Sun Java™ System

Portal Server 6 Migration Guide

2004Q2

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

Part No: 817-5320-10

Contents

About This Guide

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

Also, all instances of the Sun™ ONE Identity Server 5.1 product refer to what was formerly known as the iPlanet™ Directory Server Access Management Edition 5.1 product.

This guide explains how to:

- migrate from Sun ONE Portal Server 3.0 software to Sun ONE Portal Server 6.2 software
- upgrade from Sun ONE Portal Server 6.0 or Sun ONE Portal Server 6.1 software to Sun ONE Portal Server 6.2 software
- upgrade from Sun ONE Portal Server 6.2 software to Sun Java System Portal Server 6 2004Q2 software.

Sun Java System Portal Server provides a platform to create portals for your organization's integrated data, knowledge management, and applications. The Portal Server platform offers a complete infrastructure solution for building and deploying all types of portals, including business-to-business, business-to-employee, and business-to-consumer.

This preface includes the following sections:

- Who Should Read This Book
- What You Need to Know
- How This Book is Organized

- Document Conventions Used in This Guide
- Related Third-Party Web Site References
- Where to Find Related Information
- Where to Find This Guide Online

Who Should Read This Book

You should read this book if you are responsible at your site for:

- migrating from Sun ONE Portal Server 3.0 software to Sun ONE Portal Server 6.2 software
- upgrading from Sun ONE Portal Server 6.0 or Sun ONE Portal Server 6.1 software to Sun ONE Portal Server 6.2 software
- upgrading from Sun ONE Portal Server 6.2 software to Sun Java System Portal Server 6 2004Q2 software.

What You Need to Know

Before you migrate your Sun ONE Portal Server 3.0 installation or upgrade your Sun ONE Portal Server 6.0 or Sun ONE Portal Server 6.1 installation, you must be familiar with the following concepts:

- Basic Solaris™ administrative procedures
- LDAP
- HTML
- XML
- JavaServer Pages™
- Sun™ ONE Directory Server
- iPlanet™ Directory Server Access Management Edition 5.1
- Sun™ ONE Identity Server
- Sun™ ONE Web Server
- Sun™ ONE Portal Server 3.0

- Sun™ ONE Portal Server 6.0
- Sun™ ONE Portal Server 6.1

How This Book is Organized

This book is organized into two parts and contains the following chapters and appendices:

- About This Guide (this chapter)
- Part I, "Migrating from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2"
    - Chapter 1, "Planning the Migration"

      This chapter describes the data migration process and provides an overview of the migration tools.
    - Chapter 2, "Migrating LDAP Data"

      This chapter provides instructions for migrating LDAP data.
    - Chapter 3, "Migrating Custom Authentication Modules"

      This chapter provides instructions for migrating custom authentication modules.
    - Chapter 4, "Migrating The Rewriter"

      This chapter provides instructions for migrating the Rewriter.
    - Chapter 5, "Migrating The Desktop"

      This chapter provides instructions for migrating the Desktop, including custom providers.
    - Chapter 6, "Migrating Certificates"

      This chapter provides instructions for migrating certificates.
    - Chapter 7, "Migrating the Gateway, Netlet, and NetFile"

      This chapter provides instructions for migrating the gateway, Netlet, and NetFile from Sun ONE Portal Server 3.0 to Sun™ ONE Portal Server, Secure Remote Access.
    - Chapter 8, "Migrating Sun™ ONE Compass Server"

This chapter provides instructions for migrating Sun™ ONE Compass Server to the Sun ONE Portal Server 6.2 Search Engine.

- Chapter 9, "Changes in Sun ONE Portal Server APIs that Affect Migration"

  This chapter provides information on changes in Sun ONE Portal Server APIs that affect migration.

- Chapter 10, "Enabling Logon to a Different Organization or Suborganization"

  This chapter provides instructions for logging into a different organization or suborganization in Sun ONE Portal Server 6.2.

- Part II, "Upgrading from Sun Java System Portal Server 6.0 to Sun ONE Portal Server 6.2"

  - Chapter 11, "Planning the Upgrade from Sun Java System Portal Server 6.0 to Sun Java System Portal Server 6.2"

    This chapter provides information on the upgrade process to help you understand and plan the upgrade.

  - Chapter 12, "Upgrading from Sun ONE Identity Server 5.1 to Sun™ ONE Identity Server 6.1"

    This chapter provides information on how to migrate data stored on Sun ONE Identity Server 5.1 to Sun ONE Identity Server 6.1.

  - Chapter 14, "Upgrading Sun™ ONE Portal Server Software from Version 6.0 to Version 6.2"

    This chapter provides information on upgrading the Sun ONE Portal Server 6.0 software to Sun ONE Portal Server 6.2 software.

- Part III, "Upgrading from Portal Server 6.1 to Sun ONE Portal Server 6.2"

  - Chapter 15, "Planning the Upgrade from Sun Java System Portal Server 6.1 to Sun Java System Portal Server 6.2"

    This chapter provides information on the upgrade process to help you understand and plan the upgrade.

  - Chapter 13, "Upgrading from Sun ONE Identity Server 6.0 to Sun™ ONE Identity Server 6.1"

    This chapter provides information on how to upgrade Sun ONE Identity Server 6.0 to Sun ONE Identity Server 6.1.

  - Chapter 16, “Upgrading Sun™ ONE Portal Server Software from Version 6.1 to Version 6.2”

    This chapter provides information on upgrading the Sun ONE Portal Server 6.1 software to Sun ONE Portal Server 6.2 software.

- Part IV, “Upgrading from Sun ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2”

  - Chapter 17, “Planning the Upgrade from Sun ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2”

    This chapter provides information on the upgrade process to help you understand and plan the upgrade.

  - Chapter 18, “Upgrading from Sun ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2”

    This chapter provides information on upgrading the Sun ONE Portal Server 6.2 software to Sun Java System Portal Server 6 2004Q2 software.

- Appendix A, “Tool Design”

  This appendix describes the design of the Sun ONE Portal Server 3.0 Data Migration Tool Suite and the Sun ONE Compass Server migration tools.

- Appendix B, “Sun ONE Portal Server 3.0 Data Migration Module Author's Guide”

  This appendix provides information necessary for writing migration modules to be used by the migration tools.

- Appendix C, “Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2”

  This appendix provides a reference for the public API changes between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2 that affect migration.

- Appendix D, “Tag and String Changes Made in Templates and JSPs by the Migration Tools”

  This appendix provides a reference detailing the changes made by the migration tools to templates, JSPs, tags, and strings.

- Appendix E, “Extended Sun ONE Portal Server 3.0 Component Migration”

  This appendix provides information on migrating components that have been extended in Sun ONE Portal Server 3.0.

- Appendix F, “Authentication Framework Changes Between Sun ONE Portal Server 6.0 and Sun ONE Portal Server 6.2”

  This appendix provides reference information on the changes to the authentication framework between Sun ONE Portal Server 6.0 and Sun ONE Portal Server 6.2 for migrating custom authentication.

- Appendix G, “Example of Sun ONE Portal Server 3.0 Provider Migrated to Sun ONE Portal Server 6.x”

  This appendix provides sample code for a quotations provider for Sun ONE Portal Server 3.0 and sample code for the same provider migrated for Sun ONE Portal Server 6.x.

Document Conventions Used in This Guide

Monospaced Font

`Monospaced font` is used for any text that appears on the computer screen or text that you should type. It is also used for file names, distinguished names, functions, and examples.

Bold Monospaced Font

Sun ONE Portal Server**`Bold monospaced font`** is used to represent text within a code example that you should type. For example, you might see something like this:

```
./setup
*******************************************************************

Sun(TM) One Portal Server (6.2 release)

*******************************************************************

Installation log at /var/sadm/install/logs/ipsinstall.13343/install.log

This product will run without a license. However, you must either purchase
a Binary Code License from, or accept the terms of a Binary Software
Evaluation license with, Sun Microsystems, to legally use this product.

Do you accept? yes/[no] Starting install wizard in graphical mode
```

In this example, **./pssetup** is what you would type from the command line and the rest is what would appear as a result.

Italicized Font

An *italicized font* is used to represent text that you enter using information that is unique to your installation (for example, variables). It is used for server paths and names and account IDs.

