be added to the Community templates to show up in new communities (created after the application has been installed). The template files are located at `/var/opt/SUNWportal/portals/myPortal/communitytemplates/` by default. Add the DP to the `member.xml` file in the template's directory.

For Shared Events, do the following:

a. Add your new channel to the available ones.

```
<Available>
....
<Reference value="%COMMUNITY_CONTAINER%/Sharedevents"/>
</Available>
```

b. Add your new channel to the selected ones.

```
<Selected>
....
<Reference value="%COMMUNITY_CONTAINER%/Sharedevents"/>
</Selected>
```

c. Add the channel definition.

```
<Channels>
....
<Channel name="Sharedevents" provider="__Portlet__sharedevents.sharedevents">
    <Properties>
        <String name="title" value="Community Calendar Events"/>
        <String name="description" value="Community Events Portlet"/>
        <String name="width" value="thick"/>
        <String name="__Portlet__ps.communityId" value="|DUMMY"/>
        <String name="__Portlet__community.calendar.config" value="|DUMMY"/>
        <Collection name="__Portlet__PreferenceProperties">
            <Collection name="default">
                <String name="ps.communityId" value="|DUMMY"/>
                <String name="community.calendar.config" value="|DUMMY"/>
            </Collection>
            <Collection name="isReadOnly">
                <Boolean name="ps.communityId" value="false"/>
                <Boolean name="community.calendar.config" value="false"/>
            </Collection>
        </Collection>
        <Collection  name="userInfoDescriptions">
            <Collection name="timezone">
                <String name="en-US" value="Preferred User Time Zone"/>
            </Collection>
        </Collection>
    </Properties>
</Channel>
</Channels>
```

5. **Restart the web container.**
6. **Login and verify that the service channel is displayed on your desktop.**
   a. **Go to `http://HOST:PORT/portal/dt` and select Community Samples.**
   b. **Login as test/test and create a new community. Then log out.**
   c. **Login to the community portal as test/test.**
      The Shared Events portlet will be displayed.

CHAPTER 19

Desktop Subcommands

This supplement contains the following:

psadmin get-attribute

Description

Gets the portal attribute for a base dn or from global or default levels.

Syntax

Long Format

psadmin get-attribute --component desktop -attribute-name *attribute-name* --adminuser *uid* --passwordfile *password-filename* [--dn *dn*] -portalid *portal-ID* --verbose

Short Format

psadmin get-attribute -m desktop -a *attribute-name* -u *uid* -f *password-filename* [-d *dn*] -p *portal-ID* -v

Options

The following options are required:

[--adminuser | -u] *uid*
Specifies the administrator's distinguished name.

`[--attribute-name | -a]` *attribute-name*
Specifies the desktop attribute name for which the value is to be retrieved. User friendly desktop attributes can be fetched from the `list-attribute` subcommand, with component name as desktop.

`[--component | -m] desktop`
Always `desktop` for this module.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portalId | -p]` *portal-ID*
Specifies the portal ID.

The following options are optional:

`[--dn | -d]` *dn*	Specifies the distinguished name for whom desktop attribute is to be retrieved.
`[--verbose | -v]`	Removes display profile verbosely.

psadmin set-attribute

Description

Sets the portal desktop attribute to the supplied value or values for a base `dn` or at global or default levels.

Syntax

Long Format

`psadmin set-attribute --component desktop --attribute-name` *attribute-name* `--adminuser` *uid* `--passwordfile` *password-filename* `[-add` *add_values*`] [--remove` *remove_values*`] [--inherit ] [--dn` *dn*`] --portalid` *portal-ID* `set_values`

Short Format

`psadmin set-attribute -m desktop -a` *attribute-name* `-u` *uid* `-f` *password-filename* `[-d` *dn*`] -p` *portal-ID* `[ -A` *add_values*`] [-E` *remove_values*`][--inherit] set_values`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--attribute-name | -a]` *attribute-name*
Argument which specifies the name of the desktop attribute for which the value is to be added, removed, replaced, or inherited. User friendly desktop attributes can be fetched from `list-attribute` command, with component name as desktop.

`[--component | -m] desktop`
Always `desktop` for this module.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portalId | -p]` *portal-ID*
Specifies the portal ID.

The following options are optional:

`[--add | -A]` *add_values*	Applies to multi value attribute. Specifies a comma separated list of values to be set for an attribute. Mutually exclusive to `--inherit` option and `set_values` operand. The list of value is enclosed in `"`.
`[--dn | -d]` *dn*	Specifies the distinguished name for whom desktop attribute is to be set.
`--inherit`	Allows the specified attribute at the given `dn`, to inherit its value from the parent `dn`. The `dn` is mandatory to use this option.
`[--remove | -E]` *remove_values*	Applies to multi value attribute. Specifies a comma separated list of values to be removed for an attribute. Mutually exclusive to `--inherit` option and `set_values` operand. The list of value is enclosed in `"`.

Operands

The following operand is supported:

`set_values` Specifies the value to be set.

Limitations

You cannot do the following:

- Use `--add` or `--remove` option with the `--inherit` option.
- Use `--add`, `--remove`, or `--inherit` options with `set_values` operand.
- Use the `--inherit` option without supplying the `--dn` option.

psadmin list-attribute

Description

Lists the portal desktop attributes which are defined at any base `dn` or at global or default levels. This subcommand can be used when an administrator wants to see the entire list of desktop attributes.

Syntax

Long Format

`psadmin list-attributes --component desktop --adminuser` *uid* `--passwordfile` *password-filename* `--verbose`

Short Format

`psadmin list-attributes -m desktop -u` *uid* `-f` *password-filename* `-v`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--component | -m] desktop`
Always `desktop` for this module.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--verbose | -v]`
Removes display profile verbosely.

psadmin list-dp

Description

Retrieves and displays display profile node objects.

Syntax

Long Format

`psadmin list-dp --name` *name* `--adminuser` *uid* `--passwordfile` *password-filename* `--dn` *dn* `--portalid` *portal-ID* `[--verbose] [--global] [--dry-run]`

Short Format

`psadmin list-dp --n` *name* `-u` *uid* `-f` *password-filename* `-d` *dn* `-p` *portal-ID* `[-v] [ -g] [-r]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--dn | -d]` *dn*
Distinguished name of the target node. This is mutually exclusive to `-g` option.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portalId| -p]` *portal-ID*
Specifies the portal ID; if this is not supplied, the default is used.

The following options are optional:

`[--dry-run | -r]`	Attempt to execute command without writing out to LDAP. Default is false.
`[--global | -g]`	Global display profile. Default is false. This is mutually exclusive to `-d` option.
`[--name | -n]` *name*	Name of the target display object. If omitted, the entire display profile is displayed.
`[--verbose | -v]`	Lists display profile verbosely.

psadmin merge-dp

Description

Retrieves and displays the merged result of the given display profile node objects.

Syntax

Long Format

`psadmin merge-dp --name` *name* `--adminuser` *uid* `--passwordfile` *password-filename* `--dn` *dn* `--portalid` *portal-ID* `[--verbose] [--global] [--dry-run]`

Short Format

`psadmin merge-dp --n` *name* `-u` *uid* `-f` *password-filename* `-d` *dn* `-p` *portal-ID* `[-v] [-g] [-r]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--dn | -d]` *dn*
The distinguished name of the target node. This is mutually exclusive to `-g` option.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portalId | -p]` *portal-ID*
Specifies the portal ID; if this is not supplied, the default is used.

The following options are optional:

`[--dry-run | -r]`	Attempt to execute command without writing out to LDAP. Default is false.
`[--global | -g]`	Global display profile. Default is false. This is mutually exclusive to `-d` option.
`[--name | -n]` *name*	Name of the target display object. If omitted, merges the entire display profile.
`[--verbose | -v]`	Merges display profile verbosely.

psadmin modify-dp

Description

Changes the value for an existing display profile object.

Syntax

Long Format

`psadmin modify-dp --parent` *parent* `--adminuser` *uid* `--passwordfile` *password-filename* `--dn` *dn* `--portalid` *portal-ID* `[--verbose] [--global] [--dry-run] [--combine]`

Short Format

`psadmin modify-dp -P` *parent* `-u` *uid* `-f` *password-filename* `-d` *dn* `-p` *portal-ID* `[-v] [-g] [-r] [-m]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--dn | -d]` *dn*
Distinguished name of the target node. This is mutually exclusive to `-g` option.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portalId | -p]` *portal-ID*
Specifies the portal ID, if this is not supplied, the default is be used.

The following options are optional:

`[--combine | -m]`	Combine with the existing display profile object. For example, use this option to add a channel specified in the existing XML file to a container's existing selected list. By default, this is false.
`[--dry-run | -r]`	Attempt to execute command without writing out to LDAP. Default is false.
`[--global | -g]`	Global display profile. Default is false. This is mutually exclusive to `-d` option.
`[--parent | -P]` *parent*	Name of the parent display object. If omitted, assumes the node to be modified is under root.
`[--verbose | -v]`	Modifies display profile verbosely.

Operands

Specifies one or more files which contain XML fragments.

psadmin add-dp

Description

Adds a new display profile object to the display profile.

Syntax

Long Format

```
psadmin add-dp --parent parent --adminuser uid --passwordfile
password-filename --dn dn --portalid portal-ID [--verbose] [--global]
[--dry-run]
```

Short Format

```
psadmin add-dp -P parent -u uid -f password-filename -d dn -p portal-ID [-v]
[-g] [-r]
```

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--dn | -d]` *dn*
Distinguished name of the target node. This is mutually exclusive to `-g` option.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portalId | -p]` *portal-ID*
Specifies the portal ID, if this is not supplied, the default is used.

The following options are optional:

`[--dry-run | -r]`	Attempt to execute command without writing out to LDAP. Default is false.
`[--global | -g]`	Global display profile. Default is false. This is mutually exclusive to `-d` option.
`[--parent | -P]` *parent*	Name of the parent display object. If not specified , the object gets added to the root.
`[--verbose | -v]`	Adds display profile verbosely

Operand

Specifies one or more files each of which contains XML fragment.

psadmin remove-dp

Description

Removes an existing display profile object from the display profile.

Syntax

Long Format

`psadmin remove-dp --name` *name* `--parent` *parent* `--type` *type* `--adminuser` *uid* `--passwordfile` *password-filename* `--dn` *dn* `--portalid` *portal-ID* `[ --verbose] [--global] [--dry-run]`

Short Format

`psadmin remove-dp -n` *name* `-P` *parent* `-t` *type* `-u` *uid* `-f` *password-filename* `-d` *dn* `-p` *portal-ID* `[-v] [-g] [-r]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--dn | -d]` *dn*
Distinguished name of the target node. This is mutually exclusive to `-g` option.

`[--name | -n]` *name*
Name of the target display object.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portalId | -p]` *portal-ID*
Specifies the portal ID; if this is not supplied, the default is used.

`[--type | -t]` *type*
Type of the display object:
`root,channel,provider,property,available,selected.`

The following options are optional:

`[--dry-run | -r]`	Attempt to execute command without writing out to LDAP. Default is false.
`[--global | -g]`	Global display profile. Default is false. This is mutually exclusive to `-d` option.
`[--parent | -P]` *parent*	Name of the parent display object. If not specified , the object gets removed from the root.
`[--verbose | -v]`	Removes display profile verbosely.

CHAPTER 20

Portlet Subcommands

This supplement contains the following:

psadmin deploy-portlet

Description

Deploys the portlet web application into the Portal Server. It deploys the portlet WAR file and inserts the provider into display profile.

Syntax

Long Format

`psadmin deploy-portlet --adminuser` *uid* `--passwordfile` *password-filename* `--dn` *dn* `| --global [--rolesfile` *roles-file*`] [--userinfofile` *user-info-file*`] --portal` *portal-ID* `[--instance` *portal-instance*`] [--verbose] portletwarfile`

Short Format

`psadmin deploy-portlet -u` *uid* `-f` *password-filename* `-d` *dn* `[-r` *roles-file*`] [-n` *user-info-file*`] -p` *portal*`-ID [-i` *portal-instance*`] [-g] [-v] portletwarfile`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--dn | -d]` *dn*
Specifies the distinguished name in the LDAP node to access the display profile document. The `-d` or `-g` option is required.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

The following options are optional:

`[--global | -g]`	Specifies the global level node in LDAP to access the display profile document. The `-d` or `-g` option is required.
`[--instance | -i]` *portal-instance*	Specifies the portal instance where the portlet is to be deployed. If not specified deploys on all the instances of the portal.
`[--rolesfile | -r]` *roles-file*	Specifies the file containing the Access Manager software and portlet application role mapping information.
`[--userinfofile | -n]` *user-info-file*	Specifies the file containing the user information mapping information.
`[--verbose | -v]`	Display debug messages.

Operands

`portletwarfile` Specifies the path to the WAR file.

psadmin undeploy-portlet

Description

Removes the portlet application from the portal server. It also removes the providers associated with portlets from display profile.

Syntax

Long Format

`psadmin undeploy-portlet --adminuser` *uid* `--passwordfile` *password-filename* `--dn` *dn* `| --global --portal` *portal-ID* `[--instance` *portal-instance*`] [--verbose] portletwebapp`

Short Format

`psadmin undeploy-portlet -u` *uid* `-f` *password-filename* `-d` *dn* `-p` *portal-ID* `[-i` *portal-instance*`] [-g] [-v] portletwebapp`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--dn | -d]` *dn*
Specifies the distinguished name in the LDAP node to access the display profile document. The `-d` or `-g` option is required.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

The following options are optional:

`[--global | -g]`	Specifies the global level node in LDAP to access the display profile document. The `-d` or `-g` option is required.
`[--instance | -i]` *portal-instance*	Specifies the portal instance from where the portlet needs to be undeployed. If not specified, undeploys the portlet application from all the instances of the portal.
`[--verbose | -v]`	Display debug messages.

Operands

`portletwebapp`	Name of the portlet application which needs to be undeployed.

CHAPTER 21

Management Subcommands

This section contains details about the following:

psadmin list-portals

Description

Produces a list of all portals and Portal Server instances that are available.

Syntax

Long Format

`psadmin list-portals --adminuser` *uid* `--passwordfile` *password-file* `[--portal` *portal-name*`]`

Short Format

`psadmin list-portals -u` *uid* `-f` *password-file-name* `[-p` *portal-name*`]`

Options

The following options are required:

[--adminuser| -u *uid*
Specifies the administrator's distinguished name (dn) or user ID (uid).

[--passwordfile | -f] *password-file-name*
Specifies the administrator's password in the password file.

The following option is optional:

[--portal | -p] *portal-name* — Include to display Portal Server instances for the portal specified. Omit to display Portal Server instances for all portals.

psadmin create-portal

Description

Sets up a new portal, registers it within the administration framework, and creates the portal's first Portal Server instance.

Syntax

Long Format

psadmin create-portal --adminuser *uid* --passwordfile *password-file* --portal *portal-name* --webconfig *web-config-file* --uri *portal-uri*

Short Format

psadmin create-portal -u *uid* -f *password-file-name* -p *portal-name* -w *web-config-file* --uri *portal-uri*

Options

The following options are required:

[--adminuser | -u] *uid*
Specifies the administrator's distinguished name (dn), or user ID (uid).

[--passwordfile | -f] *password-file-name*
Specifies the administrator's password in the password file.

[--portal | -p] *portal-name*
Specifies the name of the new portal.

[--webconfig | -w] *web-config-file*
Specifies the name of the web container properties file, which contains web container-specific configuration parameters. This properties file is in the /opt/SUNWportal/template directory for your platform.

--uri *portal-uri*
Specifies the uniform resource indicator (URI) where the portal web is deployed.

psadmin delete-portal

Description

Removes a portal from the administration framework and deletes all portal server instances and file system data associated with the portal. Undeploys the portal web application and portlet web applications deployed on the web container instances.

Syntax

Long Format

`psadmin delete-portal --adminuser` *uid* `--passwordfile` *password-file-name* `--portal` *portal-name*

Short Format

`psadmin delete-portal -u` *uid* `-f` *password-file* `-p` *portal-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or user ID (`uid`).

`[--passwordfile | -f]` *password-file-name*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-name*
Specifies the name of the portal to remove.

See Also

"`psadmin list-portals`" on page 115

psadmin create-instance

Description

Sets up a new Portal Server instance and associates it with a portal.

Syntax

Long Format

`psadmin create-instance --adminuser` *uid* `--passwordfile` *password-file* `--portal` *portal-name* `--webconfig` *web-config-file* `[--instance-name` *instance-name*`]`

Short Format

`psadmin create-instance -u` *uid* `-f` *password-file-name* `-p` *portal-name* `-w` *web-config-file* `[-i` *instance-name*`]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or user ID (`uid`).

`[--passwordfile | -f]` *password-file-name*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-name*
Specifies the name of the portal that the instance belongs to.

`[--webconfig | -w]` *web-config-file*
Specifies the name of the web container properties file, which contains web container-specific configuration parameters. This properties file is in the `/opt/SUNWportal/template` directory for your platform.