Square or Straight Brackets

Square (or straight) brackets [] are used to enclose optional parameters. For example, in the Sun ONE Portal Server documentation, you will see the usage for the dpadmin command described as follows:

```
dpadmin list|modify|add|remove [command-specific options]
```

The presence of [command-specific] indicates that there are optional parameters that may be added to the dpadmin command.

Command-Line Prompts

Command-line prompts (for example, % for a C-Shell, or $ for a Korn or Bourne shell) are not displayed in the examples. Depending on which operating system environment you are using, you will see a variety of different command-line prompts. However, you should enter the command as it appears in the document unless specifically noted otherwise.

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 by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Where to Find Related Information

In addition to this guide, Sun ONE Portal Server comes with supplementary information for administrators as well as documentation for developers. Use the following URL to see all the Sun ONE Portal Server documentation:

`http://docs.sun.com/prod/s1portalsrv`

Listed below are the additional documents released with the Sun Java Sysem Portal Server 6 2004Q2 documentation suite:

- *Sun Java System Portal Server 6 2004Q2 Release Notes*
- *Sun Java System 6 2004Q2 Administrator's Guide*
- *Sun Java System Portal Server Secure Remote Access 6 2004Q2 Administrator's Guide*
- *Sun Java System Portal Server 6 2004Q2 Deployment Guide*
- *Sun Java System Portal Server 6 2004Q2 Desktop Customization Guide*
- *Sun Java System Portal Server 6 2004Q2 Developer's Guide*

Where to Find This Guide Online

You can find the *Sun Java System Portal Server 6 2004Q2 Migration Guide* online in PDF and HTML formats. This book can be found at the following URL:

h`http://docs.sun.com/prod/s1portalsrv`

Part I

Migrating from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2

Chapter 1

Planning the Migration

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

Also, all instances of the Sun™ ONE Identity Server 5.1 product refer to what was formerly known as the iPlanet™ Directory Server Access Management Edition 5.1 product.

This chapter provides an overview and discussion on how to plan for converting users, roles, domains, components, templates, providers, channels, properties files, and packs from Sun ONE Portal Server 3.0 to Sun™ ONE Portal Server 6.2.

This chapter contains the following sections:

- Overview of Data Migration
- The Migration Process
- Migration Checklist
- Migration Tools Overview

Overview of Data Migration

Several features in Sun ONE Portal Server 6.2 product require format changes in the data store of Sun ONE Portal Server 3.0 because the Sun ONE Portal Server 6.2 software uses new access layer and APIs provided by Sun ONE Identity Server 6.1 release. The Sun ONE Portal Server 3.0 Data Migration Tool Suite provided with Sun ONE Portal Server 6.2 release enables you to migrate your LDAP data, templates, JavaServer Pages™ (JSP™), resource bundles, certificates (with the exception of gateway certificates), authentication module data, and properties files fromSun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2 or to Sun ONE Identity Server as necessary. The Sun ONE Portal Server 3.0 Data Migration Tool Suite does not migrate gateway certificates.

These are the high-level steps to migrate your data from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2. (Subsequent chapters provide step-by-step instructions for performing the actual migration.)

1. Perform a full installation of Sun ONE Portal Server 6.2 including the Sun ONE Portal Server 3.0 Data Migration Tool Suite.

NOTE The Sun ONE Portal Server 6.2 product supports migration of a Sun ONE Portal Server 3.0 installation to a Sun ONE Portal Server 6.2 installation deployed on a the Sun™ ONE Web Server (which uses the Java™ 2 SDK (J2SDK™), Standard Edition 1.4.1_05 for its Java™ run-time environment) as well as Sun™ ONE Application Server, IBM Websphere application server, and BEA Weblogic application server.

2. Install the Sun ONE Portal Server 3.0 Data Migration Tool Suite on a system that has an existing version of Sun ONE Portal Server 3.0 already installed.

   If you are migrating from a Sun ONE Portal Server 3.0 system to a separateSun ONE Portal Server 6.2 system, you need two installations of the Sun ONE Portal Server 3.0 Data Migration Tool Suite, one for each system.

   If you are performing a single-system migration, you can install Sun ONE Portal Server 6.2 on a Sun ONE Portal Server 3.0 system.

3. Install the Sun™ ONE Compass Server migration tools if you need to migrate a Sun ONE Compass Server. Sun ONE Compass Server migration tools are separate from the Sun ONE Portal Server 3.0 Data Migration Tool Suite. See Chapter 8, “Migrating Sun™ ONE Compass Server” for details on migrating Sun™ ONE Compass Server 3.01C, Service Pack 1 or later to Sun ONE Portal Server 6.2.

4. Run the export tool on the Sun ONE Portal Server 3.0 system. This script stores the existingSun ONE Portal Server 3.0 data to the file system. This data collection is an automated process.

5. Move the exported data from the Sun ONE Portal Server 3.0 system to the Sun ONE Portal Server 6.2 system if you are using two separate machines.

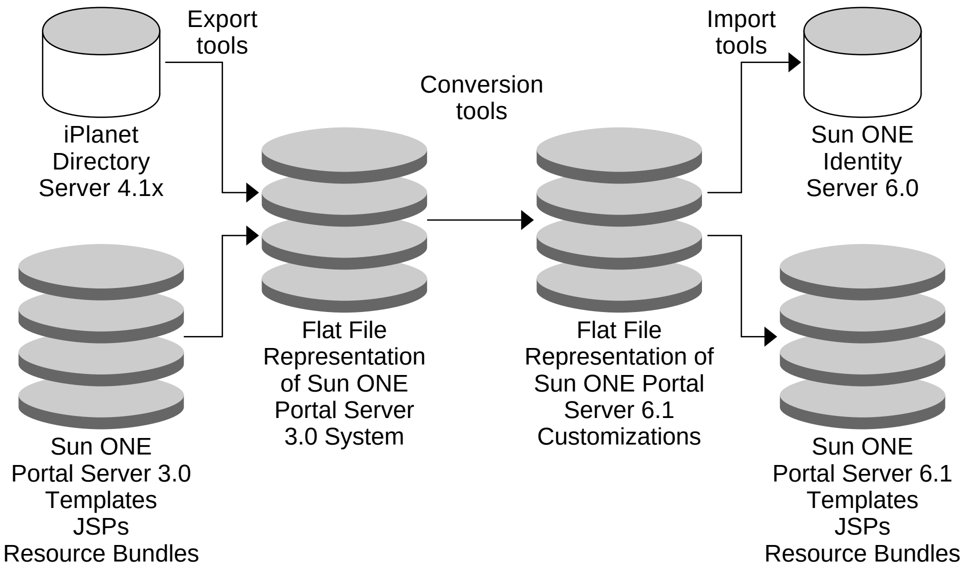
6. Run the conversion tool on the Sun ONE Portal Server 6.2 system. This script modifies the Sun ONE Portal Server 3.0 data exported using the export tool so that the data is usable by Sun ONE Portal Server 6.2. This process might require manual intervention since there is no way to know all possible customizations and their desired effect by the portal server that you have envisioned.

7. Run the import tool on theSun ONE Portal Server 6.2 system. This script imports the data which was exported and converted using the export and conversion tools. This process is automated requiring little to no intervention.

NOTE You can add functionality to the core tools by writing your own custom migration module and installing it into the migration tool suite. The migration tools will detect the modules and update user menu options as needed. See Appendix B, "Sun ONE Portal Server 3.0 Data Migration Module Author's Guide" for more information on migration modules.

Figure 1-1 on page 24 provides an overview of the migration process. It shows that the export tools take data from iPlanet Directory Server 4.1x and templates, JSPs, and resource bundles from Sun ONE Portal Server 3.0 and store them as a flat file representation of theSun ONE Portal Server 3.0 system. The conversion tools convert the flat file representation of Sun ONE Portal Server 3.0 into a flat file representation of Sun ONE Portal Server 6.2 customizations. The import tools then import LDAP data into Sun ONE Identity Server, and the templates, JSPs, and resource bundles into Sun ONE Portal Server 6.2.

Figure 1-1 Overview of Migration Process


Export tools
Import tools
Conversion tools
iPlanet Directory Server 4.1x
Sun ONE Identity Server 6.0
Flat File Representation of Sun ONE Portal Server 3.0 System
Flat File Representation of Sun ONE Portal Server 6.1 Customizations
Sun ONE Portal Server 3.0 Templates JSPs Resource Bundles
Sun ONE Portal Server 6.1 Templates JSPs Resource Bundles

Throughout the migration process, the tools produce a report listing actions taken or not taken, along with warnings and errors. You can use this report to troubleshoot the migration.

The Migration Process

This section describes how the migration process works and outlines what you need to know before beginning the migration process

How the Migration Process Works

You migrate a system by using three command-line scripts, or tools: the export tool, the conversion tool, and the import tool. This section describes how these tools work. See Appendix A, “Tool Design” for the syntax of these scripts.

Export Tool

The export tool performs the following tasks:

- Retrieves and stores to the export directory all component, domain, role, and user data used by the Sun ONE Portal Server 3.0 system.
- Adds all templates and JSPs from the `/etc/opt/SUNWips/desktop` directories on the Sun ONE Portal Server 3.0 system to the template export tar file in the export directory.
- Adds new or modified authentication templates from the `/etc/opt/SUNWips/auth` directory on the Sun ONE Portal Server 3.0 system to the template export tar file in the export directory.
- Copies resource bundles from the /*BaseDir*/`SUNWips/locale` directory on the Sun ONE Portal Server 3.0 system to the template export tar file in the export directory.
- Adds contents of the web server document root from the /*BaseDir*/`SUNWips/public_html` directory, except certain files specific to the Sun ONE Portal Server 3.0 installation, to the template export tar file in the export directory.
- Saves web server certificates to the export directory. The migration tools do not migrate gateway certificates.
- Runs commands as required by external modules.

Conversion Tool

The conversion tool performs the following tasks:

- Converts Sun ONE Portal Server 3.0 components to Sun ONE Identity Server services.
- Converts Sun ONE Portal Server 3.0 domains to Sun ONE Identity Server organizations.
- Converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server suborganizations. You have the option to choose the default of suborganizations or flat role structure when migrating Sun ONE Portal Server 3.0 roles.
- Converts Sun ONE Portal Server 3.0 users to Sun ONE Identity Server users.
- Converts Desktop, provider, and channel customizations to display profile XML fragments.
- Converts gateway rules to Rewriter rulesets.
- Bundles any gateway rules not in Sun ONE Portal Server 6.2 into a custom ruleset.

- Translates tags and URLs in Desktop templates.
- Translates tags, URLs, and function calls deprecated by theSun ONE Portal Server 6.2 JSP implementation in JSP files.
- Translates SRC values in HTML files from web server documents.
- Removes the title and description from resource bundles.
- Lists custom Desktop code in the migration report.
- Lists custom provider definitions in the migration report to state that you need to recompile the code.
- Copies certificate databases, with the exception of gateway certificates, to the import directory.
- Converts other data using external modules.

Import Tool

The import tool performs the following tasks:

- Imports all converted service, organization, role, and user data into Sun ONE Identity Server.
- Copies converted Desktop templates to the `/etc/opt/SUNWps/desktop` directory on the Sun ONE Portal Server 6.2 system.
- Copies converted resource bundles to *BaseDir*`/SUNWps/web-src/WEB-INF/classes` on the Sun ONE Portal Server 6.2 system.
- Copies certificate databases, with the exception of gateway certificates, to the proper location on theSun ONE Portal Server 6.2 system.
- Imports other data using external modules.
- Redeploys the web application at the completion of the import process.

What You Need to Know Before Beginning the Migration Process

Use the following information to help plan the migration process.

- All data to be migrated by the migration tools, including templates, JSPs, and resource bundles reside on a Sun ONE Portal Server 3.0 system. LDAP data may reside either on the Sun ONE Portal Server 3.0 system or on an external system.
- All data to be migrated by these tools will be converted to work with Sun ONE Portal Server 6.2 on Sun ONE Identity Server.
- Sun ONE Portal Server 6.2 provides only command-line migration tools. There is no graphical interface for these tools.
- Sun ONE Portal Server 6.2 can install everything, including the migration tools. The installation will provide you with a menu so you can choose what you need.
- You will need to perform additional manual steps for custom providers and templates prior to using them in Sun ONE Portal Server 6.2.
- If the Sun ONE Portal Server 6.2 machine name is different from the Sun ONE Portal Server 3.0 machine name, certificates will need to be reissued.
- Sun ONE Portal Server 6.2 now includes the Sun ONE Compass Server functionality as a core search engine. Therefore, do not install Sun ONE Compass Server on the Sun ONE Portal Server 6.2 node. Sun ONE Portal Server 6.2 refers to the functionality of Sun ONE Compass Server as Search or Search Engine.
- There is a minimum free disk space requirement of 500 MB. This requirement is for the export and conversion tools.
- Only migration from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2 as described in this guide is supported.
- Sun ONE Portal Server 3.0 must be unavailable during migration. If the Sun ONE Portal Server 3.0 system is in production during migration, after you run the export tool any updates to the data will not be migrated.
- For single-system migration, you must use different port numbers for the Sun ONE Portal Server 6.2 instance from the port numbers used for the Sun ONE Portal Server 3.0 instance.
- The migration tools export symbolic links from the Sun ONE Portal Server 3.0 system and preserve them during the migration to the Sun ONE Portal Server 6.2 system. The tools do not, however, migrate the contents of the symlink target to the Sun ONE Portal Server 6.2 system. You must manually copy or move the contents of the symlink target to the Sun ONE Portal Server 6.2 system.

Migration Checklist

1. Will you migrate from one system to another? Or will you perform a single-system migration?
2. Are there customizations to the Desktop templates that you need to migrate?
3. Is there any custom provider code? If so, this will need to be compiled and added manually.
4. Are there customizations to the authentication modules that you need to migrate?
5. Is the name of the Sun ONE Portal Server 6.2 system the same as the name of the Sun ONE Portal Server 3.0 system from which you are migrating? If not, you need new certificates.
6. Do you have test systems set up for the migration that you can put into production after the migration is complete in order to avoid down time?
7. When will the migration begin?
8. When do you expect the migration to finish?
9. How many domains are you migrating?
10. How many users are you migrating?
11. How many roles are you migrating? What is this impact on your migrating given the new role concept in Sun ONE Identity Server?
12. How many customized providers do you need to migrate manually to the new provider APIs (PAPI)?
13. Do you have a multi-node environment that requires additional planning for migration?
14. Do you have an existing installation of Sun ONE Compass Server that requires migration?
15. What other manual steps do you need to perform for successful migration (outside of the automated steps via the migration tools available)?

Migration Tools Overview

This section describes how the individual migration tools work and the options you have for migrating data. The migration package is SUNWpsmig and you need to install it on both the Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2 systems.

Export Tool

The export tool:

- Prompts the user for a directory where the tool saves all exported data.
- Verifies that adequate disk space is available.
- Verifies that Sun ONE Portal Server 3.0 is installed on the system.
- Creates the directories where the tool stores the exported Sun ONE Portal Server 3.0 data.
- Uses Sun ONE Portal Server 3.0 package information and system configuration files to find variables to be used for exporting data.
- Exports all variables.
- Searches for modules.
- Displays the menu for user selection if needed.

Export Tool Modules

The export tool uses modules to create the script menus. This modularity enables you to create new modules and add them to the menu. The export tool creates the executable files. The export tool also creates the export menu from the text returned from invoking each module with the --menu option.

- Menu order is dependant on the first two digits of the module file name in ascending numerical order. 00 is the first possible menu item, 99 the last. In the case where two or more items share the same number, the menu for those items is in ascending alphabetical order based on filename.
- The export tool does not add a menu item for those modules that return no text. These modules are considered “run-always.”
- The export tool adds a menu item for All of the above at the end of the module menu.

- The export tool adds a menu item for `Exit` as the final option

The export tool runs the modules based on user selection.

- Run order is dependant on the first two digits of the module file name in ascending numerical order. The module whose file name begins with `00` is the first module executed. The module whose file name begins with `99` is the last module executed. In the case where two or more module file names share the same number, the export tool executes these modules in ascending alphabetical order.

The export tool runs any modules that do not return a menu item along with a single module user selection.