The following option is optional:

`[--instance | -i]` *intance-name*	Specifies the name of the new portal instance. If omitted, the administration framework generates the name as *host-name_port-number*.

psadmin delete-instance

Description

Removes a Portal Server instance from the portal, configures the web container to remove the instance entry, including undeploying the Portal WAR.

Syntax

Long Format

`psadmin delete-instance --adminuser` *uid* `--passwordfile` *password-file-name* `--portal` *portal-name* `--instance` *instance-name*

Short Format

`psadmin delete-instance -u` *uid* `-f` *password-file-name* `-p` *portal-name* `-i` *instance-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`), or user ID (`uid`).

`[--passwordFile | -f]` *password-file-name*
Specifies the administrator's password in the password file.

`[--instance | -i]` *instance-name*
Specifies the name of the instance to remove.

`[--portal | -p]` *portal-name*
Specifies the name of the portal that the instance belongs to.

psadmin redeploy

Description

Redeploys a Portal Server to a web container instance.

Syntax

Long Format

`psadmin redeploy --adminuser` *uid* `--passwordfile` *password-file-name* `--portal` *portal-name* `[--instance` *instance-name*`]`

Short Format

`psadmin redeploy -u` *uid* `-f` *password-file-name* `-p` *portal-name* `-i` *instance-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`), or user ID (`uid`).

`[--passwordfile | -f]` *password-file-name*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-name*
Specifies the name of the portal that the instance belongs to.

The following option is optional:

`[--instance | -i]` *instance-name*	Specifies the ID of the instance to redeploy. Include to redeploy the specified instance. Omit to redeploy all Portal Server instances for this portal.

psadmin export

Description

Exports the portal desktop, including the provider and channels, data.

Syntax

Long Format

```
psadmin export --adminuser uid --passwordfile password-file-name
[--files-only] --portal portal-name [--dp-only] --type type [--dn dn]
[--global] [--exportfile export_file_list] par-file-name
```

Short Format

```
psadmin export -u uid -f password-file-name [--files-only] -p portal-name
[--dp-only] -t type [-d] [-g] [-x export_file_list] par-file-name
```

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or user ID (`uid`). Enclose value in double quotation marks if it includes spaces.

`[--passwordfile | -f]` *password-file-name*
Specifies administrator's password in the password file.

`[--portal | -p]` *portal-name*
Specifies the portal name.

`[--type | -t]` *type*
Specifies the type of data to be exported. Supports two types: `desktop` and `provider`. Note that the `provider` type must be used for channel as well as provider data.

The following argument is required:

par-file-name Specifies the name of the PAR file, located in the local host. If a PAR file with this name already exists, an error message is displayed.

The following options are optional:

`--files-only`	Specifies to back up the desktop file system data only. Default is `off`.
`--dp-only`	Specifies to export the display profile data only. Default is `off`.
`--export | -x` *export_file_list*	Specifies a comma separated list of export files. Applicable only if type is `provider` which is used for exporting provider and/or channels data as specified in the export file. Each export file is a text file (`.txt`), corresponds to an entry (provider, channel, or provider/channel combination) in the

	`.par` file, and specifies the data to be inserted into the specified `.par` file. See supplement 37 PAR Export File Format for export file format.
`--dn | -d` *dn*	Specifies the distinguished node from where data is to be exported. Applicable only if type is `provider`. This option is mutually exclusive to `--global`.
`--global | -g`	Specifies the global level node to access the display profile document. Applicable only if type is `provider`. This option is mutually exclusive to `--dn`.

Example

```
psadmin export -u amadmin --passwordfile /tmp/passwd -t desktop
desktop.par
```

```
psadmin export -u amadmin --passwordfile /tmp/passwd -t provider
-d "ou=DeveloperSample,dc=someplace,dc=siroe,dc=com" -x
expfile1.txt,expfile2.txt desktop.par
```

psadmin import

Description

Imports portal data. The `import` subcommand requires an archive file (PAR file) in the local system.

Syntax

Long Format

`psadmin import --adminuser` *uid* `--passwordfile` *password-file-name* `[--files-only] --portal` *portal-name* `[--continue] [--dp-only] [--overwrite] [--dp-node` *dpnode*`] [--redeploy] [--operations` *operations-list*`]` *par-file-name*

Short Format

`psadmin import -u` *uid* `-f` *password-file-name* `[--files-only] -p` *portal-name* `[-c] [--overwrite] [--dp-only] [--dpnode` *dpnode*`] [-D] [-O` *operations-list*`]` *par-file-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or user ID (`uid`).

`[--passwordfile | -f]` *password-file-name*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-name*
Specifies the portal name.

The following argument is required:

par-file-name Specifies the name of the PAR file, which is located in the local host.

The following options are optional:

`--files-only`	Specifies to desktop file system data only. Default is `off`.
`[--continue | -c]`	Specifies that the import process should continue if errors indicate that the storage structure of the archive file and the current system differ. Default is `false`.
`--dp-only`	Specifies to import display profile data only. Default is `off`.
`--overwrite`	Specifies whether to overwrite the display profile document and file system files, if data already exists. Default is `false`. If `true`, the data profile document is stored in the LDAP tree, replacing the original version, and file system data are overwritten. If `false`, the data profile document is combined with the data profile in the LDAP tree. File system data are not overwritten, if the same named file exists.
`--dpnode`	Specifies the base `dn` for the display profiles and service attributes to be imported. If the base `dn` is not specified, the display profile node defined inside the PAR file is used. If the PAR file does not provide a definition, the portal's default organization `dn` is used.
`[--redeploy | -D]`	Specifies whether to redeploy the Portal Server instances. By default, this is set to `false`.
`[--operations | -O` *operations-list*`]`	Specifies a space separated list of operations where each operation

constitutes of | (pipe) separated list of keywords that can have values most of which are optional. This option is applicable only when the imported PAR file is of type `provider`. The operations list is in the following format:

```
"dpnode=dn|entry=name|provider[=name]|channel[=name]|
"dpnode=dn|entry=name|provider[=name]|channel[=name]|
```

If this option is not specified, the operations as specified in `PS-AutoExtract` entry in PAR file is used for import. For more information on the operations format, see supplement 38 Import Operations Format.

Example

```
psadmin import -u amadmin --passwordfile /tmp/passwd dp_org.par
```

```
psadmin import -u amadmin --passwordfile /tmp/passwd --dpnode
"ou=DeveloperSample,dc=someplace,dc=siroe,dc=com" -O '
"entry=mychannel|channel=anothername|avail=topcontainer"
"entry=yourchannel|channel=anothername|avail=topcontainer" '
dp_providers.par
```

psadmin list-par

Description

Used to describe a PAR file. The `list-par` subcommand requires a PAR file located in the local system. If none of the optional options are specified (see "Options" on page 124 for more information), the `list-par` subcommand lists all the directories as specified in PAR file manifest.

Syntax

Long Format

`psadmin list-par --adminuser` *uid* `--passwordfile` *passwordFile* `--portal` *portal-name* `[--pbfiles-only] [--dp-only] [--war-only] [--static-only]` *par-file-name*

Short Format

`psadmin list-par -u` *uid* `-f` *passwordFile* `-p` *portal-name* `[--pbfiles-only] [--dp-only] [--war-only] [--static-only]` *par-file-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or user ID (`uid`).

`[--passwordfile | -f]` *password-file-name*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-name*
Specifies the portal name.

The following argument is required:

par-file-name Specifies the name of the PAR file, which is located in the local host.

The following options are optional:

`--pbfiles-only`	Specifies if only `pbfiles` directory is to be listed. By default, this is turned off.
`--war-only`	Specifies if only WAR directory is to be listed. By default, this is turned off.
`--dp-only`	Specifies if only `dp` directory is to be listed. By default, this is turned off.
`--static-only`	Specifies if only static directory is to be listed. By default, this is turned off.

psadmin schedule-tasks

Description

Designates commands to be run at specified time.

Syntax

Long Format

`psadmin schedule-tasks --adminuser` *uid* `--passwordfile` *password-file-name* `--commandfile` *command-file*

Short Format

`psadmin schedule-tasks -u` *uid* `-f` *password-file-name* `--commandfile` *command-file*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or user ID (`uid`).

`[--passwordfile | -f]` *password-file-name*
Specifies the administrator's password in the password file.

The following options are optional:

`--commandfile` *command-file* — Specifies the full path of the command file that designates the names and execution schedules of commands.

The command format is `command -options | day_of_week@hour:minute;day_of_week@hour:minute`. For `day_of_week`, valid values are `0` for Sunday through `6` for Saturday. For `hour`, valid values are `0` through `23`. For `minute`, valid values are `0` through `59`.

Example

```
psadmin schedule-tasks-a amadmin --passwordfile /tmp/passwd
--commandfile file1
```

with the content of the `file1` file as:

```
./psadmin export -u amadmin -f /tmp/password dp_org.par | 1@23:30
;
```

psadmin unschedule-tasks

Description

Cancels commands that are designated to run.

Syntax

Long Format

`psadmin unschedule-tasks --adminuser` *uid* `--passwordfile` *password-file-name* `--commandfile` *command-file*

Short Format

`psadmin unschedule-tasks -u` *uid* `-f` *password-file-name* `--commandfile` *command-file*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or user ID (`uid`).

`[--passwordfile | -f]` *password-file-name*
Specifies the administrator's password in the password file.

The following option is optional:

`--commandfile` *command-file* — Specifies the full path of the command file that designates the names and execution schedules of commands.

The command format is `command -options | day_of_week@hour:minute;day_of_week@hour:minute`. For `day_of_week`, valid values are `0` for Sunday through `6` for Saturday. For `hour`, valid values are `0` through `23`. For `minute`, valid values are `0` through `59`.

Example

```
psadmin unschedule-tasks-u amadmin --passwordfile /tmp/passwd
--commandfile file1
```

with the content of the `file1` file as:

```
./psadmin export -u amadmin -f /tmp/password dp_org.par | 1@23:30
;
```

psadmin set-domain-repository-password

Description

Sets the password used by the Portal Administration Server instance to log into the portal domain repository.

Syntax

Long Format

`psadmin set-domain-repository-password --adminuser` *uid* `--passwordfile` *password-file-name domainRepositoryPasswordFile*

Short Format

`psadmin set-domain-repository-password -u` *uid* `-f` *password-file-name domainRepositoryPasswordFile*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or user ID (`uid`).

`[--passwordfile | -f]` *password-file-name*
Specifies the administrator's password in the password file.

The following argument is required:

domainRepositoryPasswordFile Specifies the file that contains the portal domain repository password.

psadmin version

Description

Provides Portal Server version information.

Syntax

Long Format

`psadmin --version [--display] [--patches] [--jar` *jar-file-name*`]`

Short Format

`psadmin -V -u` *uid* `-f` *password-file-name* `[--display] [--patches] [--jar` *jar-file-name*`]`

Options

The following options are optional:

`--display` Specifies whether to display the portal's version information. Default is `true`.

`--patches` Specifies whether to list Portal Server patch information. Default is `false`.

`--jar` *jar-file-name* Specifies whether to display the JAR file's version information.

CHAPTER 22

Logging Subcommands

The logging subcommands configure parameters for the following applications:Portal Server, Search web application, Portal Administration Server, Administration Console and Portal Server Secure Remote Access Gateway, Netlet proxy, and Rewriter proxy. Changes you make using the logging subcommands are persistent. This section describes the following subcommands:

list-loggers

Description

Lists all the loggers.

Syntax

Long Format

`psadmin list-loggers --adminuser` *uid* `--passwordfile` *password-filename* `--component` *component-type* `[--portal` *portal-ID*`] [ --instance` *portal-instance-name*`] [ --searchserver` *search-server-ID*`] [--instance-name` *sra-instance-name*`][--detail] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin list-loggers -u` *uid* `-f` *password-filename* `-m [-p` *portal-ID*`] [-i` *portal-instance-name*`] [-s` *search-server-ID*`] [-o` *output-filename* `] [-v]`

Options

The following options are required:

[`--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.

[`--component | -m`]
Specifies the component type. Valid values are: portal, search, pas, gateway, nlproxy, rwproxy.

The following options are not required:

[`--portal	-p`] *portal-ID*	Specifies the portal ID. This is required only if the `--component-type` is `portal`.
[`--instance	-i`] *portal-instance-name*	Specifies the Portal Server instance. This is required only if the `--component-type` is `portal`.
[`--searchserver	-s`] *search-server-ID*	Specifies the search server ID. This is required only if the the `--component-type` is `search`.
[`--instance-name`] *sra-instance-name*	Specifies the SRA instance name. This is required only if the `--component-type` is gateway, nlproxy, or rwproxy.
[`--output`] *output-filename*	Names the output file.
[`--verbose	-v`]	Lists loggers verbosely.

Examples

EXAMPLE 22–1 list-loggers

```
./psadmin list-loggers -u amadmin -f passwordfile -m portal -p
myPortal -i myInstance
```

the output is:

```
debug.com.sun.portal
debug.com.sun.portal.desktop
debug.com.sun.portal.desktop.context
debug.com.sun.portal.desktop.monitoring
debug.com.sun.portal.monitoring
debug.com.sun.portal.portletappengine
debug.com.sun.portal.portletappengine.imp
debug.com.sun.portal.ubt
debug.com.sun.portal.wsrp.consumer.common
Successfully completed!
```

EXAMPLE 22–2 list-loggers with detail

```
./psadmin list-loggers -u amadmin -f passwordfile -m portal -p
myPortal -i myInstance --detail
```

the output is:

```
debug.com.sun.portal
Level=FINEST    Use Parent Handler=false
Handler=java.util.logging.FileHandler
Logs to File=/var/opt/SUNWportal/portals/myPortal/logs/
%InstanceID/%logger.%u.%g.log Print Stacktrace=true
debug.com.sun.portal.desktop
Level=FINEST    Use Parent Handler=true
debug.com.sun.portal.desktop.context
Level=FINEST    Use Parent Handler=true
debug.com.sun.portal.desktop.monitoring
Level=FINEST    Use Parent Handler=true
debug.com.sun.portal.monitoring
Level=FINEST    Use Parent Handler=true
debug.com.sun.portal.portletappengine
Level=FINEST    Use Parent Handler=true
debug.com.sun.portal.portletappengine.impl
Level=FINEST    Use Parent Handler=true
debug.com.sun.portal.ubt
Level=FINEST    Use Parent Handler=true
debug.com.sun.portal.wsrp.consumer.common
Level=FINEST    Use Parent Handler=true
Successfully completed!
```

set-logger

Description

Sets the level for logger and also separate file for the logger.

Syntax

Long Format

`psadmin set-logger --adminuser` *uid* `--passwordfile` *password-filename* `--component` *component-type* `[--portal` *portal-ID*`] [--instance` *portal-instance-name*`] [--searchserver` *search-server-ID*`] [--instance-name` *sra-instance-name*`] --name` *logger-name* `--level` *level* `[--file {--stack-trace | --parent } ] [--output` *output-filename* `] [--verbose]`

Short Format

`psadmin set-logger -u` *uid* `-f` *password-filename* `-m` *component-type* `[-p` *portal-ID*`] [-i` *portal-instance-name* `][-s` *search-server* `] -O` *logger-name* `-L` *level* `[ -F{-T | -P}] -o` *output-filename*`] [- v]`

Options

The following options are required:

[`--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.

[`--component | -m`]
Specifies the component type. Valid values are: portal, search, pas, gateway, nlproxy, rwproxy.

[`--logger | -O`] *logger-name*
Specifies the name of the logger.

[`--level | -L`] *level*
Specifies the level.

The following options are optional:

[`--file	-F`]	Specifies if the logger is to be logged to a separate file.
[`--stack-trace	-T`]	This option can be specified only if the `--file` option is specified. Specifies whether the stack trace is to be printed in the log file. If this option is specified, `--parent` option cannot be specified. The default is false. If the user specifies true, then the stack trace will be printed in the logger's log file.
[`--parent	-P`]	This option can be specified only if the `--file` option is specified. Specifies whether the log data is to be printed in the parent log file of the current logger. If this option is specified, `--stack-trace` option cannot be specified. The default is false, if the user does not specify it, the log data will be printed only in the current logger's log file. If the user specifies true, the log data will be printed in the parent log file and also in the current logger's log file.
[`--portal	-p`] *portal-ID*	Specifies the portal ID. This is required only if component-type is portal.
[`--instance	-i`] *portal-instance-name*	Specifies the portal server instance. This is required only if component-type is portal.

[`--searchserver	-s`] *search-server-ID*	Specifies the search server ID. This is required only if the the `--component-type` option is `search`.
[`--instance-name`] *sra-instance-name*	Specifies the Portal Server Secure Remote Access instance. This is required only if the `--omponent-type` is **`gateway`**, **`nlproxy`**, or **`rwproxy.`**
[`--verbose	-v`]	Lists loggers verbosely.

EXAMPLE 22–3 set-logger level

Obtain the logger name from list-logger subcommand.

Set the level of the logger.

```
./psadmin set-logger -u amadmin -f passwordfile -m portal -p
myPortal -i myInstance -O debug.com.sun.portal.desktop -L INFO
```

EXAMPLE 22–4 set-logger for separate file

Set a separate file and level for the logger. In this new file, no stack trace is logged and no content is logged to the parent (the log related to `debug.com.sun.portal.desktop` is not logged into the file associated with `debug.com.sun.portal`):

```
./psadmin set-logger -u amadmin -f passwordfile -m portal -p
myPortal -i myInstance -O debug.com.sun.portal.desktop -L INFO -F
```

the output is:

```
Successfully completed!
```

EXAMPLE 22–5 set-logger for stack trace

Set a separate file and level for the logger. In this new file, stack trace will be logged. But log content will not logged to the parent (the log related to `debug.com.sun.portal.desktop` will not be logged into the file associated with `debug.com.sun.portal`)

```
./psadmin set-logger -u amadmin -f passwordfile -m portal -p
myPortal -i myInstance -O debug.com.sun.portal.desktop -L INFO -F
-T
```

output is:

```
Successfully completed!
```

reset-logger

Description

Resets the log level and log file to that of the root logger.

Syntax

Long Format

`psadmin reset-logger --adminuser` *uid* `--passwordfile` *password-filename* `--component` *component-type* `[--portal` *portal-ID*`] [--instance` *portal-instance-name*`] [--searchserver` *search-server-ID*`] [--instance-name` *sra-instance-name*`]--name` *logger-name* `[--output` *output-filename*`] [--verbose]`

Short Format

`psadmin reset-logger -u` *uid* `-f` *password-filename* `-m` *component-type* `[-p` *portal-ID*`] [--instance` *portal-instance-name*`] [-s` *search-server*`] [-i` *instance-name*`] [-O` *logger-name*`][ -o` *output-filename*`] [-v]`

Options

The following options are required:

[`--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.

[`--logger | -O`] *logger-name*
Specifies the name of the logger.

The following options are not required:

[`--portal | -p`] *portal-ID*
Specifies the portal ID. This is required only if component-type is portal.

[`--instance | -i`] *portal-instance-name*
Specifies the portal server instance. This is required only if component-type is portal.

[`[--searchserver | -s`] *search-server-ID*
Specifies the search server ID. This is required only if the `--component-type` option value is `search`.

[`--instance-name`] *sra-instance-name*
Specifies the Portal Server Secure Remote Access instance. This is required only if the `--component-type` option value is either `gateway` or `nlproxy` or `rwproxy`.

[--verbose | -v]
Lists output filenames verbosely.

EXAMPLE 22–6 reset-logger

Obtain the logger-name using the list-logger subcommand.

```
./psadmin reset-logger -u amadmin -f passwordfile -p myPortal -i
myInstance -O debug.com.sun.portal.desktop
```

The logs no longer go to the logger's log file, instead they go to the parent's log file with the same level as the parent. The log related to `debug.com.sun.portal.desktop` is only logged into the file associated with `debug.com.sun.portal`)

outputs:

```
Successfully completed!
```

Location of Log Files

The location of log files is as follows:

Portal Instance

Log Config Filename and Location

ps-data-directory`/portals/`*portal-ID*`/config/PSLogConfig.properties`

Example

`/var/opt/SUNWportal/portals/myPortal/config/PSLogConfig.properties`

Logfile Location

ps-data-directory`/portals/`*portal-ID*`/logs/`*instance-ID*

By default only one file is created - `portal.admin.portal.0.0.log`

Portlet Applications

Log Config Filename and Location

ps-data-directory/portals/*portal-ID*/config/PSLogConfig.properties

Example

/var/opt/SUNWportal/portals/myPortal/config/PSLogConfig.properties

Logfile Location

ps-data-directory/portlets/*portal-ID*/logs/*instance-ID*

By default only one file is created - portal.0.0.log

Search Webapp

Log Config Filename and Location

ps-data-directory/searchserver/
search-ID/config/SearchLogConfig.properties

Example

/var/opt/SUNWportal/searchserver/mySearch/config/SearchLogConfig.properties

Logfile Location

ps-data-directory/searchserver/ *search-ID*/logs/

By default three files will be created - rdmserver.0.0.log, rdm.0.0.log, rdmgr.0.0.log

Admin CLIs

Log Config Filename and Location

ps-config-directory/PSAdminLogConfig.properties

Example

/etc/opt/SUNWportal/PSAdminLogConfig.properties

Logfile Location

ps-data-directory/logs/admin

By default only one file is created - portal.admin.cli.0.0.log

Admin Server

Log Config Filename and Location

ps-config-directory/PSAdminLogConfig.properties

Example

/etc/opt/SUNWportal/PSAdminLogConfig.properties

Logfile Location

ps-data-directory/logs/admin

By default only one is created - portal.0.0.log

Admin Console

Log Config Filename and Location

ps-data-directory/portals/*portal-ID*/config/PSLogConfig.properties

Example

/var/opt/SUNWportal/portals/myportal/config/PSLogConfig.properties

Logfile Location

ps-data-directory/logs/admin

By default only one is created - portal.admin.portal.0.0.log

Gateway

Log Config Filename and Location

ps-config-directory/platform.conf.*gateway-profile-name*

Example

/etc/opt/SUNWportal/platform.conf.*gateway-profile-name*

Logfile Location

ps-data-directory/logs/sra/*gateway-profile-name*

By default only one file is created - portal.gateway.0.0.log

Netlet Proxy

Log Config Filename and Location

ps-config-directory/platform.conf.*gateway-profile-name*

Example

`/etc/opt/SUNWportal/platform.conf`*gateway-profile-name*

Logfile Location

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

By default only one file is created - `portal.nlproxy.0.0.log`

Rewriter Proxy

Log Config Filename and Location

ps-config-directory`/platform.conf.`*gateway-profile-name*

Example

`/etc/opt/SUNWportal/platform.conf.`*gateway-profile-name*

Logfile Location

ps-data-directory`/logs/sra/`*gateway-profile-name*

By default only one file is created - `portal.rwproxy.0.0.log`

CHAPTER 23

User-Behavior Tracking Subcommands

This section describes the following commands:

list-ubt-report

Description

Lists all the reports that are available to be generated.

Syntax

Long Format

```
list-ubt-report
```

Short Format

```
list-ubt-report
```

Options

This sub-command has no options.

generate-ubt-report

Description

Generates the selected report.

Syntax

Long Format

```
generate-ubt-report --adminuser uid --passwordfile
password-filename
 --portal portal-ID  --destinationfolder
folder-name
   [--reportname report-name] [--format format
]
```

Short Format

```
generate-ubt-report -u uid -f
password-filename
   [-reportname report-name]
 -p portal-ID -destinationfolder
folder-name [--format format]
```

Options

The following options are required:

[`--adminuser | -u`] *uid*
Specifies the administrator's distinguished name.

[`--passwordfile | -f`] *password-filename*
Specifies the administrator's password in the password file.

[`--portal | -p`] *portal-ID*
Specifies the portal ID.

[`--destinationfolder`] *folder-name*
Specifies the name of the destination folder into which report is generated and stored.

[`--format`] *format*
Specifies the format of the report, such as PDF, HTML, XML. If not provided, the report is stored in PDF format.

CHAPTER 24

WSRP Producer Subcommands

This supplement contains the following:

psadmin create-producer

Description

Create a Producer in a portal.

Syntax

Long Format

`psadmin create-producer --portal` *portal-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--dn` *orgnaization-DN* `--producer` *producer-name*

Short Format

`psadmin create-producer -p` *portal-ID* `-u` *uid* `-f` *password-filename* `-d` *orgnaization-DN* `-N` *producer-name*

Options

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--dn | -d]` *organization-DN*
Specifies the distinguished name for the organization where the Producer is to be created.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

`[--producer | -N]` *producer-name*
Specifies the name of the Producer to be created.

psadmin delete-producer

Description

Delete a Producer in a portal.

Syntax

Long Format

`psadmin delete-producer --adminuser` *uid* `--passwordfile` *password-filename* `--portal` *portal-ID* `--producer` *producer-name*

Short Format

`psadmin delete-producer -u` *uid* `-f` *password-filename* `-p` *portal-ID* `-N` *producer-name*

Options

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

`[--producer | -N]` *producer-name*
Specifies the name of the Producer to be deleted.

psadmin list-producers

Description

List all Producers for a portal.

Syntax

Long Format

`psadmin list-producers --adminuser` *uid* `--passwordfile` *password-filename* `[--dn` *organization-DN*`] --portal` *portal-ID*

Short Format

`psadmin list-producers -u` *uid* `-f` *password-filename* `[-d` *organization-DN*`] -p` *portal-ID*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

The following option is optional:

`[--dn | -d]` *organization-DN*	Specifies the distinguished name for the organization from where the producers are to be listed. If none is specified, all the producers that belong to the portal are listed.

psadmin list-attributes

Description

List the attributes for a Producer.

Syntax

Long Format

`psadmin list-attributes --component producer --adminuser` *uid* `--passwordfile` *password-filename* `--portal` *portal-ID* `[--consumerregistration]`

Short Format

`psadmin list-attributes -m producer -u` *uid* `-f` *password-filename* `-p` *portal-ID* `[-S]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--component | -m] producer`
Specifies the name of the component. The value should be `producer`.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

The following options are optional:

`[--consumerregistration | -S]`	Specifies that this is for a consumer registration.

psadmin get-attribute

Description

Display the value of an attribute for a Producer.

Syntax

Long Format

`psadmin get-attribute --component producer --portal` *portal-ID* `--attribute-name` *attribute-name* `--adminuser` *uid* `--passwordfile` *password-filename* `[--producer` *producer-name*`] [--registrationhandle` *handle*`]`

Short Format

`psadmin get-attribute -m producer -p` *portal-ID* `-a` *attribute-name* `-u` *uid* `-f` *password-filename* `[-N` *producer-name*`] [-H` *handle*`]`

Options

The following options are required:

`[--attribute-name | -a]` *attribute-name*
Specifies the name of the Producer attribute for which the value is to be retrieved.

Attributes can be:

- `AllDisabled` (`true` or `false`, `global attribute`)
- `Enabled` (`true` or `false`)
- `WSDL_URL (read-only)`
- `RegistrationRequired` (`true` or `false`)

- `InBandRegistrationSupported` (`true` or `false`)
- `RegistrationValidatorClassName`
- `PublishedPortlets`
- `UnpublishedPortlets (read-only)`
- `RegistrationPropertyDescription`

Attributes for Consumer Registrations can be:

- `Enabled` (`true` or `false`)
- `ConsumerName`
- `ConsumerAgent`
- `MethodGetSupported`
- `ConsumerModes`
- `ConsumerWindowStates`
- `ConsumerUserScopes`
- `CustomUserProfileData`
- `RegistrationProperties`

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--component | -m] producer`
Specifies the name of the component. The value should be `producer`.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portalid | -p]` *portal-ID*
Specifies the portal ID.

The following options are optional:

`[--producer | -N]` *producer-name*
Specifies the name of the Producer being targeted.

`[--registrationhandle | -H]` *handle*
Specifies the registration handle.

psadmin set-attribute

Description

Set the value of an attribute for a Producer.

Syntax

Long Format

`psadmin set-attribute --component producer --portal` *portal-ID* `--attribute-name` *attribute-name* `--adminuser` *uid* `--passwordfile` *password-filename* `[--producer` *producer-name*`] [--registrationhandle` *handle*`] [--add] [--remove]` *values*

Short Format

`psadmin set-attribute -m producer -p` *portal-ID* `-a` *attribute-name* `-u` *uid* `-f` *password-filename* `[-N` *producer-name*`] [-H` *handle*`] [-A] [-E]` *values*

Options

`[--component | -m] producer`
Specifies the name of the component. The value should be `producer`.

`[--portalid | -p]` *portal-ID*
Specifies the portal ID.

`[--attribute-name | -a]` *attribute-name*
Specifies the name of the Producer attribute for which the value is to be retrieved.

Attributes can be:

- AllDisabled (`true` or `false`, global attribute)
- Enabled (`true` or `false`)
- WSDL_URL (read-only)
- RegistrationRequired (`true` or `false`)
- InBandRegistrationSupported (`true` or `false`)
- RegistrationValidatorClassName
- PublishedPortlets
- UnpublishedPortlets (read-only)
- RegistrationPropertyDescription

Attributes for Consumer Registrations can be:

- Enabled (`true` or `false`)
- ConsumerName
- ConsumerAgent
- MethodGetSupported
- ConsumerModes
- ConsumerWindowStates
- ConsumerUserScopes
- CustomUserProfileData
- RegistrationProperties

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

The following options are optional:

`[--producer | -N]` *producer-name*
Specifies the name of the Producer being targeted.

`[--registrationhandle | -H]` *handle*
Specifies the registration handle.

`[--add | -A] [--remove | -E]` *values*
Specifies values to add or remove.

psadmin list-consumer-registrations

Description

List the Consumers that are currently registered with a Producer.

Syntax

Long Format

`psadmin list-consumer-registrations --portal` *portal-ID* `--producer` *producer-name* `--adminuser` *uid* `--passwordfile` *password-filename*

Short Format

`psadmin list-consumer-registrations -p` *portal-ID* `-N` *producer-name* `-u` *uid* `-f` *password-filename*

Options

The following options are required:

`[--producer | -N]` *producer-name*
Specifies the name of the Producer being targeted.

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

psadmin create-consumer-registration

Description

Create a new Consumer registration at a Producer.

Syntax

Long Format

`psadmin create-consumer-registration --portal` *portal-ID* `--producer` *producer-name* `--registrationdatafile` *file* `--adminuser` *uid* `--passwordfile` *password-filename*

Short Format

`psadmin create-consumer-registration -p` *portal-ID* `-N` *producer-name* `-x` *file* `-u` *uid* `-f` *password-filename*

Options

The following options are required

`[--producer | -N]` *producer-name*
Specifies the name of the Producer being targeted.

`[--registrationdatafile | -x]` *file*
Specifies the properties file that lists the options to set on the new consumer registration.

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

psadmin delete-consumer-registration

Description

Delete a Consumer registration from a Producer.

Syntax

Long Format

`psadmin delete-consumer-registration --producer` *producer-name* `--adminuser` *uid* `--passwordfile` *password-filename* `--portal` *portal-ID* `--registrationhandle` *handle*

Short Format

`psadmin delete-consumer-registration -N` *producer-name* `-u` *uid* `-f` *password-filename* `-p` *portal-ID* `-H` *handle*

Options

The following options are required

`[--producer | -N]` *producer-name*
Specifies the name of the Producer being targeted.

`[--registrationhandle | -H]` *handle*
Specifies the registration handle that is to be deleted.

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

psadmin publish-registry

Description

Publish Producer and Remote Portlet artifacts to the ebXML registry.

Syntax

Long Format

`psadmin publish-registry --component producer --adminuser` *uid* `--passwordfile` *password-filename* `--portal` *portal-ID* `--producerdatafile` *file* `[--portlets` *portlets*`] [--allportlets] [--orgdatafile` *org-data-file*`]`

Short Format

`psadmin publish-registry -m producer -u` *uid* `-f` *password-filename* `-p` *portal-ID* `-U` *file* `[-T` *portlets*`] [-L] [-O` *org-data-file*`]`

Options

The following options are required:

`[--component | -m] producer`
Specifies the name of the component. The value should be `producer`.

`[--producerdatafile | -U]` *file*
Contains the following attributes:

`producer.name=PSAdmin Producer`
Name of the Producer to be published to the registry.

`producer.description=PSAdmin Producer from Portal Server`
Description of the Producer

`producer.id=psadminCLI`
Producer ID as it exists in the LDAP

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portalid | -p]` *portal-ID*
Specifies the portal ID; if this is not supplied, the default is used.

The following options are optional:

`[--portlets | -T]` *portlets*

List of portlets to be published. The list is specified within double-quotes space separated items. For example, `NotepadPortlet BookmarkPortlet WeatherPortlet` .

`[--orgdatafile | -O]` *org-data-file*

Organization to be published. The `orgdata` is the file specifying the details of the organization such as:

- `org.name=PSAdmin Organization`
- `org.description=Organization that offers Sun Portal WSRP Services`
- `org.primarycontact.name=anyuser`
- `org.primarycontact.phoneno=2298989`
- `org.primarycontact.email=anyuser@anydomain.com`
- `org.key=urn:cli:organization:cli`

Note – The `org.key` is a special attribute. If this is specified, all other attributes are ignored and the system searches for an organization that is already published in the registry with a key as specified in the value of this attribute.

`[--allportlets | -L]`

Publishes all the portlets that are offered by the Producers specified in the `producer.id` attribute. This also ignores the `--portlets` options.

CHAPTER 25

WSRP Consumer Subcommands

This supplement contains the following:

psadmin list-attributes

Description

List the attributes for a Consumer.

Syntax

Long Format

`psadmin list-attributes --adminuser` *uid* `--passwordfile` *password-filename* `--component consumer --portal` *portal-ID* `[--configuredproducer]`

Short Format

`psadmin list-attributes -u` *uid* `-f` *password-filename* `-m consumer -p` *portal-ID* `[-R]`

Options

The following options are required:

`[--component | -m]` *consumer*
Specifies the name of the component. The value should be `consumer`.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

The following option is optional:

`[--configuredproducer | -R]` Specifies that this is for a configured producer.

psadmin get-attribute

Description

Display the value of an attribute for a Consumer.