Files are considered export modules when they meet the following criteria:

- They exist under the installed migration directory `/modules.`
- They end with the text "`export`."
- They are executable.
- They have the digits 0-9 as the first two characters of module file name.
- They return a single line of text or nothing when invoked with the `--menu` option.

LDAP Database Export Module

The LDAP database export module:

- Alerts you to the fact that the LDAP database tool `ldapsearch` is required to export the LDAP database if `ldapsearch` does not exist on the system.
- Saves the LDIF representation of the database.
- Uses `ipsadmin` to export XML files for domains, roles, and components.

Flat File Export Module

The flat file export module creates a tar file that contains:

- Desktop templates
- Resource bundles
- Web server root documents
- Modified and new authentication templates

Certificate Export Module

The certificate export module copies:

- The web server certificate database, if it exists, to the export directory. The certificate export module does not copy gateway certificates.

Conversion Tool

The conversion tool:

- Prompts for the location of the exported data and verifies that it contains the results of a successful export of Sun ONE Portal Server 3.0 data.
- Prompts for theSun ONE Identity Server Internal LDAP Authentication password.
- Prompts for an import directory where the tool saves all converted data.
- Verifies that adequate disk space is available in the import directory.
- Creates the directories to store the converted Sun ONE Portal Server 3.0 data.
- Verifies that a proper installation of Sun ONE Portal Server 6.2 exists on the system.
- Converts all variables.
- Searches for modules.
- Displays the menu for user selection if needed.

Conversion Tool Modules

The conversion tool uses modules to create the script menus. This modularity enables you to create new modules and add them to the menu. The conversion tool creates the executable files. The conversion tool also creates the export menu from the text returned from invoking each module with the `--menu` option.

- Menu order is dependant on the first two digits of the module file name in ascending numerical order. 00 is the first possible menu item, 99 the last. In the case where two or more items share the same number, the menu for those items is in ascending alphabetical order based on filename.
- The conversion tool does not add a menu item for those modules that return no text. These modules are considered “run-always.”

- The conversion tool adds a menu item for `All of the above` at the end of the module menu.
- The conversion tool adds a menu item for `Exit` as the final option.

The conversion tool runs the modules based on user selection.

- Run order is dependant on the first two digits of the module file name in ascending numerical order. The module whose file name begins with `00` is the first module executed. The module whose file name begins with `99` is the last module executed. In the case where two or more module file names share the same number, the conversion tool executes these modules in ascending alphabetical order.

The conversion tool runs any modules that do not return a menu item along with a single module user selection.

Files are considered conversion modules when they meet the following criteria:

- They exist under the installed migration directory `/modules.`
- They end with the text “`convert`.”
- They are executable.
- They have the digits 0-9 as the first two characters of module file name.
- They return a single line of text or nothing when invoked with the `--menu` option.

LDAP Database Conversion Module

The LDAP database conversion module:

- Converts all domain data from the Sun ONE Portal Server 3.0 system into organizations for Sun ONE Portal Server 6.2. The tool creates organization XML files for importing using the `amadmin` program.
- Converts all role data from Sun ONE Portal Server 3.0 into suborganizations or roles for Sun ONE Portal Server 6.2. The tool creates organization XML files for importing using the `amadmin` program.
- Converts all user data from the Sun ONE Portal Server 3.0 system into users for Sun ONE Portal Server 6.2. The tool creates an LDIF file for importing into the Sun ONE Identity Server instance.
- Converts provider and channel entries from the exported LDAP database data into display profile fragments and stores them in the import directory.
- Saves component entries in the appropriate service file.

- Saves domain entries in the appropriate organization file.
- Saves role entries in the appropriate role file.
- Saves user entries in the appropriate user file.
- Checks each provider and channel component for a Sun ONE Portal Server 6.2 equivalent.

  A list of all providers and channels is available in the Sun ONE Portal Server 3.0 attribute `iwtDesktop-allProviders`. The values have the format `providerName|className`.

  A provider is the first `providerName` for a unique `className.A` channel is the second and later to use a unique `className`.
- Checks attributes of providers and channels to determine customization.
- Checks each `className` for a Sun ONE Portal Server 6.2 equivalent and notes in the migration report each custom `className`.
- Creates service definition files based on known component translations.
- Places in the migration report a list of components that appear to be imported for a customer application.

  Included in this report are components that have not been identified as a provider, channel, or authentication module in addition to `iwtGateway`, `iwtDesktop`, `iwtNetlet`, and `iwtNetfile` components.

NOTE Any special use of domains, roles, and users might require manual conversion.

Gateway Rules to Rewriter Rulesets Conversion Module

The gateway rules to Rewriter rulesets conversion module:

- Places any rules not defined as part of the default Sun ONE Portal Server 6.2 ruleset in a new “custom” ruleset.
- Creates a file to reflect this new “custom” ruleset in the import directory.
- Updates the rulesets for channels that use `URLScraper` or `XMLprovider` based on Rewriter migration. The module also notes in the migration report that any channels that use or extend these classes should have their rulesets updated.

Desktop Conversion Module

The Desktop conversion module:

- Extracts the flat file tar file into a temporary directory. All operations described here pull templates from this temporary directory and place changes in another temporary directory for archiving after all conversion is complete.
- Copies all files from the temporary export directory to the temporary import directory, preserving ownership and permissions whenever possible.
- Scans templates for invalid tags. The module notes these templates in the migration report. If the template is used by a provider that has a Sun ONE Portal Server 6.2 equivalent, the module notes this fact so that you can check for customization you may have made to the templates.
- Converts Sun ONE Portal Server 3.0 tags to Sun ONE Portal Server 6.2 tags for all files in the `desktop` directory. The module notes modifications to templates in the migration report.
- Changes file names from `.html` to `.template` for all files in the `desktop` directory.
- Changes `DesktopServlet` URLs to the Desktop servlet name defined by the Sun ONE Portal Server 6.2 installation for all files in the `desktop` directory.
- Changes links to static content in the URL defined by the Sun ONE Portal Server 6.2 installation for all files in the `desktop` directory.
- Removes deprecated APIs from JSPs within the `desktop` directory. The module notes modifications to JSPs in the migration report.
- Changes links to static content within `.html` files in the web server document directory.
- Copies each resource bundle from the export directory to the import directory.
- Creates a tar file from the temporary import directory and places it in the import directory.

Certificate Conversion Module

The certificate conversion module copies web server certificates from the export directory to the import directory. It does not copy gateway certificates.

Import Tool

The import tool:

- Prompts for the directory containing all converted data.
- Prompts for theSun ONE Identity Server Internal LDAP Authentication password.
- Verifies that a proper installation of Sun ONE Portal Server 6.2 exists on the system.
- Exports all variables.
- Searches for modules.
- Displays the menu for user selection if needed.
- Deploys the data.

Import Tool Modules

The import tool uses modules to create the script menus. This modularity enables you to create new modules and add them to the menu. The import tool creates the executable files. The import tool also creates the export menu from the text returned from invoking each module with the `--menu` option.

- Menu order is dependant on the first two digits of the module file name in ascending numerical order. 00 is the first possible menu item, 99 the last. In the case where two or more items share the same number, the menu for those items is in ascending alphabetical order based on filename.
- The import tool does not add a menu item for those modules that return no text. These modules are considered “run-always.”
- The import tool adds a menu item for `All of the above` at the end of the module menu.
- The import tool adds a menu item for `Exit` as the final option.

The import tool runs the modules based on user selection.

- Run order is dependant on the first two digits of the module file name in ascending numerical order. The module whose file name begins with 00 is the first module executed. The module whose file name begins with 99 is the last module executed. In the case where two or more module file names share the same number, the import tool executes these modules in ascending alphabetical order.

The import tool runs any modules that do not return a menu item along with a single module user selection.

Files are considered import modules when they meet the following criteria:

- They exist under the installed migration directory `/modules`.
- They end with the text “`import`”
- They are executable.
- They have the digits 0-9 as the first two characters of the module file name.
- They return a single line of text or nothing when invoked with the `--menu` option.

LDAP Database Import Module

The LDAP database import tool module:

- Looks for a configuration file to find a Sun ONE Identity Server installation.
- Alerts you to the fact that you must run the tool on the node running Sun ONE Identity Server if a Sun ONE Identity Server does not exist on the node. The files in the import directory can be used on any Sun ONE Identity Server node.
- Uploads customizations into the Sun ONE Identity Server node.