Syntax

Long Format

`psadmin get-attribute --component consumer --portal` *portal-ID* `--attribute-name` *attribute-name* `--adminuser` *uid* `--passwordfile` *password-filename* `[--dn` *organization-DN*`] [--producer` *producer-id* `]]`

Short Format

`psadmin get-attribute -m consumer -p` *portal-ID* `-a` *attribute-name* `-u` *uid* `-f` *password-filename* `[-d` *organization-DN*`] [-I` *producer-id*`]]`

Options

The following options are required:

`[--component | -m] consumer`
Specifies the name of the component. Value should be `consumer`.

`[--attribute-name | -a]` *attribute-name*
Specifies the name of the Consumer attribute for which the value is to be retrieved.

Attributes for the Consumer can be:

- `AllDisabled ("true" or "false", global attribute)`
- `UserProfileMapping`
- `Name`

Attributes for the Configured Producers can be:

- Name
- Enabled ("true" or "false")
- WSDL_URL
- UserCategoryMapping
- RegistrationProperties

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

The following options are optional:

`[--dn | -d]` *organization-DN*	Specifies the distinguished name for the organization from where the attributes are to be listed.
`[--producer | -I]` *producer-id*	Specifies the ID of the configured producer to fetch the attribute for.

psadmin set-attribute

Description

Set the value of an attribute for a Consumer.

Syntax

Long Format

`psadmin set-attribute --component consumer --portal` *portal-ID* `--attribute-name` *attribute-name* `--adminuser` *uid* `--passwordfile` *password-filename* `[--dn` *organization-DN*`] [--producer` *producer-id* `] [--add] [--remove]` *values*

Short Format

`psadmin set-attribute -m consumer -p` *portal-ID* `-a` *attribute-name* `-u` *uid* `-f` *password-filename* `[-d` *organization-DN*`] [-I` *producer-id*`] [-A] [ -E]` *values*

Options

The following options are required:

`[--component | -m]` *consumer*
Specifies the name of the component. Value should be `consumer`.

`[--attribute-name | -a]` *attribute-name*
Specifies the name of the Consumer attribute for which the value is to be retrieved.

Attributes for the Consumer can be:

- `AllDisabled ("true" or "false", global attribute)`
- `UserProfileMapping`
- `Name`

Attributes for the Configured Producers can be:

- `Name`
- `Enabled ("true" or "false")`
- `WSDL_URL`
- `UserCategoryMapping`
- `RegistrationProperties`

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

The following options are optional:

`[--dn | -d]` *organization-DN*	Specifies the distinguished name for the organization from where the attributes are to be listed.
`[--producer | -I]` *producer-id*	Specifies the ID of the configured producer to fetch the attribute for.
`[--add | -A][--remove | -E]` *values*	Specifies the values to be added or removed.

psadmin list-configured-producers

Description

List the configured Producers at a Consumer.

Syntax

Long Format

`psadmin list-configured-producers --portal` *portal-ID* `--dn` *organization-DN* `--adminuser` *uid* `--passwordfile` *password-filename*

Short Format

`psadmin list-configured-producers -p` *portal-ID* `-d` *organization-DN* `-u` *uid* `-f` *password-filename*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--dn | -d]` *organization-DN*
Specifies the distinguished name for the organization from where the attributes are to be listed.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

psadmin create-configured-producer

Description

Create a new configured Producer at a Consumer.

Syntax

Long Format

`psadmin create-configured-producer --portal` *portal-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--dn` *organization-DN* `--producer` *producer-name* `--url` *producer-URL* `[--registrationdata` *file* `| --registrationhandle` *handle*`]`

Short Format

`psadmin create-configured-producer -p` *portal-ID* `-u` *uid* `-f` *password-filename* `-d` *organization-DN* `-N` *producer-name* `-Z` *producer-URL* `[-x` *file* `| -H` *handle*`]`

Options

The following options are required:

`[--producer | -N]` *producer-name*
Specifies the name to assign to the Producer being created.

`[--url | -Z]` *producer-URL*
Specifies the URL of the Producer to be configured.

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--dn | -d]` *organization-DN*
Specifies the distinguished name for the organization from where the attributes are to be listed.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

The following options are optional:

`[--registrationdata | -x]` *filename*	Specifies the properties filename that will contain the registration properties for the producer being configured.
`[--registrationhandle | -H]` *handle*	Specifies the registration handle of the Producer being configured.

psadmin delete-configured-producer

Description

Delete a configured Producer at a Consumer.

Syntax

Long Format

`psadmin delete-configured-producer --adminuser` *uid* `--passwordfile` *password-filename* `--dn` *organization-DN* `--portal` *portal-ID* `--producer` *producer-id*

Short Format

`psadmin delete-configured-producer -u` *uid* `-f` *password-filename* `-d` *organization-DN* `-p` *portal-ID* `-I` *producer-id*

Options

The following options are required:

`[--producer | -I]` *producer-id*
Specifies the ID of the configured producers that are to be deleted.

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--dn | -d]` *organization-DN*
Specifies the distinguished name for the organization from where the attributes are to be listed.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

psadmin update-configured-producer-service-description

Description

Update the service description for a configured Producer in a Consumer.

Syntax

Long Format

`psadmin update-configured-producer-service-description --adminuser` *uid* `--passwordfile` *password-filename* `--dn` *organization-DN* `--portal` *portal-ID* `--producer` *producer-id*

Short Format

`psadmin update-configured-producer-service-description -u` *uid* `-f` *password-filename* `-d` *organization-DN* `-p` *portal-ID* `-I` *producer-id*

Options

The following options are required:

`[--producer | -I]` *producer-id*
Specifies the ID of the configured Producer to update the Service Description for.

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--dn | -d]` *organization-DN*
Specifies the distinguished name for the organization from where the attributes are to be listed.

`[--portal | -p]` *portal-ID*
Specifies the portal ID.

psadmin search-registry

Description

Searches for WSRP artifacts in the ebXML repository.

Syntax

Long Format

`psadmin search-registry --component consumer --adminuser` *uid* `--passwordfile` *password-filename* `--portal` *portal-ID* `[--searchproducerdatafile` *file* `| --searchportletdatafile` *file* `| --searchorgdatafile` *file*`]`

Short Format

`psadmin search-registry -m consumer -u` *uid* `-f` *password-filename* `-p` *portal-ID* `[-C` *file* `| -D` *file* `| -L` *file*`]`

Options

The following options are required:

`[--component | -m] consumer`
Specifies the name of the component. Value should be `consumer`.

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name.

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--portal | -p]` *portalid*
Specifies the portal ID.

The following options are optional:

`[--searchproducerdatafile | -C]` *file*
Specifies the name of the file that contains the attributes to be searched for while searching for WSRP Producer. Contents of the file include:

- `producer.name ="%ACME Producer%"` - Wildcard search
- `producer.description="%stocks%"` - Wildcard search

`[--searchportletsdatafile | -D]` *file*
Specifies the name of the file that contains the attributes to be searched for while searching for WSRP Portlets. Contents of the file include:

- `portlet.name ="ACME Portlet"`
- `portlet.description="%news%"` - Wildcard search

`[--searchorgdatafile | -L]` *file*
Specifies the name of the file that contains the attributes to be searched for while searching for Organization that offers WSRP service. Contents of the file include:

- `organization.name ="%acme%"`
- `organization.description="%news%"` - Wildcard search

CHAPTER 26

Single Sign-On Subcommands

This supplement contains the following:

psadmin list-ssoadapters

Description

Lists all templates and configurations for single sign-on (SSO) adapter

Syntax

Long Format

`psadmin list-ssoadapters --adminuser` *uid* `--passwordfile` *password-filename* `[--templates] [--configs] [--dn` *dn*`]`

Short Format

`psadmin list-ssoadapters -u` *uid* `-f` *password-filename* `[-T] [-S] [-d` *dn*`]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

The following options are optional:

`[--templates | -T]` *uid*	Lists templates only.
`[--configs | -S]` *password-filename*	Lists configurations only.
`[--dn | -d]` *dn*	The distinguished name from where the configuration will be picked in LDAP. It is to be used in conjunction with the flag `[--configs | -S]`. If unspecified, it defaults to global.

Example

Example for listing only configurations in a dn:

```
./psadmin list-ssoadapters -u amadmin -f ps_password -S -d
uid=devauthlessanonymous,ou=People,o=DeveloperSample,dc=india,dc=sun,dc=com
```

psadmin create-ssoadapter-template

Description

Creates a new SSO adapter template

Syntax

Long Format

`psadmin create-ssoadapter-template --adminuser` *uid* `--passwordfile` *password-filename* `--name` *template-name* `[--template` *existing-template*`]`

Short Format

`psadmin create-ssoadapter-template -u` *uid* `-f` *password-filename* `-N` *template-name* `[-M` *existing-template*`]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--name | -N]` *template-name*
Specifies the name of the new template.

The following option is optional:

`[--template | -M]` *exisiting-template* — Specifies which template provides the pattern for the new template.

Example

Example for creating a template:

```
./psadmin create-ssoadapter-template -u amadmin -f ps-password -N
test-template
```

psadmin get-ssoadapter-template

Description

Displays the details of the specified SSO adapter template

Syntax

Long Format

`psadmin get-ssoadapter-template --adminuser` *uid* `--passwordfile` *password-filename* `--template` *template-name*

Short Format

`psadmin get-ssoadapter-template -u` *uid* `-f` *password-filename* `-M` *template-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile |-f]` *password-filename*
Specifies the administrator's password in the password file.

`[--template | -M]` *template-name*
Specifies which SSO adapter template to display.

Example

Example for getting a template:

```
./psadmin get-ssoadapter-template -u amadmin -f ps-password -M
test-template
```

psadmin delete-ssoadapter-template

Description

Removes the specified SSO adapter template

Syntax

Long Format

`psadmin delete-ssoadapter-template --adminuser` *uid* `--passwordfile` *password-filename* `--template` *template-name*

Short Format

`psadmin delete-ssoadapter-template -u` *uid* `-f` *password-filename* `-M` *template-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--template | -M]` *template-name*
Specifies the name of the template to be removed.

Example

Example for deleting a template:

```
./psadmin delete-ssoadapter-template -u amadmin -f ps-password -M
test-template
```

psadmin create-ssoadapter-config

Description

Creates an SSO adapter configuration

Syntax

Long Format

`psadmin create-ssoadapter-config --adminuser` *uid* `--passwordfile` *password-filename* `--name` *configuration-name* `--template` *existing-template* `[--dn` *dn*`]`

Short Format

`psadmin create-ssoadapter-config -u` *uid* `-f` *password-filename* `-N` *configuration-name* `-M` *existing-template* `[-d` *dn*`]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--name | -N]` *configuration-name*
Specifies the name of the new configuration.

`[--template | -M]` *existing-template*
Specifies which template provides the pattern for the new configuration.

The following option is optional:

`[-dn | -d]` *dn* — The distinguished name where the configuration is created in LDAP. Default is `global`.

Example

Example for creating a configuration in a dn:

```
./psadmin create-ssoadapter-config -u amadmin -f ps-password -N
test-config -M test-template -d
uid=test-user,ou=People,o=DeveloperSample,dc=india,dc=sun,dc=com
```

psadmin get-ssoadapter-config

Description

Displays the details of the specified SSO adapter configuration

Syntax

Long Format

`psadmin create-ssoadapter-config --adminuser` *uid* `--passwordfile` *password-filename* `--config` *configuration-name* `[-dn` *dn*`]`

Short Format

`psadmin create-ssoadapter-config -u` *uid* `-f` *password-filename* `-G` *configuration-name* `[-d` *dn*`]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--config | -G]` *configuration-name*
Specifies which SSO adapter configuration to display.

The following option is optional:

`[--dn | -d]` *dn* The distinguished name where the configuration is created in LDAP. Default is `global`.

Example

Example for getting a configuration's details in a `dn`:

```
./psadmin get-ssoadapter-config -u amadmin -f ps-password -G
test-config -d
uid=test-user,ou=People,o=DeveloperSample,dc=india,dc=sun,dc=com
```

psadmin delete-ssoadapter-config

Description

Removes the specified SSO adapter configuration

Syntax

Long Format

`psadmin delete-ssoadapter-config --adminuser` *uid* `--passwordfile` *password-filename* `--config` *configuration-name* `[-dn` *dn*`]`

Short Format

`psadmin delete-ssoadapter-config -u` *uid* `-f` *password-filename* `-G` *configuration-name* `[-d` *dn*`]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--config | -G]` *configuration-name*
Specifies which SSO adapter configuration to remove.

The following option is optional:

`[--dn | -d]` *dn* The distinguished name where the configuration is created in LDAP. Default is `global`.

Example

Example for deleting a configuration in a dn:

```
./psadmin delete-ssoadapter-config -u amadmin -f ps-password -G
test-config -d
uid=test-user,ou=People,o=DeveloperSample,dc=india,dc=sun,dc=com
```

psadmin create-ssoadapter-property

Description

Creates an SSO adapter property of either the admin or user type.

Syntax

Long Format

`psadmin create-ssoadapter-property --adminuser` *uid* `--passwordfile` *password-filename* `--template` *template-name* `--type [`*admin* `|` *user*`] --name` *property-name* `[--value` *property-value*`] [--encrypt]`

Short Format

`psadmin create-ssoadapter-property -u` *uid* `-f` *password-filename* `-M` *template-name* `-t [`*admin* `|` *user*`] -N` *property-name* `[-e` *property-value*`] [-Y]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--template | -M]` *template-name*
Specifies which SSO adapter template the new property is assigned to.

`[--type | -t] [admin | user]`
Specifies the type of the new property.

`[--name | -N]` *property-name*
Specifies the name of the new property.

The following options are optional:

`[--value | -e]` *property-value*	Specifies the value of the property. This option is used only while creating an `admin` type property
`[--encrypt | -Y]`	Specifies whether the property value will be stored in encrypt form.

Example

Example for creating a property in a template:

```
./psadmin create-ssoadapter-property -u amadmin -f ps-password -M
test-template -N test-property -e test-value -t admin
```

psadmin get-ssoadapter-property

Description

Displays information about the specified SSO adapter property

Syntax

Long Format

`psadmin get-ssoadapter-property --adminuser` *uid* `--passwordfile` *password-filename* `[--template` *template-name* `| --config` *configuration-name*`] [-dn` *dn*`] -name` *property-name*

Short Format

`psadmin get-ssoadapter-property -u` *uid* `--f` *password-filename* `[-M` *template-name* `| -G` *configuration-name*`] [-d` *dn*`] -N` *property-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--template | -M] | [--config | -G]` *template-name* | *configuration-name*
Specifies the name of the template, if the property type is `admin`. Specifies the name of the configuration if the property type is `user`.

`[-name | -N]` *property-name*
Specifies which `admin` property to display.

The following option is optional:

`[--dn | -d]` *dn* — Specifies the distinguished name where the configuration is picked from LDAP. Used in conjunction with `[--config | -G]`. Default is `global`.

Example

Example for getting a property of configuration in a `dn`:

```
./psadmin get-ssoadapter-property -u amadmin -f ps-password -G
test-config -N test-property -d
uid=test-user,ou=People,o=DeveloperSample,dc=india,dc=sun,dc=com
```

psadmin set-ssoadapter-property

Description

Defines the value and type of the specified SSO adapter property

Syntax

Long Format

`psadmin set-ssoadapter-property --adminuser` *uid* `--passwordfile` *password-filename* `[--template` *template-name* `| --config` *configuration-name*`] [-dn` *dn*`] --name` *property-name* `[--type [-admin | -user]] [--value` *property-value*`] [--encrypt | --decrypt]`

Short Format

`psadmin set-ssoadapter-property -u` *uid* `-f` *password-filename* `[-M` *template-name* `| -G` *configuration-name*`] [-d` *dn*`] -N` *property-name* `[-t [-admin | -user]] [-e` *property-value*`] [-Y | -D]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--template | -M] | [--config | -G]` *template-name* | *configuration-name*
Specifies the name of the template, if the property type is `admin`. Specifies the name of the configuration if the property type is `user`.

`[--name | -N]` *property-name*
Specifies the property's name.

The following options are optional:

`[--type | -t [ admin | user]` Changes the property's type to new setting.

`[--value | -e]` *property-value* Specifies the property's value.

`[-dn | -d]` *dn* Specifies the distinguished name where the configuration is picked from LDAP. Used in conjunction with `[--config | -G]`. Default is `global`.

`[--encrypt | -Y] | [--decrypt | -D]`
Specifies whether to store the property value in encrypted or clear text form.

Example

Example for setting a property of a configuration in a dn:

```
./psadmin set-ssoadapter-property -u amadmin -f ps-password -G
test-config -N test-property -e test-value -d
uid=test-user,ou=People,o=DeveloperSample,dc=india,dc=sun,dc=com
```

psadmin delete-ssoadapter-property

Description

Deletes the specified SSO adapter property

Syntax

Long Format

`psadmin delete-ssoadapter-property --adminuser` *uid* `--passwordfile` *password-filename* `--template` *template-name* `--name` *property-name*

Short Format

`psadmin delete-ssoadapter-property -u` *uid* `-f` *password-filename* `-M` *template-name* `-N` *property-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--template | -M]` *template-name*
Specifies which SSO adapter template the property is assigned to.

`[--name | -N]` *property-name*
Specifies the name of the property to delete.

Example

Example for deleting property from a template:

```
./psadmin delete-ssoadapter-property -u amadmin -f ps-password -M
test-template -N test-property
```

psadmin create-ssoadapter-authless

Description

Creates a single user distinguished name (`dn`) for accessing the Portal Server without authentication and adds it to the support list

Syntax

Long Format

`psadmin create-ssoadapter-authless --adminuser` *uid* `--passwordfile` *password-filename* `--value` *authless-user-dn*

Short Format

`psadmin create-ssoadapter-authless -u` *uid* `-f` *password-filename* `-e` *authless-user-dn*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--value | -e]` *authless-user-dn*
Specifies the name of the single user DN for enabling access to the SSO adapter service without authentication.

Example

Example for creating an authless dn:

```
./psadmin create-ssoadapter-authless -u amadmin -f ps-password -e
test-dn
```

psadmin get-ssoadapter-authless

Description

Gets the specified single user distinguished name (`dn`) from the support list for accessing the Portal Server without authentication

Syntax

Long Format

`get-ssoadapter-authless --adminuser` *uid* `--passwordfile` *password-filename*

Short Format

`get-ssoadapter-authless -u` *uid* `-f` *password-filename*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[-passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

Example

Example for getting all the authless dn:

```
./psadmin get-ssoadapter-authless -u amadmin -f ps-password
```

psadmin delete-ssoadapter-authless

Description

Removes the specified single user distinguished name (`dn`) from the support list for accessing the Portal Server without authentication

Syntax

Long Format

`psadmin delete-ssoadapter-authless --adminuser` *uid* `--passwordfile` *password-filename* `--value` *authless-user-dn*

Short Format

`psadmin delete-ssoadapter-authless -u` *uid* `-f` *password-filename* `-e` *authless-user-dn*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (`dn`) or the user ID (`uid`).

`[-passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--value | -e]` *authless-user-dn*
Specifies which single user DN for enabling access to the SSO adapter service without authentication to remove.

Example

Example for deleting an authless dn:

```
./psadmin delete-ssoadapter-authless -u amadmin -f ps-password -e
test-dn
```

CHAPTER 27

Search Subcommands - Search Database Management

This chapter discusses the following search subcommands:

Search Database

analyze-search-database

Description

Analyzes a database for a search server.

Syntax

Long Format

```
psadmin analyze-search-database --searchserver search-server-ID
--adminuser uid --passwordfile password-filename --database database-name
[--debug] [--output output-filename] [--verbose] [--refresh]
```

Short Format

```
psadmin analyze-search-database -s search-server-ID -u uid -f
password-filename -r database-name [--debug] [--output output-filename] [-v]
[--refresh]
```

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--database | -r]` *database-name*
Specifies the database.

The following options are optional:

`[--refresh]`	Refreshes the analysis cache.
`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`create-search-database`

Description

Creates a search database.

Syntax

Long Format

`psadmin create-search-database --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [--verbose] [--attributes]` *'"key1=value1" "key2=value2" ...'*

Short Format

`psadmin create-search-database -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `-r` *database-name* `[--debug] [--output` *output-filename*`] [-v] [--attributes]` *'"key1=value1" "key2=value2" ...'*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--database | -r]` *database-name*
Specifies the database.

The following options are optional:

`[--attributes]` *'"key1=value1" "key2=value2" ...'*
Specifies a list of attributes for the database.

`[--debug]`
Specifies if debug mode is turned on.

`[--output]` *output-filename*
Specifies if the output mode is used.

`[--verbose | -v]`
Specifies if the verbose mode is used.

expire-search-database

Description

Expires a database for a search server.

Syntax

Long Format

`psadmin expire-search-database --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin expire-search-db -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `-r` *database-name* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--database | -r]` *database-name*
Specifies the database.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`list-search-databases`

Description

Lists all databases of a search server.

Syntax

Long Format

`psadmin list-search-databases --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin list-search-dbs -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

purge-search-database

Description

Purges the database for a search server.

Syntax

Long Format

`psadmin purge-search-database --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin purge-search-db -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `-r` *database-name* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--database | -r]` *database-name*
Specifies the database.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

reindex-search-database

Description

Reindexes a database for a search server.

Syntax

Long Format

`psadmin reindex-search-database --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `-database` *database-name* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin reindex-search-db -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `-r` *database-name* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--database | -r]` *database-name*
Specifies the database.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`get-search-database-attribute`

Description

Displays a search database attribute.

Syntax

Long Format

`psadmin get-search-database-attribute --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name* `--attribute` *key* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin get-search-db-attribute -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `-r` *database-name* `--attribute` *key* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--database | -r]` *database-name*
Specifies the database.

`[--attribute]` *key*
Specifies the attribute name.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`list-search-database-attributes`

Description

Lists a database's attributes.

Syntax

Long Format

`psadmin list-search-database-attributes --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin list-search-db-attributes -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `-r` *database-name* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--database | -r]` *database-name*
Specifies the database.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`set-search-database-attribute`

Description

Lists the attributes for a database.

Syntax

Long Format

`psadmin set-search-database-attribute --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name* `--attribute` *key* `--value` *value* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin set-search-db-attribute -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `-r` *database-name* `--attribute` *key* `--value` *value* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--database | -r]` *database-name*
Specifies the database.

`[--attribute]` *key*
Specifies the attribute name.

`[--value]` *value*
Specifies the attribute value.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

CHAPTER 28

Search Subcommands - Import Agents

This chapter discusses the following search subcommands:

Search Import

psadmin disable-search-importagents

Description

Disables the import agents in a search server.

Syntax

Long Format

`psadmin disable-search-importagents --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--importagents` *'"ID1" "ID2" ...'* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin disable-search-importagents -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--importagents` *'"ID1" "ID2" ...'* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--importagents]` *'"ID1" "ID2" ...'*
Specifies a list of import agent IDs.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`psadmin enable-search-importagents`

Description

Enables import agents in a search server.

Syntax

Long Format

`psadmin enable-search-importagents --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--importagents` *'"ID1" "ID2" ...'* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin enable-search-importagents -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--importagents` *'"ID1" "ID2" ...'* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--importagents]` *'"ID1" "ID2" ...'*
Specifies a list of import agent IDs.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

psadmin get-search-importagent-status

Description

Gets the import agent status in a search server.

Syntax

Long Format

`psadmin get-search-importagent-status --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--importagent` *import-agent-ID* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin get-search-importagent-status -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--importagent` *import-agent-ID* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--importagent]` *import-agent-ID*
Specifies the import agent ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

psadmin list-search-importagents

Description

Lists the import agents from a search server.

Syntax

Long Format

`psadmin list-search-importagents --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin list-search-importagents -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

psadmin run-search-importagents

Description

Runs an import agent from a search server.

Syntax

Long Format

`psadmin run-search-importagents --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin run-search-importagents -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

CHAPTER 29

Search Subcommands: Search Server Management

This chapter discusses the following topics:

Search Server Management

psadmin create-search-server

Description

Creates a new search server.

Syntax

Long Format

`psadmin create-search-server --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--webconfig` *properties-filename* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin create-search-server -s` *search-server-ID* `-u` *uid* `-f` *password-filemname* `-w` *properties-filename* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--webconfig | -w]` *properties-filename*
Specifies the web container properties file.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

psadmin delete-search-server

Description

Deletes a search server.

Syntax

Long Format

`psadmin delete-search-server --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--webconfig` *properties-filename* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin create-search-server -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `-w` *propertiesFile* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--webconfig | -w]` *properties-filename*
Specifies the web container properties file.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

psadmin list-search-servers

Description

Lists all search servers for a Portal Server domain.

Syntax

Long Format

`psadmin list-search-servers --adminuser` *uid* `--passwordfile` *password-filename* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin list-search-servers -u` *uid* `-f` *password-filename* `--debug --output` *output-filename* `-v`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

psadmin get-popular-search

Description

The `get-popular-search` subcommand displays the popular searches from a search server.

Syntax

Long Format

`psadmin get-popular-search --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `[--debug] [--output` *output-filename*`] [--verbose] [--cache] [--nobrowsing]`

Short Format

`psadmin get-popular-search -s` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `[--debug] [--output` *output-filename*`] [-v] [--cache] --nobrowsing]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.
`[--cache]`	Specifies that the `get-popular-search` subcommand uses the cache results.
`[--nobrowsing]`	Specifies that the `get-popular-search` subcommand excludes browsing results.

CHAPTER 30

Search Subcommands - Resource Description

This chapter discusses the following search subcommands:

Resource Description

create-search-resourcedescription

Description

Creates a resource description in a search server.

Syntax

Long Format

`psadmin create-search-resourcedescription --searchserver` *search-server-ID* `--file` *SOIF-filename* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin create-search-rd --s` *search-server-ID* `--file` *SOIF-filename* `-u` *uid* `-f` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--file]` *SOIF-filename*
Specifies the path and name of the SOIF file.

`[--database]` *database-name*
Specifies the database.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`delete-search-resourcedescription`

Description

Deletes resource descriptions from a search server.

Syntax

Long Format

`psadmin delete-search-resourcedescription --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--urls '"`*URL1*`" "`*URL2*`" "..." "`*URLN*`"' --database` *database-name* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin delete-search-rd -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--urls '"`*URL1*`" "`*URL2*`" "..." "`*URLN*`"' --database` *database-name* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--urls] '"`*URL1*`" "`*URL2*`" "...` `"`*URLN*`"'`
Specifies a list of URLs. To pass a list of strings, use the format '" string1" "string2" "..." "stringN"'.

`[--database]` *database-name*
Specifies the database.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`modify-search-resourcedescription`

Description

Modifies a resource description in a search server.

Syntax

Long Format

`psadmin modify-search-resourcedescription --searchserver` *search-server-ID* `--file` *SOIF-filename* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [--verbose]`

Short Format

`psadmin modify-search-rd -s` *search-server-ID* `--file` *SOIF-filename* `-u` *uid* `-f` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [-v]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--file]` *SOIF-filename*
Specifies the path and name of the SOIF file.

`[--database]` *database-name*
Specifies the database.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`list-search-resourcedescriptions`

Description

Lists the resource descriptions from a search server.

Syntax

Long Format

`psadmin list-search-resourcedescriptions --searchserver` *search-server-ID* `--searchtype` *search-type* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [--verbose] [--query` *query*`] [--viewattributes '"`*view1*`" "`*view2*`" "..." "`*viewN*`"'] [--startpage` *start-page*`] [--hitsperpage` *hits-per-page*`]`

Short Format

`psadmin list-search-rds -s` *search-server-ID* `--searchtype` *search-type* `-u` *uid* `-f` *password-filename* `--database` *database-name* `[--debug] [--output` *output-filename*`] [-v] [--query` *query*`] [--viewattributes '"`*view1*`" "`*view2*`" "..." "`*viewN*`"'] [--startpage` *start-page*`] [--hitsperpage` *hits-per-page*`]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--searchtype]` *search-type*
Specifies the type of search. The values are all, uncategorized, categorized, by_category, by_url, by_query.

`[--database]` *database-name*
Specifies the name of the database.

The following options are optional:

`[--debug]`
Specifies if debug mode is turned on.

`[--output]` *output-filename*
Specifies if the output mode is used.

`[--verbose | -v]`
Specifies if the verbose mode is used.

`[--query]` *query*
Specifies the search query. The values are `by_category`, `by_url` and`by_query`.

`[--veiwattributes] '"`*view1*`" "`*view2*`" "..." "`*viewN*`"'`
Displays the list of view attributes. To pass a list of strings, use the format `'"` *string1*” “*string2*” “...” “*stringN*”`'`.

`[--startpage]` *start-page*
Specifies pagination. The `startpage` option is used with `hitsperpage` to manage how the results are displayed. The `startpage` option determines the page for a grouping of results, and `hitsperpage` determines how many results are in one group. The `startpage` value can be `1...n`.

`[--hitsperpage]` *hits-per-page*
Specifies the number of hits per page that are displayed. It can be `1...n`. For example, to display the hits in groups of 5 on the page, specify the value for `hitsperpage` as 5.

CHAPTER 31

Search Subcommands-Robot Administration and Autoclassify

This chapter discusses the following search subcommands:

Search Robot and Autoclassify

get-robot-status

Description

Displays the search robot status.

Syntax

Long Format

`psadmin get-robot-status --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename*

Short Format

`psadmin get-robot-status -s` *search-server-ID* `-u` *uid* `-f` *password-filename*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`start-robot`

Description

Starts the search robot.

Syntax

Long Format

`psadmin start-robot --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename*

Short Format

`psadmin start-robot -s` *search-server-ID* `-u` *uid* `-f` *password-filename*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.

`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

`stop-robot`

Description

Stops the search robot.

Syntax

Long Format

`psadmin stop-robot --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename*

Short Format

`psadmin stop-robot -s` *search-server-ID* `-u` *uid* `-f` *password-filename*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

`create-robot-site`

Description

Creates a search robot crawling site.

Syntax

Long Format

`psadmin create-robot-site --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--sitename` *site-name* `[--isdomain] [--depth]`

Short Format

`psadmin create-robot-site -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--sitename` *site-name* `[--isdomain] [--depth]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--sitename]` *site-name*
Specifies the site name for the robot to crawl. For example `www.siroe.com`.

The following options are optional:

`[--isdomain]`	Specifies if the site is a domain.
`[--depth]`	Specifies the link depth to be crawled.
`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`delete-robot-site`

Description

Deletes a search robot crawling site.

Syntax

Long Format

`psadmin delete-robot-site --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--site` *site-ID*

Short Format

`psadmin delete-robot-site -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--site` *site-ID*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--site]` *site-ID*
Specifies the robot site ID. Use the `list-robot-sites` subcommand to find the site ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`list-robot-sites`

Description

Lists the search robot crawling sites.

Syntax

Long Format

`psadmin list-robot-sites --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `[--site]` *site-ID*

Short Format

`psadmin list-robot-sites -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `[--site]` *site-ID*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--site]` *site-ID*	Specifies the robot site ID to be displayed. If not specified, all robot sites are displayed.
`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`enable-robot-site`

Description

Enables a search robot crawling site.

Syntax

Long Format

`psadmin enable-robot-site --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--site` *site-ID*

Short Format

`psadmin enable-robot-site -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--site` *site-ID*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--site]` *site-ID*
Specifies the robot site ID. Use list-robot-site to find the site ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.

`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

`disable-robot-site`

Description

Enables a search robot crawling site.

Syntax

Long Format

`psadmin disable-robot-site --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--site` *site-ID*

Short Format

`psadmin disable-robot-site -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--site` *site-ID*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--site]` *site-ID*
Specifies the robot site ID. Use list-robot-site to find the site ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

`list-robot-converters`

Description

Lists all robot document converters for a search server.

Syntax

Long Format

`psadmin list-robot-converters --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename*

Short Format

`psadmin list-robot-converters -s` *search-server-ID* `-u` *uid* `-f` *password-filename*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`enable-robot-converter`

Description

Enables a search robot document converter.

Syntax

Long Format

`psadmin enable-robot-converter --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--converter` *converter-ID*

Short Format

`psadmin enable-robot-converter -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--converter` *converter-ID*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--converter]` *converter-ID*
Specifies the converter ID. Use the `list-robot-converters` subcommand to find the converter ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

`disable-robot-converter`

Description

Disables a search robot document converter.

Syntax

Long Format

`psadmin disable-robot-converter --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--converter` *converter-ID*

Short Format

`psadmin disable-robot-converter -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--converter` *converter-ID*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--converter]` *converter-ID*
Specifies the converter ID. Use the `list-robot-converter` subcommand to find the ID.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose	-v]`	Specifies if the verbose mode is used.

`run-robot-simulator`

Description

Runs the robot simulator in the search server.

Syntax

Long Format

```
psadmin run-robot-simulator --searchserver search-server-ID
--adminuser uid --passwordfile password-filename [--sitename site-name]
[--nodns] [--noredirect]
```

Short Format

```
psadmin run-robot-simulator -s search-server-ID -u uid -f password-filename
[--sitename site-name] [--nodns] [--noredirect]
```

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

The following options are optional:

`[--sitename]` *site-name*	Specifies the robot site for running robot simulator, for example, `www.siroe.com`. If this is not specified, the simulator runs against all robot sites that are not a domain .

`[--nodns]`	Specifies not to check for DNS aliases when running the robot simulator.
`[--noredirect]`	Specifies not to check for server redirect when running the robot simulator.
`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

`run-robot-siteprobe`

Description

Runs the robot simulator in a search server.