Flat Files Import Module

The flat file import module extracts the following from the import tar file to the appropriate locations:

- Desktop templates
- Resource bundles
- Web server root documents
- Modified and new authentication templates

Certificate Import Module

The certificate import module:

- Copies the web server certificate database, if it exists, to the appropriate location. It does not copy the gateway certificate database.

NOTE The import directory must have read and write permissions. Without write permissions, the import tool will exit when trying to create the log files. Without read permissions, the import script will be unable to read all the directory contents and there will be errors throughout the import process.

You need to be logged in as `root` when importing data from a local directory or ensure that the machine that is remotely mounted recognizes the user as root. If you are not concerned about security, you may open read permission recursively on the import directory by doing the following.

```
chmod -R a+r ImportDir
```

Table 1-1 shows the destination of migrated data on the Sun ONE Portal Server 6.2 system. This two-column table lists the data types in the first column and the destination directory in Sun ONE Portal Server 6.2.

Table 1-1 Destination of Migrated Data on the Sun ONE Portal Server 6.2 System

Data Type	Destination Directory in Sun ONE Portal Server 6.2
Desktop data	`/etc/opt/SUNWps/desktop`
Resource bundles	*BaseDir*`/SUNWps/web-src/WEB-INF/classes`
Sun ONE Portal Server 3.0 user documents from *BaseDir*`/SUNWips/public_html/docs`	*BaseDir*`/SUNWps/web-src/docs`
Static web data	*BaseDir*`/SUNWam/public_html`

Chapter 2

Migrating LDAP Data

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

This chapter provides procedures to migrate LDAP data from Sun ONE Portal Server 3.0 to the Sun™ ONE Identity Server 6.2. In particular, migration of LDAP data includes the migration of Sun ONE Portal Server 3.0 domains, roles, and users to Sun ONE Identity Server organizations, suborganizations, and users respectively. This chapter contains the following sections:

- Using the Export Tool
- Using the Conversion Tool
- Using the Import Tool
- Writing Data Migration Modules
- Setting the Sun ONE Portal Server 6.2 Organization to the Sun ONE Portal Server 3.0 Domain
- Post-Migration Issues and Tasks

Before beginning the migration process, ensure that you have installed the Sun™ ONE Portal Server 6.2 migration tools on your Sun ONE Portal Server 3.0 system and on your Sun ONE Portal Server 6.2 system. See the *Sun ONE Portal Server* 6.2 *Installation Guide* for instructions on installing both Sun ONE Portal Server 6.2 and the Sun ONE Portal Server 3.0 Data Migration Tool Suite.

NOTE The authentication menu attribute provided by Sun ONE Identity Server 5.1 admin console is not supported in Sun ONE Identity Server 6.1 release. If you need to configure a menu of valid authentication modules, use the admin console to set each authentication module with the same level in the authentication level attribute. Users can access a page with all the authentication modules for a specific authentication level by specifying the authlevel in the login URL. For example, a user performs a login with the following syntax:

`http://`*HOSTNAME*:*PORT*/*DEPLOY_URI*`/UI/Login?authlevel=`*N*

Refer to the *Sun ONE Portal Server 6.2 Administrator's Guide* for information on configuring authentication modules.

As described in Chapter 1, "Planning the Migration," there are three steps to the migration process. You will need to run the export tool, the conversion tool, and the import tool to migrate the LDAP data.

Using the Export Tool

The export tool, `exportps`, stores data from the Sun ONE Portal Server 3.0 system in an export directory. To migrate LDAP data, you need to begin by running the export tool to gather the LDAP data and store it to disk.

To Run the Export Tool

1. On the Sun ONE Portal Server 3.0 system, change to the migration tool directory:

   `cd` *BaseDir***`/SUNWps/migration/bin/`**

2. Run the `exportps` command:

   `./exportps [-a] [-p` *LDAP passphrase***`] [`***ExportDir***`]`**

   Table 2-1 describes the options available for exporting LDAP data. This two-column table lists the options in the first column and their descriptions in the second column.

Table 2-1 Options Available for Exporting LDAP Data

Option	Description
`-a`	Runs all modules without a menu. To choose to export only LDAP data from the export menu, do not use the `-a` option.
`-p` *LDAP passphrase*	Sets the LDAP passphrase in order to avoid the LDAP passphrase prompt. If you do not use the `-p` *LDAP passphrase* option with the `-a` option, the tool will prompt you for the LDAP passphrase.
ExportDir	Specifies the directory where the information is to be saved. If you do not use the *ExportDir* option, the tool prompts you for the export directory.

3. The system displays a message similar to the following:

   ```
   Which directory should be created to store the Portal Server system?
   [/tmp/psExport]
   ```

   You see this prompt only if you do not specify the output directory on the command line. You can choose the default directory `/tmp/psExport` or type the export directory.

   Type the directory and press Enter.

4. The system displays a message similar to the following:

   ```
   Delete the directory /tmp/psExport?
   ```

   You see this message only if there is already exported data in the `ExportDir` directory. If you have not already exported the LDAP database, you can keep the export directory without risking any conflicts in the data. If you have already exported the LDAP database, you should delete the directory to avoid conflicts.

5. The system displays a message similar to the following:

```
Found iPS version 3.0sp3a

Begin export process at  Tue Jul  9 10:10:04 PDT 2003

Error file: /tmp/psExport/logs/error.7644

Report file: /tmp/psExport/logs/report.7644

Metrics file: /tmp/psExport/logs/export_metrics.7644


Export Menu:

1) LDAP Database

2) Desktop

3) Certificate Databases

4) All of the above

5) Exit

Select one of the listed options to export:
```

   You see the export menu only if you do not specify the `-a` option.

6. Type **1** and press Enter to export LDAP data only.

7. The system displays the following message:

   ```
   Enter the LDAP admin passphrase :
   ```

   You see this prompt only if you select a full export using the `-a` option and you do not specify the `-p` *LDAP passphrase* option.

   Type the LDAP admin passphrase and press Enter.

   You see output similar to the following:

   ```
   Dumping the ldap database
     organizationalPerson
     user
     role
     domain
     application
   Dumping xml
   .....................


   Successful completion of export process at  Tue Jul  9 10:21:25 PDT
   2003
   ```

8. Change to the export directory you specified in Step 2 or Step 3. For example:

   `cd` *ExportDir*

   Substitute the export directory for *ExportDir*. If you used the default export directory, for example, use `/tmp/psExport`.

   Type **`ls`** to see the directories created by the export tool.

   Table 2-2 on page 44 shows the directories created by the export tool. This two-column table lists the directories in the first column and their descriptions in the second column.

Table 2-2 Directories Created by the Export Tool When Exporting LDAP Data

Directory	Description
ExportDir/`config`	Contains exported configuration data
ExportDir/`ldif`	Contains exported LDIF files
ExportDir/`logs`	Contains `error.PID`, `export_metrics.PID`, and `report.PID` files (where `PID` is the process ID)
ExportDir/`xml`	Contains exported XML data

After the export has completed, or if you run into problems, check the *ExportDir*/`logs` directory for the `report.PID`, `error.PID`, and `export_metrics.PID` files (where `PID` is the process ID). The `report.PID` file contains actions the export tool has or has not taken. The `error.PID` file contains warnings or errors so that you can correct any problems. The `export_metrics.PID` file contains metrics detailing when various elements of the export tool started, stopped, and the total time it took to export the data. You can look at the `stdout` header to know which log and report files to examine for the `exportps` command you are running.

9. Examine the exported data and manually change, before converting the data, all instances of the name of the Sun ONE Portal Server 3.0 system to the name of the Sun ONE Portal Server 6.2 system. If the port numbers on the two systems are different, you must change these manually as well.

   If you are performing a single-system migration, server names will be the same, but port numbers will be different. You need to change port numbers manually before converting the data.

NOTE The migration tools do not modify server names and port numbers.