Syntax

Long Format

`psadmin run-robot-siteprobe --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--sitename` *site-name* `[--nodns]`

Short Format

`psadmin run-robot-siteprobe -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--sitename` *site-name* `[--nodns]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--sitename]` *siteName*
Specifies the robot site for running robot simulator. For example, `www.siroe.com`.

The following option is optional:

`[--nodns]`	Specifies not to check for DNS aliases when running the robot simulator.
`[--debug]`	Specifies if debug mode is turned on.

`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

`run-search-autoclassify`

Description

Runs autoclassify in a search server.

Syntax

Long Format

`psadmin run-search-autoclassify --searchserver` *search-server-ID* `--adminuser` *uid* `--passwordfile` *password-filename* `--database` *database-name*

Short Format

`psadmin run-search-autoclassify -s` *search-server-ID* `-u` *uid* `-f` *password-filename* `--database` *database-name*

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the Administrator's ID.

`[--passwordfile | -f]` *password-filename*
Specifies the name of the password file.

`[--searchserver | -s]` *search-server-ID*
Specifies the search server ID.

`[--database]` *database-name*
Specifies the database name against which to run autoclassify.

The following options are optional:

`[--debug]`	Specifies if debug mode is turned on.
`[--output]` *output-filename*	Specifies if the output mode is used.
`[--verbose | -v]`	Specifies if the verbose mode is used.

CHAPTER 32

Service Attribute Subcommands

This chapter discusses the following topics:

Service Attribute CLI

list-attributes

Description

The `list-attributes` command will list the configuration attributes for a specific Portal component.

Syntax

Long Format

`psadmin list-attributes --adminuser` *uid* `--passwordfile` *password-filename* `[--portal` *portal-name*`] --component` *component-name* `[--consumerregistration] [--configuredproducer]`

Short Format

`psadmin list-attributes -u` *uid* `-f` *password-filename* `[-p` *portal-name*`] --component` *component-name* `[--consumerregistration] [--configuredproducer]`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (dn) or the user identification name (uid).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--component | -m]` *component-name*
Specifies the component name. Valid names are: `producer`, `consumer`, `mobilemail`, `mobilecalendar`, `mobileaddressbook`, `gateway`, `netlet`, `proxylet`, `netfile`, and `subscriptions`.

The following options are optional:

`[--portal | -p]` *portal-name*
Specifies the portal name. The `--portal` option is required when a portal-centric component attribute is being accessed. The following components are portal-centric and you must provide the `--portal` option for `list`, `get`, and `set` operations while using the following components: desktop, producer, consumer, and subscriptions. For all other components, do not use the `-portal` option since they are common to all portals.

`[--consumerregistration]`
Specifies that the attributes of a producer's consumer registration are listed instead of the attributes of a producer. This is used only when the component is producer.

`[--configuredproducer]`
Specifies that the attributes of a consumer's configured producer are listed instead of the attributes of a consumer. This is used only when the component is consumer.

`[--gateway-profile]` *gateway-profile-name*
Specifies the gateway profile name. This is used only when the component is gateway.

Examples

```
psadmin list-attributes -u amadmin -f password_file --component
desktop -p myPortal
```

get-attribute

Description

The `get-attribute` command gets the attribute value for a specific component and attribute.

Syntax

Long Format

`psadmin get-attribute --adminuser` *uid* `--passwordfile` *password-filename* `[--portal` *portal-name*`] --component` *component-name* `[--producer` *producer_id*`] [--registrationhandle` *handle*`] --attribute-name` *attribute-name* `[--dn` *dn*`]`

Short Format

```
psadmin get-attribute -u uid -f password-filename [-p portal-name]
--component component-name [--producer producer-id]
[--registrationhandle handle] -a attribute-name [-d dn]
```

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (dn) or the user identification name (uid).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--component | -m]` *component-name*
Specifies the component name. Valid names are: `producer`, `consumer`, `mobilemail`, `mobilecalendar`, `mobileaddressbook`, `gateway`, `netlet`, `proxylet`, `netfile`, and `subscriptions`.

`[--attribute-name | -a]` *attribute-name*
Specifies the attribute name. For list of the attribute names for a component, please run the command `psadmin list-attributes --component` *component-name*.

The following options are optional:

`[--portal | -p]` *portal-name*
Specifies the portal name. The `--portal` option is required when a portal-centric component attribute is being accessed. The following components are portal-centric and you must provide the `--portal` option for `list`, `get`, and `set` operations while using the following components: desktop, producer, consumer, and subscriptions. For all other components, do not use the `-portal` option since they are common to all portals.

`[--producer]` *producer-ID*
Specifies the producer id. This is the ID or the name of a producer when the component is *producer* and it is the ID of a configured producer when the component is *consumer*. This option is valid only if the component is *producer* or *consumer*.

`[--registrationhandle]` *handle*
Specifies the registration handle of a producer's consumer registration. The attribute of a producer's consumer registration is returned (not the attribute of a producer itself). This option is valid only if the component is *producer*.

`[--dn | -d]`
Specifies the dn of the node. This must be an organization DN when the component is *consumer*.

`[--gateway-profile]` *gateway-profile-name*
Specifies the gateway profile name. This is used only when the component is *gateway*.

Example

```
psadmin get-attribute -u amadmin -f password-file --component
desktop -p myPortal --attribute-name AuthorizedAuthlessUserIds
[uid=devauthlessanonymous,ou=People,o=DeveloperSample,dc=red,dc=iplanet,dc=com
|devauthlessanonymous,
uid=authlessanonymous,ou=People,dc=red,dc=iplanet,dc=com |
authlessanonymous,
uid=commauthlessanonymous,ou=People,o=CommunitySample,dc=red,dc=iplanet,dc=com
| commauthlessanonymous,
uid=authlessenterprisesample,ou=People,o=EnterpriseSample,dc=red,dc=iplanet,dc=com
| authlessenterprisesample]

Successfully completed!
```

set-attribute

Description

The `set-attribute` command sets the attribute value for a specific component and attribute.

Syntax

Long Format

`set-attribute --adminuser` *uid* `--passwordfile` *password-filename* `[--portal` *portal-name*`] --component` *component-name* `[--producer` *producer-id*`] [--registrationhandle` *handle*`] --attribute-name` *attrribute-name* `[--dn` *dn*`] [--add` *add-values*`] [--remove` *remove-values*`] [--inherit] set_values`

Short Format

`set-attribute -u` *uid* `-f` *password-filename* `[-p` *portalName*`] --component` *component-name* `[--producer` *producer-id*`] [--registrationhandle` *handle*`] -a` *attribute-name* `[-d` *dn*`] [-A` *add-values*`] [-E` *remove-values*`] [--inherit] set_values`

Options

The following options are required:

`[--adminuser | -u]` *uid*
Specifies the administrator's distinguished name (dn) or the user identification name (uid).

`[--passwordfile | -f]` *password-filename*
Specifies the administrator's password in the password file.

`[--component | -m]` *component-name*
Specifies the component name. Valid names are: `producer`, `consumer`, `mobilemail`, `mobilecalendar`, `mobileaddressbook`, `gateway`, `netlet`, `proxylet`, `netfile`, and `subscriptions`.

`[--attribute-name | -a]` *attribute-name*
Specifies the attribute name. For list of the attribute names for a component, please run the command `psadmin list-attributes --component` *component-name*.

The following is an operand:

`set_values` A list of values to be set into the attribute. This can be specified as `'"v1" "v2"'`. If the `set_values` operand is used, the `--inherit`, `--add`, and `--remove` options are not allowed. The `set_values` operand is used at the end of the command.

The following options are optional:

`[--portal | -p]` *portal-name*
Specifies the portal name. The `--portal` option is required when a portal-centric component attribute is being accessed. The following components are portal-centric and you must provide the `--portal` option for `list`, `get`, and `set` operations while using the following components: desktop, producer, consumer, and subscriptions. For all other components, do not use the `-portal` option since they are common to all portals.

`[--producer]` *producer-ID*
Specifies the producer id. This is the ID or the name of a producer when the component is *producer* and it is the ID of a configured producer when the component is *consumer*. This option is valid only if the component is *producer* or *consumer*.

`[--registrationhandle]` *handle*
Specifies the registration handle of a producer's consumer registration. The attribute of a producer's consumer registration is modified (not the attribute of a producer itself). This option is valid only if the component is *producer*.

`[--dn | -d]`
Specifies the dn of the node. This must be an organization DN when the component is *consumer*.

`[--inherit]`
Specifies that the value should be inherited from parent. This option can only be used for users. This option does not use any values. If the `--inherit` option is used, the `--add` option, the `--remove` option, and the `set_values` operand are not allowed. The value is inherited from the parent. The `--inherit` option is used only for user attribute.

`[--gateway-profile]` *gateway-profile-name*
Specifies the gateway profile name. This is used only when the component is *gateway*.

`[--add | -A]` *add-values*
Specifies a list of values to add like `'"value1" "value2"'`. If the add option is specified, the `--inherit` option and the `set_values` operand are not allowed. The remove option is allowed, but if it is not specified, the set is added to existing values. If the removed option is specified, the result is that some values will be added and some values will be removed. The options can be done in any order, but the operations are such that remove done after add.

`[--remove | -E]` *remove-values*
Specifies a list of values to remove like `'"value1" "value2"'`. If the `--remove` option is specified, the `--inherit` option and the `set_values` operand are not allowed. The `--add` option is allowed, but if it is not specified, the set is removed from existing values. If the `--add` option is specified, some values are added and some values are removed. The options can be done in any order, but the operations are such that remove done after add.

Examples

Example for adding attributes:

```
psadmin set-attribute -u amadmin -f password-file --component
desktop -p myPortal --attribute-name AuthorizedAuthlessUserIds
--add '"val1" "val2" "val3"'

3 values, val1, val2 and val3 will be added to the existing list
```

Example for removing attributes:

```
psadmin set-attribute -u amadmin -f password-file --component
desktop -p myPortal --attribute-name AuthorizedAuthlessUserIds
--remove '"val1" "val2" "val3"'

The 3 values, val1, val2 and val3 will be removed from the
existing list.
```

Example for using the `set_values` operand:

```
psadmin set-attribute -u amadmin -f password-file --component
desktop -p myPortal --attribute-name AuthorizedAuthlessUserIds
'"a1" "a2" "a3"'

The existing list will be replaced by 3 values, a1, a2 and a3.
```

CHAPTER 33

SRA Subcommands

This sections describes the following commands:

create-sra-instance

Description

Creates a Gateway, Netlet Proxy or Rewriter Proxy instance.

Syntax

Long Format

`create-sra-instance --adminuser` *uid* `--passwordfile` *password-filename* `--instance-type` *instance-type* `--instance-hostname` *instance-hostname* `--sraconfig` *sra-config-filename*

Short Format

`create-sra-instance -u` *uid* `-f` *password-filename* `--instance-type` *instance-type* `--instance-hostname` *instance-hostname* `--sraconfig` *sra-config-filename*

Options

The following options are required:

[`---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.

`--instance-type` *instance-type*
Specifies the type of the SRA instance. Enter: gateway, nlproxy, or rwproxy.

The following options are not required:

`--instance-hostname` *instance-hostname*
Specifies the host name where the instance is located.

`--sraconfig` *sra-config-filename*
Specifies the file name of the configuration file.

`list-sra-instances`

Description

Lists all instances of Gateway, Netlet Proxy, and Rewriter Proxy.

Syntax

Long Format

`psadmin list-sra-instances --adminuser` *uid* `--passwordfile` *password-filename* `--instance-type` *instance-type*

Short Format

`psadmin list-sra-instances -u` *uid* `-f` *password-filename* `--instance-type` *instance-type*

Options

The following options are required:

[`--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.

`--instance-type` *instance-type*
Specifies the type of Secure Remote Access instance. Enter: gateway, nlproxy, or rwproxy.

`delete-sra-instance`

Description

Removes a Gateway, Netlet Proxy, or Rewriter Proxy instance.

Syntax

Long Format

`psadmin delete-sra-instance --adminuser` *uid* `--passwordfile` *password-filename* `--instance-name` *sra-instance-name* `--instance-type --instance` *instance-type* `--instance-hostname` *instance-hostname*

Short Format

`psadmin delete-sra-instance -u` *uid* `-f` *password-filename* `--instance-name` *sra-instance-name* `--instance-type` *instance-type* `--instance-hostname` *instance-hostname*

Options

The following options are required:

[`--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.

`--instance-type` *instance-type*
Specifies the type of the Secure Remote Access instance. Enter: gateway, nlproxy, or rwproxy.

`--instance-name` *sra-instance-name*
Lists the instance names for Gateway, Netlet Proxy and Rewriter Proxy.

The following options are not required:

`--instance-hostname` *instance-hostname*
Specifies the hostname where the instance is located.

start-sra-instance

Description

Starts a Gateway, Netlet Proxy, or Rewriter Proxy instance.

Syntax

Long Format

`psadmin start-sra-instance --adminuser` *uid* `--passwordfile` *password-filename* `--instance-name` *sra-instance-name* `--instance-type` *instance-type* `--instance-hostname` *instance-hostname*

Short Format

`psadmin start-sra-instance -u` *uid* `-f` *password-filename* `--instance-name` *sra-instance-name* `--instance-type` *instance-type* `--instance-hostname` *instance-hostname*

Options

The following options are required:

[`--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.

`--instance-type` *instance-type*
Specifies the type of the SRA instance. Enter: gateway, nlproxy, or rwproxy.

`--instance-name` *sra-sra-instance-name*
Lists the instance names for Gateway, Netlet Proxy and Rewriter Proxy.

The following options are not required:

`--instance-hostname`*instance-hostname*	Specifies the hostname where the instance is located.

stop-sra-instance

Description

Stops a Gateway, Netlet Proxy, or Rewriter Proxy instance.

Syntax

Long Format

`stop-sra-instance --adminuser` *uid* `--passwordfile` *password-filename* `--instance-name` *sra-instance-name* `--instance-type` *instance-type* `--instance-hostname` *instance-hostname*

Short Format

`stop-sra-instance -u` *uid* `-f` *password-filename* `--instance-name` *sra-instance-name* `--instance-type` *instance-type* `--instance-hostname` *instance-hostname*

Options

The following options are required:

[`--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.
`--instance-type` *instance-type*	Specifies the type of Secure Remote Access instance. Enter: gateway, nlproxy, or rwproxy.
`--instance-name` *sra-instance-name*	Lists the instance names for Gateway, Netlet Proxy and Rewriter Proxy.

The following options are not required:

`--instance-hostname` *instance-hostname*
Specifies the hostname where the instance is located.

change-loguser-password

Description

Changes the Gateway logging user password.

Syntax

Long Format

`psadmin change-loguser-password --adminuser` *uid* `--passwordfile` *password-filename* `--newpasswordfile` *new-password-filename* `--instance-name` *sra-instance-name* `--instance-hostname` *instance-hostname*

Short Format

`psadmin change-loguser-password -u` *uid* `-f` *password-filename* `--newpasswordfile` *new-password-filename* `--instance-name` *sra-instance-name* `--instance-hostname` *instance-hostname*

Options

The following options are required:

[`--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.

`--newpasswordfile` *new-password-filename*
Specifies the new name for the password file.

The following options are not required:

`--instance-hostname` *instance-hostname*
Specifies the hostname where the instance is located.

sra-watchdog

Description

Configures watchdog for the Gateway, Netlet, Netfile, Proxylet, and Accesslist.

Syntax

Long Format

`psadmin sra-watchdog --adminuser` *uid* `--passwordfile` *password-filename* `--instance-type` *instance-type* `[--instance-hostname` *instance-hostname* `]on|off`

Short Format

`psadmin sra-watchdog -u` *uid* `-f` *password-filename*`--instance-type` *instance-type* `[instance-hostname` *instance-hostname* `]on|off`

Options

The following options are required:

`[--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.
`--instance-type` *instance-type*	Specifies the type of the SRA instance. Enter: gateway, nlproxy, or rwproxy.

The following options are not required:

`--instance-hostname` *instance-hostname*
Specifies the hostname where the instance is located.

The following operand is available:

`on|off` Specifies to switch watchdog on or off for the particular instance.

get-attribute

Description

Gets an attribute for a Portal Server Secure Remote Access component: Gateway, Netlet, Netfile, Proxylet, or Accesslist. If the component is `gateway`, use the `--gateway-profile` *gateway-profile-filename* option to specify the name to obtain the value of the attribute.

list-attributes

Description

Lists all attributes for a Portal Server Secure Remote Access component: Gateway, Netlet, Netfile, Proxylet, or Accesslist. If the component is `gateway`, use the `--gateway-profile` *gateway-profile-filename* option to specify the name to obtain the value of the attribute.

set-attribute

Description

Sets an attribute value for a Portal Server Secure Remote Access component: Gateway, Netlet, Netfile, Proxylet, or Accesslist. If the component is `gateway`, use the `--gateway-profile` *gateway-profile-filename* option to specify the name to obtain the value of the attribute.

CHAPTER 34

Rewriter Subcommands

This section describes the following commands:

get-rewriter-ruleset

Description

Returns the ruleset xml.