For example, in the following extract of the exported data in the *ExportDir*/config/platform.config file, you need to change manually all instances of host1.siroe.com, which is the name of the Sun ONE Portal Server 3.0 system, to the name of the Sun ONE Portal Server 6.2 system if the systems use different names. If the systems use different port numbers, you need to change them as well.

```
ips.defaultDomain=siroe.com

ips.server.protocol=http

ips.server.host=host1.siroe.com

ips.server.port=8080

ips.profile.host=host1.siroe.com

ips.profile.port=8080

ips.gateway.protocol=https

ips.gateway.host=host1.siroe.com

ips.gateway.port=443

ips.gateway.trust_all_server_certs=false

ips.virtualhost=host1.siroe.com 192.18.66.42

ips.naming.url=http://host1.siroe.com:8080/namingservice

ips.notification.url=http://host1.siroe.com:8080/
notificationservice
```

Using the Conversion Tool

The conversion tool, convertps, enables you to select which data to convert from the data exported from a valid installation of Sun ONE Portal Server 3.0. Because the conversion tool runs on the Sun ONE Portal Server 6.2 system, you need to move the export directory from the Sun ONE Portal Server 3.0 system to the Sun ONE Portal Server 6.2 system.

To Run the Conversion Tool

1. On the Sun ONE Portal Server 3.0 system, change to the directory above the `export` directory containing the data output by the export tool.

   For example, if you used the default directory (`/tmp/psExport`) for exporting the Desktop, you would type:

   ```
   cd /tmp
   ```

2. Save the export directory using the `tar` command. For example:

   ```
   tar cvf export.tar psExport
   ```

3. Use an FTP program to transfer the `export.tar` file to the Sun ONE Portal Server 6.2 system.

4. On the Sun ONE Portal Server 6.2 system, extract the files from `export.tar` using the `tar` command. For example:

   ```
   tar xvf export.tar
   ```

5. On the Sun ONE Portal Server 6.2 system, change to the migration tool directory:

   ```
   cd BaseDir/SUNWps/migration/bin
   ```

6. Run the `convertps` command:

   ```
   ./convertps [-a] [-f] [-i ExportDir] [-o ImportDir]
   ```

   Table 2-3 describes the options available for the `convertps` command. This two-column table lists the options in the first column and their descriptions in the second column.

Table 2-3 Options Available for Converting LDAP Data

Option	Description
`-a`	Runs all modules without a menu. To choose to convert only the LDAP database from the conversion menu, do not use the `-a` option.

Table 2-3 Options Available for Converting LDAP Data *(Continued)*

Option	Description
`-f`	Converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server roles and places users under the organization. This option retains the multiple role to user assignment similar to Sun ONE Portal Server 3.0, but does not retain the hierarchical role functionality. All roles are created under the organization and are not prioritized. The display profile documents are prioritized and merged accordingly. To see if the roles have been migrated to roles, view the roles under the organization in the admin console. If you do not use the `-f` option, the conversion tool converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server suborganizations and places users under the suborganization. Without the `-f` option, the conversion tool retains the hierarchical functionality and customizations from Sun ONE Portal Server 3.0. The disadvantage is that it is difficult to move users from one suborganization to another. To see if the roles have been migrated to suborganizations, view the hierarchy in the admin console.
`-i` *ExportDir*	Specifies the input directory for the conversion. The input directory is the directory which contains the exported data created by the export tool. The conversion tool searches for export data in `/tmp/psExport` unless you specify a different input directory using the `-i` *ExportDir* option. If you specify a directory which does not contain data exported with the export tool, the conversion tool notifies you that the directory does not have export data and prompts you to enter a valid export directory.
`-o` *ImportDir*	Specifies the output directory for the conversion tool. The output directory is the directory used by the import tool. If you choose an import directory that already contains converted data, the conversion tool notifies you that modifying an existing migration may render import data unusable and prompts you to delete the import directory.

7. The system displays the following messages:

   ```
   Found Portal Server version 6.2
   ```

   ```
   Enter Identity Server Internal LDAP Authentication User password:
   ```

   Type in a valid password.

   ```
   Which directory should be created to store the converted data?
   [/tmp/psImport]
   ```

   You see this prompt only if you do not specify the output directory using the `-o` *ImportDir* option. You can choose the default directory or type another import directory.

   ```
   Import directory /tmp/psImport already exists.
   ```

   ```
   Modifying an existing migration may render import data unusable.
   ```

   ```
   Delete the directory /tmp/psImport?
   ```

   You see the message about an existing import directory only if the import directory already contains converted data. If you have already converted LDAP data, you should delete the directory to avoid conflicts.

8. After you choose the output directory, or if you specified the output directory using the `-o` *ImportDir* option on the command line, you see output similar to the following:

```
Begin conversion process at Wed Jul 10 13:23:14 PDT 2003
Error file: /tmp/psImport/logs/error.16125
Report file: /tmp/psImport/logs/report.16125
Metric file: /tmp/psImport/logs/convert_metrics.16125


Conversion Menu
1) LDAP Database
2) Gateway Rules to Rewriter Rules
3) Desktop
4) Certificate Databases
5) All of the above
6) Exit
Select one of the listed options to convert:
```

You see the conversion menu only if you do not specify the `-a` option.

9. Type **1** and press Enter to convert only the LDAP database.

   The system displays messages similar to the following:

   ```
   Warning - Template data not found in /tmp/psExport

   *****

   Begin LDAP conversion process at Wed Jul 10 13:27:51 PDT 2003

   *** Creating display profile ***

   Warning - No templates tar found, some locale information will not
   be included in the display profile

   Converting provider and channel display profile.

   Warning: Cannot open locale directory, skipping locale printing for
   iwtAppProvider

   Warning: Cannot open locale directory, skipping locale printing for
   iwtBookmarkProvider

   Warning: Cannot open locale directory, skipping locale printing for
   iwtIPInfoProvider

   Warning: Cannot open locale directory, skipping locale printing for
   iwtLoginProvider

   Warning: Cannot open locale directory, skipping locale printing for
   iwtMailCheckProvider

   Warning: Cannot open locale directory, skipping locale printing for
   iwtNetletProvider

   Warning: Cannot open locale directory, skipping locale printing for
   iwtPostitProvider

   Warning: Cannot open locale directory, skipping locale printing for
   iwtSampleJSP

   Warning: Cannot open locale directory, skipping locale printing for
   iwtSampleRss

   Warning: Cannot open locale directory, skipping locale printing for
   iwtTabProvider

   Warning: Cannot open locale directory, skipping locale printing for
   iwtUserInfoProvider

   Converting domain desktop preferences.
   ```

```
Converting role desktop preferences.

Converting user desktop preferences.


*** Converting LDAP data ***

Converting domain xml data

         siroe.com

Converting role xml data

         siroe.com/defaultRole

         siroe.com/AdminRole

Converting domain/role level attributes

Converting user ldif data

         processed 5 entries

Converting ldap user ldif data

Converting component xml data

         iwtGateway

Converting Auth Component xml data

End LDAP conversion process at Wed Jul 10 13:28:11 PDT 2003

Successful completion of conversion process at Wed Jul 10 13:28:11
PDT 2003
```

You see the warnings about the templates not being found because only LDAP data was exported. If you export Desktop data in addition to LDAP data or if you choose to export all data using `exportps -a`, you will not see these warnings.

If the locale data is unavailable, locale customizations to channel title and descriptions will not be migrated. If you have installed language packs it is highly recommended that you use a full export to ensure your locale information will not be lost in the migration.

10. Change to the import directory making sure to substitute the import directory that you selected in Step 7. For example:

    cd *ImportDir*

    Substitute the import directory for *ImportDir*. If you used the default import directory, for example, use `/tmp/psImport`.

11. Type **`ls`** to see the directories created by the conversion tool when converting LDAP data.

    Table 2-4 shows the directories created by the conversion tool when converting LDAP data. This two-column table lists the directories in the first column and their descriptions in the second column.

Table 2-4 Directories Created by the Conversion Tool When Converting LDAP Data

Directory	Description
ImportDir`/dp`	Contains display profile data to be imported
ImportDir`/ldif`	Contains converted LDIF files to be imported
ImportDir`/logs`	Contains `error.PID`, `convert_metrics.PID`, and `report.PID` files (where `PID` is the process ID)
ImportDir`/xml`	Contains converted XML data to be imported

After the conversion has completed, or if you run into problems, check the *ImportDir*`/logs` directory for the `report.PID`, `error.PID`, and `convert_metrics.PID` files (where `PID` is the process ID). The `report.PID` file contains actions the conversion tool has or has not taken and also alerts you to any customizations which you need to convert manually. The `error.PID` file contains warnings or errors so that you can correct any problems. The `convert_metrics.PID` file contains metrics detailing when various elements of the conversion tool started, stopped, and the total time it took to convert the data. You can look at the `stdout` header to know which log and report files to examine for the `convertps` command you are running.