Syntax

Long Format

`psadmin get-rewriter-ruleset --adminuser` *uid* `--passwordfile` *password-filename* `--rule` *ruleset-ID*

Short Format

`psadmin get-rewriter-ruleset -u` *uid* `-f` *password-filename* `-R` *ruleset-ID*

Options

The following options are required:

[`--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.

[`--rule | -R`] *ruleset-ID*
Specifies the ruleset ID.

list-rewriter-rulesets

Description

Lists the rewriter rulesets.

Syntax

Long Format

`psadmin list-rewriter-rulesets --adminuser` *uid* `--passwordfile` *password-filename*

Short Format

`psadmin list-rewriter-rulesets -u` *uid* `-f` *password-filename*

Options

The following options are required:

[`--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.

create-rewriter-ruleset

Description

Stores the rewriter ruleset.

Syntax

Long Format

`psadmin create-rewriter-ruleset --adminuser` *uid* `--passwordfile` *password-filename* `--file` *ruleset-filename*

Short Format

`psadmin create-rewriter-ruleset -u` *uid* `-f` *password-filename* `-F` *ruleset-filename*

Options

The following options are required:

[`--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.

[--file | -F] ruleset-filename
Specifies the ruleset xml filename.

delete-rewriter-ruleset

Description

Removes the rewriter ruleset.

Syntax

Long Format

psadmin delete-rewriter-ruleset --adminuser *uid* --passwordfile *password-filename*

Short Format

psadmin delete-rewriter-ruleset -u *uid*

Options

The following options are required:

[--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.

[-rule | -R] *ruleset-ID*
Specifies the ruleset ID.

set-rewriter-ruleset

Description

Sets the rewriter ruleset.

Syntax

Long Format

psadmin set-rewriter-ruleset --adminuser *uid* --passwordfile *password-filename* --file *ruleset-filename*

Short Format

psadmin set-rewriter-ruleset -u *uid* -f *password-filename* -F *ruleset-filename*

Options

The following options are required:

[`--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.

[`--file | F`] *ruleset-filename*
Specifies the ruleset xml filename.

CHAPTER 35

Mobile Access Subcommands

This supplement contains the following Mobile Access subcommands:

psadmin get-attribute

Description

The `get-attribute` subcommand will return all Mobile Access specified component attributes

Syntax

Long-Named Format

`psadmin get-attribute --adminuser` *uid* `--passwordfile` *passwordfile* `--component` *componentname* `[--dn` *dn* `| -global | --org] --attribute`*attributename*

Short-Named Format

`psadmin get-attribute -u` *uid* `-f` *passwordFile* `--component` *componentname* `[--dn` *dn* `| --g | --org] -a` *attributename*

Options

The following options are required:

`[--adminuser | -u]`	Specifies the administrator's name or the user identification name.
`[--passwordfile | -f]`	Specifies the administrator's password in the password file.
`--component`	Specifies the name of the component

[--dn | -d]	Specifies the name of the object. For example, user, organization, role to which this command can be applied.
[--gobal | -g]	Specifies the command can applied for global settings.
--org	Specifies the command can applied for organizational settings.
[--attribute | -a]	Specifies the attribute.

psadmin set-attribute

Description

The `set-attribute` subcommand sets the specified Mobile Access component attribute.

Syntax

Long-Named Format

`psadmin set-ma-addressbook-attribute--adminuser` *uid* `--passwordfile` *passwordfile* `--component` *componentname* `[--dn` *dn*`| -global | --org] --attribute` *attributename* `[--add <`*valuelist*`>] [--remove <`*valuelist*`>] <`*valuelist*`>`

Short-Named Format

`psadmin get-ma-addressbook-attribute -u` *uid* `-f` *passwordFile* `--component` *componentname* `[--dn` *dn* `| --g | --org] -a` *attributename* `[-A <`*valuelist*`>] [-E <`*valuelist*`>] <`*valuelist*`>`

Options

The following options are required:

[--adminuser | -u]	Specifies the administrator's distinguished name or the user identification name.
[--passwordfile | -f]	Specifies the administrator's password in the password file.
--component	Specifies the name of the component
[--dn | -d]	Specifies the name of the object. For example, user, organization, role to which this command can be applied.
[--gobal | -g]	Specifies the command can applied for global settings.

`--org`	Specifies the command can applied for organizational settings.
`[--attribute | -a]`	Specifies the attribute for the value to be retrived.
`[--add|-A]`	Specifies the list of values that can be added to the attribute. This option is valid for attributes that hold list of values.
`[--remove|-E]`	Specifies the list of values that can be removed from the attribute. This option is valid for attributes that hold list of values.
<*valuelist*>	Specifies the value or the list of values that can be set to a specified attribute.

psadmin list-attributes

Description

The `psadmin list-attributes` subcommand will list all available attributes and attribute metadata for the specified component.

Syntax

Long-Named Format

`psadmin list-attributes--adminuser` *uid* `--passwordfile` *passwordfile* `--component`*componentname*

Short-Named Format

`psadmin list-attributes-u` *uid* `-f` *passwordFile* `-component` *componentname*

Options

The following options are required:

`[--adminuser | -u]` *uid*	Specifies the administrator's name or the user identification name.
`[--passwordfile | -f]` *passwordFile*	Specifies the administrator's password in the password file.
`--component`	Specifies the name of the component which lists the attributes.

CHAPTER 36

Inter Portlet Communication

This supplement contains the following sections:

Introduction

The Portal Server software includes an application programming interface (API) which is an extension to JSR-168 Portlets, in the `com.sun.portal.portlet` Java package. Using this API, JSR-168 portlets will be able communicate with each other even if they are in different web applications. It is assumed that all these portlets will be on the same instance of a Portal Server and running inside the Portal Server Portlet container.

Inter Portlet Communication API

This section contains the following:

Overview

This API uses event generation and notification to convey the information/data among portlets. The event notification will be for the portlets which have registered themselves for listening to that particular event. Portlets interested in receiving an event will implement a single interface `PortletEventListener`.

The `EventRequest` interface can then be used to obtain the event name and event payload data. Event payload data can be obtained either by getting the event stream and reading from it or by calling `getEventData()` method which returns an Object and then casting it appropriately.

The `EventResponse` interface can then be used to set the render parameters so that the information can be passed on to the render method after processing the event received in `handleEvent()` method. The portlets can generate events only from within the `handleEvent()` or `processAction()` methods. Event can be generated by instantiating `PortletEventBroker` and calling `createEvent()` method on it.

Event Generation and Subscription

Event generation starts with an event generated in `processAction()` method of a portlet. Further events can also be generated in `handleEvent()` method in portlet class while handling the event received. The `handleEvent()` method will be called if and only if an event is fired and the portlet has subscribed to listen to that event. The events can also be generated in response to other events in `handleEvent()` method.

To create portlets which are interested in listening to certain events and taking some action in response to these events, the portlet must implement `PortletEventListener` interface. This interface has only `handleEvent()` method. The portlet gets the data from *EventRequest* and can take appropriate action. The developer can set any information required for the correct rendering of the portlets on *EventResponse* as render parameters.

All the portlets which are interested in listening or generating an event must declare it in `sun-portlet.xml` file. If a portlet requests an event which has not been declared in the `sun-portlet.xml` file, an exception `NotRegisteredException` will be thrown. Wild cards can not be used for declaring the events that will be generated. However, portlets interested in consuming all the events can use wildcard character (`*`) only inside `<consumes—event></consumes—event>` block.

Event Handling Life Cycle

The event cycle begins with the response to user interaction from inside `processAction()` method referred to as Generation 1 events. These events are placed in the event queue by the Portlet Container and dispatched in the order they are created. The dispatching of the events continues till all the events in the event queue are dispatched to appropriate portlets. Dispatching of the events amount to calling the `handleEvent()` methods of the appropriate portlets. Portlets can generate events in the `handleEvent()` method which are referred to as Generation next events. If a portlet has subscribed to events which are generated in different generations, it will receive the events in proper order; that is, the `handleEvent()` method will be called with Generation i event first and upon completion of that method, `handleEvent()` method will be called with event from Generation i+1.

Scope of Event Processing

The events are sent to all the portlets obtained by recursively calling `getSelectedChannels()` method on the top level container (if any) as referenced by the request.

Infinite Event Cycle Detection

Events are generated in response to the user interaction (Generation 1) or in response to other events (Generation next). This could lead to more and more generations being created. To control the number of generations, the *maxEventGenerations* parameter in the `desktopConfig.properties` file can be configured for maximum number of generations of events per request. When the event creation exceeds the specified maximum number, a failure event, `eventHandlingFailed` will be sent to all the participating portlets.

Deterministic Behavior

If a portlet generates events X and Y in that order, then events X and Y will be delivered to the portlets in that same order. If portlet A and B are interested in Event X , either A or B can get event X first. If portlets A and B are interested in Event X, and upon receipt of that event generate events Y and Z respectively, and Portlet C is interested in Event Y and Z, then portlet C can receive events Y and Z in any order.

Failure and Exception Handling

In case of failure, the `handleEvent()` method may throw `PortletException`. The container catches that exception and will stop sending the events from the event queue. The container will then send another event called `eventHandlingFailed` to all the portlets participating in that particular interaction. The container will not take any action if the `PortletException` is thrown while processing `eventHandlingFailed` event. Portlets can not generate and send any events while handling the event `eventHandlingFailed`.

Samples

Two developer samples are included. The first sample demonstrates first-order eventing to communicate information from source portlet to target portlet. In this case, the source portlet fires one event and the target portlet gets that event and updates its rendering.

In the second sample, a portlet generates an event which is delivered to the target portlet. In response to this event, the target portlet generates another event, which is consumed by a third portlet which updates its rendering in response to the event received.

CHAPTER 37

Tag Library Reference

This chapter contains the following sections:

Tag library for Desktop Channel and Container Management Tasks

`obtainChannelAdmin`	All the channel administration tags must be nested inside this tag and will operate on the base distinguished node that is passed into this tag. For example: `<dtadmin:obtainChannelAdmin baseDN="dc=example,dc=siroe,dc=com"> ... <dtadmin:obtainChannelAdmin>` The `baseDN` is a required attribute and must be a valid node in the directory server. For example, `dc=red,dc=iplanet,dc=com`. Returns variables available inside the nesting with extra information for this tag: `CHANNEL_NAME_SEPARATOR`, `STRING_DP`, `INTEGER_DP`, `BOOLEAN_DP`, `COLLECTION_DP`, `UNKNOWN_DP`.
`getBaseDNs`	Gets the list of distinguished nodes the currently authenticated delegated administrator can administer. For example: `<dtadmin:getBaseDNs id="nodes"/>` .

	The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. Returns `java.util.Set`.
`getDNDisplayName`	Gets the display name for the distinguished name that is passed in. For example, `<dtadmin:getDNDisplayName id="name" dn="dc=red,dc=iplanet,dc=com"/>.` The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `dn` attribute is required and takes the distinguished name for which the display name is requested. Returns `java.lang.String`.
`getAssignableChannels`	Returns the set of channels or containers that are assignable to the available and selected list of a container. This tag returns the list of channels which includes the container's children and the container's parent's children, parent's parent's children, and traverses recursively until it reaches the display profile root. Mergers are taken into account when the channels are traversed. For example, `<dtadmin:getAssignableChannelsid="AssignableChanne container="containerName"/>` The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `container` attribute is required and takes the name of container for which assignable channels are requested. Returns `java.util.Set`.

`getExistingChannels`	Returns the set of existing channels. The channels in this list can be modifiable or deleted by the user who can administer the base distinguished node that this tag is operating on. This tag returns the list of channels which includes the channels defined at the base distinguished node, channels defined inside the containers that are defined at the base distinguished node. If *all* is set to true, all the channels from the merged list is returned. This tag returns the set of channels that match the search string if *regExp* is provided. For example, `<dtadmin:getExistingChannels id="FilteredChannels" regExp="*" all="false"/>` The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `all` attribute is required and if true, takes mergers into account. The `regExp` attribute takes a search string (if filtering resulting channels based on a regular expression) specifically of the form `foo*`, `*foo`, `foo*bar`, `foo*bar*`. Returns `java.util.Set`.
`getExistingContainers`	Returns the set of existing containers. The containers in this list can be modifiable or deleted by the user who can administer the base distinguished node that this tag is operating on. This tag returns the list of containers which includes the containers defined at the base distinguished node, containers defined inside the containers that are defined at the base distinguished node. If *all* is set to true, all the containers from the merged list is returned. This tag returns the set of containers that match the search string if `regExp` is provided. For example, `<dtadmin:getExistingContainers id="ExistingContainers" all="true"/>`

	The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `all` attribute is required and if true, takes mergers into account. The `regExp` attribute takes a search string (if filtering resulting channels based on a regular expression) specifically of the form `foo*`, `*foo`, `foo*bar`, `foo*bar*`. Returns `java.util.Set`.
`getExistingProviders`	Returns the set of existing providers. The providers in this list can be used to create channels by the user that can administer the base distinguished node that this tag is operating on. Always takes mergers into account. For example, `<dtadmin:getExistingProviders id="ExistingProviders"/>` The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. Returns `java.util.Set`.
`getExistingContainerProviders`	Returns the set of existing container providers. The container providers in this list can be used to create containers by the user who can administer the base distinguished node that this tag is operating on. Always takes mergers into account. For example, `<dtadmin:getExistingContainerProviders id="ExistingContainerProviders"/>` . The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. Returns `java.util.Set`.

`createChannel`	Creates a new channel. A new channel is created based on the named provider. To create a nested channel, supply a hierarchical channel name. For example, to create channel A inside of container X, based on provider P: `<dtadmin:createChannel channelName="A/X" providerName="P"/>`. Use the `CHANNEL_NAME_SEPARATOR` variable for constructing the hierarchical channel names. The `channelName` attribute is required and takes the name of the channel. The `providerName` attribute is required and must contain the name of the base provider.
`createContainer`	Creates a new container. A new container is created based on the named container provider. To create a nested container, supply a hierarchical channel name. For example, to create container A inside of container X, based on provider P: `<dtadmin:createContainerl channelName="A/X" providerName="P"/>` The `channelName` attribute is required and takes the name of the channel. The `providerName` attribute is required and must contain the name of the base provider.
`deleteChannel`	Deletes a channel or container. To delete a nested channel or container, supply parent name. For example, to delete channel A inside of container X, at base distinguished node `dc=iplanet,dc=com,<dtadmin:deleteChannel node="dc=iplanet,dc=com" channelName="A" parentContainer="X"/>`. The `channelName` attribute is required and takes the name of the channel. The `parentContainer` attribute is required and takes the parent container name.

`getAvailableChannels`	Gets the list of available channels in a container. For example, `<dtadmin:getAvailableChannels id="AvailableChannels" container="containerName"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `container` attribute is required and takes the name of the container for which assignable channels are requested. Returns `java.util.List`.
`getSelectedChannels`	Gets the list of selected channels in a container. For example, `<dtadmin:getSelectedChannels id="SelectedChannels" container="containerName"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `container` attribute is required and takes the name of the container for which assignable channels are requested. Returns `java.util.List`.
`setAvailableChannels`	Sets the list of available channels in a container. For example, `<dtadmin:setAvailableChannels available="$available" container="JDCFrontPageTabPanel"/>`. The `container` attribute is required and takes the name of the container for which assignable channels are requested. The `available` attribute is required and takes the new available list.