12. If you run the `convertps` tool again in order to convert data other than LDAP data, the tool will prompt you to delete the import directory. Type **`no`** if you wish to keep the LDAP data you have already converted.

Using the Import Tool

The import tool, `importps`, enables you to import the data exported and converted using the export and conversion tools to an Sun ONE Portal Server 6.2 system. The import tool searches for a directory containing valid data created by the conversion tool.

To Run the Import Tool

1. On the Sun ONE Portal Server 6.2 system, change to the import tool directory. For example:

   `cd` *BaseDir*/**`SUNWps/migration/bin`**

2. Run the `importps` command:

   `./importps [-a] [-k] [-m] [`*ImportDir***`]`**

   Table 2-5 describes the options available for importing LDAP data. This two-column table lists the options in the first column and their descriptions in the second column.

Table 2-5 Options Available for Importing LDAP Data

Option	Description
`-a`	Runs all modules without a menu. To choose to import only the LDAP database from the import menu, do not use the `-a` option.
`-k`	Specifies that the tool does not overwrite existing users. When you run `importps` without the `-k` option, if it encounters a conflict with an existing user, it will delete the existing user and import the new user entry defined in *ImportDir*`/ldap/user.ldif`. If you run `importps -k`, if it encounters a conflict with an existing user, it leaves the existing user and places the rejected user entry in *ImportDir*`/ldap/rejected_users.ldif`. If the user has a display profile document in the *ImportDir*`/dp/user/` directory, the display profile file may change. The `-k` option does not affect importing user display profile documents. The `-k` option only affects whether an existing LDAP user's LDIF record is updated. The updates to the LDIF record include everything except display profile updates.

Table 2-5 Options Available for Importing LDAP Data *(Continued)*

Option	Description
`-m`	Merges display profile documents. When you run `importps` without the `-m` option, it overwrites any existing display profile documents it finds. This means that the root suffix, organization, suborganization, and user level display profile documents, if they exist, are overwritten. If you run `importps -m`, when the tool encounters a conflict with an existing display profile document it will merge it.
ImportDir	Specifies the directory where the converted data is located. If the directory does not exist or if the directory does not contain valid converted data, the import tool prompts you to enter a directory containing the converted data to be imported. If you do not specify an import directory, the import tool will search `/tmp/psImport`.

3. The system displays a message similar to the following:

   ```
   Found Portal Server version 6.2
   ```

   ```
   Enter Identity Server Internal LDAP Authentication User password:
   ```

   Type in a valid password.

   ```
   Enter Appserver Administrator password.
   ```

   You see this prompt only if you are migrating onto a Sun ONE Application Server web container.

   ```
   Begin import process at Fri Jul 12 10:04:43 PDT 2003

   Error file: /tmp/psImport/logs/importerror.17206

   Report file: /tmp/psImport/logs/importreport.17206

   Metric file: /tmp/psImport/logs/import_metrics.17206


   Import Menu:

   1) LDAP Database

   2) Rewriter Rules

   3) Desktop

   4) Certificate Databases

   5) All of the above

   6) Exit

   Select one of the listed options to import:
   ```

   You see the import menu only if you do not specify the `-a` option.

4. Type **1** and press Enter to import only the LDAP database.

   The system displays messages similar to the following:

```
Importing Organization xml
         siroe.com
Success 0: Successfully completed.
Importing Organization Updates xml
Success 0: Successfully completed.
Success 0: Successfully completed.
Success 0: Successfully completed.
Importing Role xml
         siroe.com/defaultRole
Success 0: Successfully completed.
         siroe.com/AdminRole
Success 0: Successfully completed.

Importing Component xml
         iwtGateway
Success 0: Successfully completed.
Importing user ldif into ldap database
adding new entry uid=root,ou=People,o=AdminRole,o=siroe.com,o=isp

adding new entry
uid=authentication,ou=People,o=AdminRole,o=siroe.com,o=isp

adding new entry
uid=anonymous,ou=People,o=defaultRole,o=siroe.com,o=isp

adding new entry
uid=user1,ou=People,o=defaultRole,o=siroe.com,o=isp
```

```
adding new entry
uid=gateway-default,ou=People,o=defaultRole,o=siroe.com,o=isp


Any existing users will not be added.
/tmp/psImport/ldif/rejected_users.ldif

has all of the existing users.  Please manually use

/usr/ldap/shared/bin/ldapmodify to modify the existing users
attributes
```

You see this information about existing users not being added only if you use the `-k` option.

```
Importing user ldif modifications into ldap database

Moving PS 6.2 defaultOrg anonymous user to installAnonymous

adding new entry uid=installAnonymous,ou=People,o=siroe.com,o=isp

deleting entry uid=anonymous,ou=People,o=siroe.com,o=isp

Importing dp xml

SUCCESS!

Redeploying portal web application

Deploying to instance host1.siroe.com...

Successful completion of import process at Fri Jul 12 10:20:04 PDT
2003
```

After the import has completed, or if you run into problems, check the *ImportDir*`/logs` directory for the `importreport.PID`, `importerror.PID`, and `import_metrics.PID` files (where `PID` is the process ID). The `importreport.PID` file contains actions the import tool has or has not taken and also alerts you to any customizations which you need to import manually. The `importerror.PID` file contains warnings or errors so that you can correct any problems. The `import_metrics.PID` file contains metrics detailing when various elements of the import tool started, stopped, and the total time it took to import the data. You can look at the `stdout` header to know which log and report files to examine for the `importps` command you are running.

If the role migration type is role to suborganization:

- The *ImportDir*`/psImport/dp` directory structure will have the directory `suborg`.

- Role `dn` form is `o=`*suborg*`,o=`*defaultorg*`,o=`*rootsuffix | dc=defaultorg, dc=rootsuffix*, for example,

  `o=support, o=eng,o=siroe.com,o=isp`

  or

  `o=support, o=eng,dc=siroe, dc=com`

  depending on the naming attribute used to define the Sun ONE Identity Server root suffix in the directory.

- User `dn` form is `uid=`*user*`,ou=People,o=`*suborg*`,o=`*defaultorg*`,o=`*rootsuffix | dc=defaultorg, dc=rootsuffix*, for example:

  `uid=bob,ou=People,o=eng,o=siroe.com,o=isp`

  or

  `uid=bob,ou=People,o=eng,dc=siroe, dc=com`

If the role migration type is role to role:

- The *ImportDir*`/psImport/dp` directory structure will have the directory `role`.
- Role `dn` form is `cn=`*role*`,o=`*defaultorg*`,o=`*rootsuffix | dc=defaultorg, dc=rootsuffix*, for example:

  `cn=eng,o=siroe.com,o=isp`

  or

  `cn=eng,dc=siroe, dc=com`

- User `dn` form `uid=`*user*`,ou=People,o=`*defaultorg*`,o=`*rootsuffix | dc=defaultorg, dc=rootsuffix*, for example:

  `uid=bob,ou=People,o=siroe.com,o=isp`

  or

  `uid=bob,ou=People,dc=siroe, dc=com`

The Sun ONE Portal Server 6.2 installation creates an anonymous user at the default organization level, where the default organization is the Sun ONE Portal Server 6.2 installation organization. The Sun ONE Portal Server 6.2 anonymous user has its own customizations, including a display profile document.

For role to suborganization migrations, the migration tools migrate and place the Sun ONE Portal Server 3.0 anonymous users under a suborganization. This means that the Sun ONE Portal Server 6.2 anonymous user will always be used before the Sun ONE Portal Server 3.0 anonymous user. Therefore, during migration the migration tools rename the Sun ONE Portal Server 6.2 anonymous user to `installAnonymous`. This solves the conflict problem and uses the migrated anonymous user for the anonymous Desktop.

For role to role migrations, the migration tools migrate and place the Sun ONE Portal Server 3.0 anonymous user under the organization. The migration tools, therefore, rename the Sun ONE Portal Server 6.2 anonymous user before importing the Sun ONE Portal Server 3.0 anonymous user.