`setSelectedChannels`	Sets the list of selected channels in a container. For example, `<dtadmin:setSelectedChannels selected="$selected" container="JDCFrontPageTabPanel"/>` . The `container` attribute is required and takes the name of the container for which assignable channels are requested. The `selected` attribute is required and takes the new selected list.
`getClassName`	Gets the fully classified class name for the provider class that this channel is based on. For example, `<dtadmin:getClassName id="classname" channel="channelName"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `channel` attribute is required and takes the channel name for which the class is requested. Returns `java.lang.String`.
`getPropertyNames`	Gets the list of properties in a channel or container. This tag returns the list of basic property names at the given channel name that is passed in if *advanced* attribute is false or not set. If *advanced* attribute is set to true, it returns the advanced property names. If `regExp` is specified, returns the property names that match the search string. For example, `<dtadmin:getPropertyNames channel="SampleURLScraper"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `channel` attribute is required and takes the channel name for which the class is requested. The `pflist` attribute is

	optional and takes the list of `PropertyFilter` objects. The `regExp` attribute takes a search string (if filtering the resulting set based on a regular expression.) The `advanced` attribute is optional. Returns `java.util.Set`.
`getPropertyType`	Returns the property type for the given property name for a given channel. For example, `<dtadmin:getPropertyType channel="JDCTab/JDCChannel" property="title" id="titleType"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `channel` attribute is required and takes the channel name for which the class is requested. The `property` attribute is required and takes the property name. Returns `java.lang.short`. The returned value can be compared to one of the following variables available as extra info to determine the type: `STRING_DP`, `INTEGER_DP`, `BOOLEAN_DP`, `COLLECTION_DP`, `UNKNOWN_DP`
`getStringProperty`	Gets the string property value given a channel or container name and the property key. The `PropertyFilter` list is optional; if given, gets the string property based on the property filter's condition and value. If channel name is not set, then the string property value from the global properties is returned. For example, `dtadmin:getStringProperty channel="MyFrontPageTabPanelContainer" key="title"/`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `channel` attribute is required and takes

	the channel name for which the class is requested. The `key` attribute is required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects. Returns `java.lang.String`.
`getIntegerProperty`	Gets the integer property value given a channel or container name and the property key. The `PropertyFilter` list is optional; if given, gets the integer property based on the property filter's condition and value. If channel name is not set, then the integer property value from the global properties is returned. For example, `<dtadmin:getIntegerProperty channel="MyFrontPageTabPanelContainer" key="timeout"/>.` The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `channel` attribute is required and takes the channel name for which the class is requested. The `key` attribute is required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects. Returns `int`.
`getBooleanProperty`	Gets the boolean property value given a channel or container name and the property key. `PropertyFilter` list is optional; if given, gets the boolean property based on the property filter's condition and value. If channel name is not set, then the boolean property value from the global properties is returned. For example, `<dtadmin:getBooleanProperty channel="MyFrontPageTabPanelContainer" key="parallelChannelsInit"/>.` The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `channel` attribute is required and takes

	the channel name for which the class is requested for. The `key` attribute is required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects. Returns `boolean`.
`getListProperty`	Gets the list property value given a channel or container name and the property key. `PropertyFilter` list is optional; if given, gets the collection property based on the property filter's condition and value. If channel name is not set, then the collection property value from the global properties is returned. The value returned is a List. For example, `<dtadmin:getListProperty channel="MyFrontPageTabPanelContainer" key="categories"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `channel` attribute is required and takes the channel name for which the class is requested for. The `key` attribute is required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects. Returns `java.util.List`.
`getMapProperty`	Gets the map property value given a channel or container name and the property key. `PropertyFilter` list is optional; if given, gets the collection property based on the property filter's condition and value. If channel name is not set, then the collection property value from the global properties is returned. The value returned is a Map. For example, `<dtadmin:getMapProperty channel="MyFrontPageTabPanelContainer" key="channelsRow"/>`.

	The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `channel` attribute is required and takes the channel name for which the class is requested for. The `key` attribute is required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects. Returns `java.util.Map`.
`setStringProperty`	Sets the string property value given a channel or container name and the property key and the value. `PropertyFilter` list is optional; if given, sets the string property based on the property filter's condition and value. If channel name is not set, then the string property value at the global properties is set. For example, `<dtadmin:setStringProperty channel="MyFrontPageTabPanelContainer" key="title" value="New Front Page title-1"/>`. The `channel` attribute is required and takes the channel name for which the class is requested for. The `key` and `value` attributes are required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects.
`setIntegerProperty`	Sets the integer property value given a channel or container name and the property key and the value. `PropertyFilter` list is optional; if given, sets the integer property based on the property filter's condition and value. If channel name is not set, then the integer property value at the global properties is set. For example, `<dtadmin:setIntegerProperty channel="MyFrontPageTabPanelContainer" key="timeout" value="80"/>`.

	The `channel` attribute is required and takes the channel name for which the class is requested for. The `key` attribute and `value` attributes are required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects.
`setBooleanProperty`	Sets the boolean property value given a channel/container name and the property key and the value. `PropertyFilter` list is optional. If given sets the boolean property based on the property filter's condition and value. If channel name is not set, then the boolean property value at the global properties is set. For example, `<dtadmin:getBooleanProperty channel="MyFrontPageTabPanelContainer" key="parallelChannelsInit"/>`. The `channel` attribute is required and takes the channel name for which the class is requested for. The `key` and `value` attributes are required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects.
`setListProperty`	Sets the list property value given a channel or container name, and the property key and the value. `PropertyFilter` list is optional; if given, sets the list property based on the property filter's condition and value. If channel name is not set, then the list property value at the global properties is set. For example, `<dtadmin:setListProperty channel="MyFrontPageTabPanelContainer" key="categories" value="$list"/>`. The `channel` attribute is required and takes the channel name for which the class is requested for. The `key` attribute is required. The `value` attribute is required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects.
`setMapProperty`	Sets the map property value given a channel/container name and the property key and the value. `PropertyFilter` list

is optional. If given sets the map property based on the property filter's condition and value. If channel name is not set, then the map property value at the global properties is set. For example, `<dtadmin:setMapProperty channel="MyFrontPageTabPanelContainer" key="channelsRow" value="$map/>`.

The `channel` attribute is required and takes the channel name for which the class is requested for. The `key` and `value` attributes are required. The `pflist` attribute is optional and takes the list of `PropertyFilter` objects.

Tag library for Portlet Management Tasks

`obtainPortletAdmin`

All the portlet administration tags should be called from within this nested tag and will be operating on the base distinguished node specified here. For example, `<dtportletadmin:obtainPortletAdmin baseDN="$baseDN">...</dtportletadmin:obtain` .

The `baseDN` attribute is required and must be a valid node in Directory Server. For example, `dc=red,dc=iplanet,dc=com`.

`createPortletChannel`

Allows creation of a portlet channel based on a portlet. For example, `<dtportletadmin:createPortletChannel channelName="myPortletChannel" portletName="portletsamples.JSPPortlet"/>` .

The `channelName` attribute is required and must contain the name of the channel. The `portletName` attribute is required and must contains the name of the base portlet.

`getExistingPortlets`	Gets the list of existing portlets. For example, `<dtportletadmin:getExistingPortlets id="ExistingPortlets"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. Returns `java.util.Set`.
`getPortletPreferenceNames`	Gets the list of portlet preference names. For example, `<dtportletadmin:getPortletPreferenceNames id="prefNames" portletName="BookmarkPortlet"/>` . The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `portletName` attribute is required and must contain the name of the portlet channel. Returns `java.util.Set`.
`getPortletPreferenceStringValue`	Gets the portlet preference value as a string given a portlet channel name and the preference key. For example, `<dtportletadmin:getPortletPreferenceStringValue id="targetstring" portletName="BookmarkPortlet" prefNam e="targets"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `portletName` attribute is required and must contain the name of the portlet channel. The *prefName* attribute is required and must contain the preference key. Returns `java.lang.String`.

`getPortletPreferenceValues`	Gets the portlet preference values given a portlet channel name and the preference key. For example, `<dtportletadmin:getPortletPreferenceValues id="targets" portletName="BookmarkPortlet" prefName="targets"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `portletName` attribute is required and must contain the name of the portlet channel. The `prefName` attribute is required and must contain the preference key. Returns `java.util.List`.
`setPortletPreferenceStringValue`	Sets the portlet preference value as a string given a portlet channel name and the preference key and value. For example, `<dtportletadmin:setPortletPreferenceStringV portletName="JSPPortlet" prefName="contentpage" value="content.jsp"/>`. The `portletName` attribute is required and must contain the name of the portlet channel. The `prefName` attribute is required and must contain the preference key. The `value` attribute is required and must contain a string value.
`setPortletPreferenceValues`	Sets the portlet preference values given a portlet channel name and the preference key and an array of String values. For example, `<dtportletadmin:setPortletPreferenceValues portletName="BookmarkPortlet" prefName="targets" prefValues= "$prefnamesvalues"/>`.

	The `portletName` attribute is required and must contain the name of the portlet channel. The `prefName` attribute is required and must contain the preference key. The `prefvalues` attribute is required and must contain a string array of values.

Tag library for User Management Tasks

`obtainUserAdmin`	All the user administration tags should be called from within this nested tag. For example, `<dtuseradmin:obtainUserAdmin>...</dtuseradmin:obtainUserAdmin`
`setUserStatus`	Activates or deactivates a user. For example, `<dtuseradmin:setUserStatus userDN="uid=jdcuser,ou=people,dc=red,dc=iplanet,dc=com" activate="true"/>` The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `userdn` is a required attribute. The `activate` attribute is required and takes a boolean to specify activate or deactivate. Returns boolean specifying the status of user.
`resetPassword`	Resets the user's password. For example, `<dtuseradmin:resetPassword userDN="uid=jdcuser,ou=people,dc=red,dc=iplanet,dc=com" newPasswd="jdcuser12"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `userdn` and `newPasswd` attributes are required. Returns a boolean specifying if the reset operation was a success or failure.
`searchUsers`	allows to search for a user. For example, `<dtuseradmin:searchUsers wildcard="j*" id="users"/>`.

	The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `wildcard` attribute is required and takes a search string (if filtering the resulting set based on a regular expression) specifically of the form `foo*`, `*foo`, `foo*bar`, `foo*bar*`. Returns `java.util.Set`.
`getAssignableRoles`	Gets the list of roles that the currently authenticated user can assign/remove. For example, `<dtuseradmin:getAssignableRoles id="nodes"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. Returns `java.util.Set`.
`assignRole`	Assigns a particular role to a user. For example, `<dtuseradmin:assignRole userDN="uid=jdcuser,ou=people,dc=red,dc=iplanet,dc=com" roleDN="cn=JDC,dc=red,dc=iplanet,dc=com"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `roleDN` attribute is required and takes the role to be assigned. The `userDN` attribute is required and takes the distinguished name of the user for whom the role is to be assigned. Returns a boolean specifying if the assign operation is success or failure.
`removeRole`	Removes the assigned role for a user. For example, `<dtuseradmin:removeRole userDN="uid=jdcuser,ou=people,dc=red,dc=iplanet,dc=com" roleDN="cn=JDC,dc=red,dc=iplanet,dc=com"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `userDN` attribute is required and takes the distinguished name of the user for whom the role has to be removed. The `roleDN` attribute is required and takes the role to be removed.

	Returns a boolean specifying if the assign operation is success or failure.
`getUsers`	Gets the list of user distinguished names in the currently logged in administrator's people container. For example, `<dtuseradmin:getUsers id="ExistingUsers"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. Returns `java.util.Set`.
`createUser`	Creates a user given all the required user attributes. For example, `<dtuseradmin:createUser uid="createtest" firstname="create" lastname="test" fullname="test user" password="createtest"/>`. The `uid`, `password`, `fullname`, `firstname`, and `lastname` attributes are required.
`deleteUsers`	Deletes a list of users based on the user distinguished names passed in. For example, `<dtuseradmin:deleteUsers userDNs="$userDNs"/>` where `$userDNs` is a `java.util.Set` of use distinguished names. The *userDNs* attribute is required and takes `java.util.Set` of user DNs to be deleted.
`getUserRoles`	Gets the list of role distinguished nodes the user has already been assigned to. For example, `<dtuseradmin:getUserRoles id="alreadyAssignedUserRoleSet" userDN="uid=jdcuser,ou=people,dc=red,dc=iplanet,dc=com"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. Returns `java.util.Set`.

Tag library for WSRP Management Tasks

`obtainWSRPAdmin`	All the WSRP administration tags should be called from within this nested tag and will be operating on the `baseDN` that is passed in. For example, `<wsrpadmin:obtainWSRPAdmin baseDN="$baseDN">...</wsrpadmin:obtainWSRPAdmin>` . The `baseDN` attribute is required and must be a valid node in Directory Server. For example: `-dc=red,dc=iplanet,dc=com`
`getExistingProducerIds`	Gets the set of producer IDs that are available in the organization of the `baseDN` that this tag is operating on. The producers in this list can be used to get the portlets list. For example, `<wsrpadmin:getExistingProducerIds id="ExistingProducerIds"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. Returns `java.util.Set`.
`getWSRPPortletHandlers`	Returns the list of portlet ids for the given `producerID` . For example, `<wsrpadmin:getWSRPPortletHandlers id="PortletHandlersList" producerID="EO5DM1IAAAAAJAM42KBAAAA"/>`. The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `producerID` attribute is required and takes the id of the producer. Returns `java.util.Set`.
`createWSRPChannel`	Creates a new WSRP portlet channel given the `channelName`, `portletID`, `producerID`. For example: `<wsrpadmin:createWSRPChannel channelName="MyFrontPageTabPanelContainer/jspPortlet producerId="EO5DM1IAAAAAJAM42KBAAAA" portletId="JSPPortlet"/>`. The `channelName`, `producerID`, and `portletID` attributes are required.

`getProducerName`	Returns the name of producer for the given `producerID`. For example: `<wsrpadmin:getProducerName id="producerName" producerID="EO5DM1IAAAAAJAM42KBAAAA"/>.` The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `producerID` attribute is required and takes the id of the producer. Returns `java.lang.String`.
`getPortletName`	Returns the name of portlet for the given `producerID` and `portletID`. For example, `<wsrpadmin:getPortletName id="portletName" producerID="EO5DM1IAAAAAJAM42KBAAAA" portletID="JSPPortlet"/>.` The `id` attribute is optional and takes the name of the exported scoped variable for the resulting value. The `scope` attribute is optional and takes the scope for `id`. The `producerID` attribute is required and takes the id of the producer. The `portletID` attribute is required. Returns `java.lang.String`.

CHAPTER 38

PAR Export File Format

This supplement describes the format of the export files used to specify data to be inserted into a PAR file.

File Format

The export file consists of lines containing a keyword, followed by a colon (`:`) and white space delimited fields. The `from:` line is required and it must be the first line of the file. Lines beginning with `#` are treated as comments.

`from:` *types name*
The `from` line indicates what entity is being exported. The `types` can be `channel`, `provider`, or `channel,provider`, and `channel+provider`. The `name` indicates the channel name, or a provider name if a provider is being exported. The name must be URL encoded if the name contains white space, commas (`,`), colons (`:`), semicolons (`;`), plus signs (`+`), or percent signs (`%`).

`file:` *root*`|.` *path* [*types*]
The `file` line indicates that a file, based on a property setting, is to be included. The property can come from either the desktop properties file, located by default in `/var/opt/SUNWportal/portals/`*portalID*`/config/desktopconfig.properties` file. The `root` specifies the root of the file location and `path` specifies the path to the rest of the file. The `root` is a property name that corresponds to a directory (like). If `root` is given as `.`, the file is assumed to be static content located at the web server's doc root. You can also specify the types of operation the file is to be associated with, defaulting to `channel`. The types can be `channel`, `provider`, or `channel,provider`, and `channel+provider`.

`class:` *class* [*types*]
The `class` line indicates that a class file is to be packaged with the entry, and you may optionally specify the types of operations that the class file are associated with. If not specified, provider is assumed. Types can be `channel`, `provider`, or `channel,provider`, and `channel+provider`; also, when specifying both channel and provider, you can use a space.

`directory:` *root*`|.` *dir* `+|.|`*filter* `[`*types*`]`

The `directory` line implies an entire directory search with all non-directory files to be included as if entered as file lines. It includes the capability of specifying a filter that is a directory component which must be present in recursive directory searches. The `root` specifies the root of the directory, or `.` to indicate static content. The `dir` is the directory underneath the root to search from, which can be given as `.` to start at the root itself. The `filter` specifies the filter component which must be in the directory, which implies a recursive descent. It can be given as `+` for a recursive descent with no filter, or `.` for no recursive descent (just the contents of the actual directory). You can also specify the types of operation, which default to channel. Types can be `channel`, `provider`, or `channel,provider`, and `channel+provider`.

`entry:` *name*

The `entry` line specifies the entry name used in the `.par` file. If not specified, it defaults to the name from the `from:` line.

`desc:` *text*

Any number of `desc` lines may appear. These are concatenated together as a user-visible description packaged with the entry.

CHAPTER 39

Import Operations Format

This supplement describes the PAR import operations list format.

Operations Format

Each operation (`op`) on the command line, must be specified as a pipe (`|`) separated list of keywords that can have values, most of which are optional. The operations are in space separated list. Each operation is in the following format:

`dpnode=`*dn*`,entry=`*name*`,provider[=`*name*`],channel[=`*name*`],container=`*name*`[,avail=`*name*`,selected]`

Here:

`dpnode` Specifies the distinguished name in the directory server (or the keyword global) for the display profile document that this operation is targeted at. This may not apply if the context it is being specified in has already provided this. For example, if the `import` subcommand defines the distinguished name, the distinguished name in the file is ignored.

`entry` Specifies the entry name in the `.par` file. This is not needed if the:

- `.par` file only contains one entry, as the operation defaults to that entry
- Operation is already associated with an entry such as the auto extract option for an entry.

The utility defaults to the first entry in the file if an entry is not specified.

`provider` Indicates that a provider extraction is to take place. If the name is missing, it uses the name packaged with the provider in the `.par` file.

`channel` Indicates that a channel extraction is to take place. If the name is missing, it uses the name provided with the channel in the `.par` file.

`container`	Indicates which container the channel is to be inserted into and applies only to channel extractions. If omitted, the channel is inserted into the channels element at the display profile document root.
`avail`	Indicates a container whose avail (or available) list is to receive a reference to the new channel and applies only to channel extractions. If omitted, no new channel reference is created.
`selected`	Indicates that the container whose avail list received a reference, also has a reference placed in its selected list and applies only if `avail` was used.

If the `op` information is in both the `import` subcommand and in the `.par` file, the command information takes precedence