Writing Data Migration Modules

See Appendix B, “Sun ONE Portal Server 3.0 Data Migration Module Author's Guide” for information on writing your own data migration modules.

Setting the Sun ONE Portal Server 6.2 Organization to the Sun ONE Portal Server 3.0 Domain

If you want to change the Sun ONE Portal Server 6.2 default organization to the domain used in your Sun ONE Portal Server 3.0, you follow these four steps:

1. Identify the 3.0 default Domain value.

   If you are unsure of the Sun ONE Portal Server 3.0 default domain, open the *ExportDir*`/ldif/application.ldif file and search for the attribute iwtPlatform-defaultDomain-at`.

2. Using the Sun ONE Identity Server admin console, create an administrative user under the organization you want to have as the default organization. For information on creating users, see the *Sun ONE Portal Server* 6.2 *Administrator’s Guide*.

3. Edit the *BaseDir*`/SUNWam/lib/AMConfig.properties` file and replace the `com.iplanet.am.defaultOrg` value with the new default organization name.

4. Restart the server using `/etc/init.d/amserver start`.

Post-Migration Issues and Tasks

After migrating LDAP data from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2, you may find that you receive an `Authentication Failed` error message and are unable to log in to the admin console. This section describes two possible causes for the problem, the pre-migration and post-migration conditions for each cause, and the solutions to the problem.

Cause 1

The Sun ONE Portal Server 3.0 default domain that corresponds to the Sun ONE Portal Server 6.2 default organization has LDAP authentication customized to authenticate against an external LDAP server. The Sun ONE Portal Server 6.2 `amAdmin` user will be unable to authenticate to the admin console after migration.

Pre-Migration Conditions

- The Sun ONE Portal Server 3.0 default domain name matches the Sun ONE Portal Server 6.2 default organization name. For example, you installed Sun ONE Portal Server 3.0 with the domain name of `siroe.com` and you installed Sun ONE Portal Server 6.2 with the organization name of `siroe.com`.
- The Sun ONE Portal Server 6.2 admin super user, `amAdmin`, is a user in the default organization. By default, the Sun ONE Portal Server 6.2 `amAdmin` user uses internal LDAP authentication.
- Sun ONE Portal Server 3.0 has been configured at the default domain level to have LDAP Authentication use an External LDAP server for user authentication.

Post-Migration Condition

- The Sun ONE Portal Server 6.2 admin console authentication type is LDAP.

Solution

The `amAdmin` user cannot authenticate because it is trying to authenticate against an External LDAP server and the `amAdmin` UID does not exist in the External LDAP server. To resolve the problem, follow these steps:

1. Authenticate with the `amAdmin` user's full `dn`. For example:

   `login: uid=amAdmin,ou=People,o=`*defaultorg*`,o=`*rootsuffix | dc=defaultorg, dc=rootsuffix*
   `passwd:` *passwd*

   where

   - *default organization* = response to the Sun ONE Portal Server 6.2 installation question:

     ```
     What is the organization name? [host1.siroe.com] siroe.com
     ```

   - *root suffix* = response to the Sun ONE Portal Server 6.2 installation question:

     ```
     What is the root suffix of the directory tree? [o=isp]
     ```

   - *passwd* = response to the Sun ONE Portal Server 6.2 installation question:

     ```
     What is the passphrase for this server? Again?
     ```

   With the answers to the installation questions, the login might look like:

   ```
   login: uid=amAdmin,ou=People,o=siroe.com,o=isp passwd:
   adminpassword
   ```

   or if the domain container naming attributes were used

   ```
   login: uid=amAdmin,ou=People,dc=siroe,dc=com passwd: adminpassword
   ```

2. Configure the admin super user. Note that this step is customer dependent.

   The easiest approach would be to add the `amAdmin` user userid to the external LDAP server.

Another approach would be to create an admin user under a suborganization of the default organization. For example, you could create a suborganization under the default organization for the super admin user:

1. Select the organization.
2. Select `New` to create a new suborganization.
3. Enter a name. For example, enter `admin`.
4. Select `Create`.
5. Select the suborganization. For example, select admin.
6. Select `Services` from the choice menu.
7. Select `Register`.

8. Select `Core`, `LDAP`, and `Register` in same view.
9. Select `Core` properties and then select `Create`.
10. Select `LDAP properties` and then select `Create`.
11. Select `Users` from the choice menu.
12. Select `New`.
13. Enter the following information, as you prefer. For example:
    - User Id = `sadmin`
    - password = `sadmin`
14. Select `Create`.
15. Select `root suffix` from `Location` in top frame, for example 'isp'.
16. Select `Roles` from the choice menu.
17. Select `Top-level Admin Role`.
18. Select `Add`.
19. Enter `sadmin` in User Id field.
20. Select `Search`.
21. Select `sadmin` checkbox.
22. Select `Submit`.
23. Select `Logout`.
24. Access the admin console with `http://`*host*:*port*`/amconsole/login?org=admin`

    The admin user will only authenticate with this URL.
25. Enter `sadmin/sadmin`.

Cause 2

The Sun ONE Portal Server 3.0 default domain that corresponds to the Sun ONE Portal Server 6.2 default organization has the authentication admin authenticator customized. The Sun ONE Portal Server 6.2 `amAdmin` user may be unable to authenticate to the admin console after migration.

Pre-Migration Conditions

- The Sun ONE Portal Server 3.0 default domain name matches the Sun ONE Portal Server 6.2 default organization name. For example, Sun ONE Portal Server 3.0 is installed with `siroe.com` as the default domain name and Sun ONE Portal Server 6.2 is installed with `siroe.com` as the default organization name.
- The Sun ONE Portal Server 6.2 admin super user (`amAdmin`) is a user in the default organization and uses internal LDAP authentication.

Post-Migration Condition

The Sun ONE Portal Server 6.2 admin console authentication type is not LDAP.

Solution

When trying to authenticate to the admin console, if the authenticator type is not LDAP, then you can set it to LDAP by using the `amadmin` utility:

```
BaseDir/SUNWam/bin/amadmin -u
"uid=amAdmin,ou=People,o=defaultorg,o=rootsuffix | dc=defaultorg, dc=rootsuffix" -w
"adminpassphrase" -t changeAdminAuth.xml
```

For example, if the base directory is `/opt`, the domain is `siroe.com`, the organization naming attribute is `o=`, and the admin passphrase is `passphrase`, the command would be:

```
/opt/SUNWam/bin/amadmin -u "uid=amAdmin,ou=People,o=siroe.com,o=isp" -w
"passphrase" -t changeAdminAuth.xml
```

where `changeAdminAuth.xml` is:

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

<!DOCTYPE Requests PUBLIC "-//Sun ONE//iDSAME 5.0 Admin CLI DTD//EN "
"file:/opt/SUNWam/dtd/amAdmin.dtd">

<Requests>

<OrganizationRequests DN="o=siroe.com,o=isp">

<ModifyServiceTemplate serviceName="iPlanetAMAuthService"
schemaType="Organization">

<AttributeValuePair>

<Attribute name="iplanet-am-auth-admin-auth-module"/>

<Value>LDAP</Value>
```

```
</AttributeValuePair>

</ModifyServiceTemplate>

</OrganizationRequests>

</Requests>
```

In this example, replace `siroe.com` with your organization and `isp` with your root suffix.

This will set `amAdmin`'s admin authentication module to LDAP. You will need to restart the server by issuing the following command for the change to take effect:

```
/etc/init.d/amserver start
```

Please refer to “Cause 1” on page 60 if you still have problems authenticating.

Chapter 3

Migrating Custom Authentication Modules

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

Custom authentication modules from Sun ONE Portal Server 3.0 must be simulated in Sun™ ONE Identity Server as plugged-in authentication modules. All custom authentication modules must implement the Authentication Service Provider Interface (SPI) which provides the authentication framework, and should be written using `com.sun.identity.authentication.spi.AMLoginModule` class. In addition, to implementing the`AMLoginModule` SPI, there are a number of configuration files that must also be created. This chapter describes the interfaces and classes required to migrate custom authentication modules from a Sun ONE Portal Server 3.0 and the high-level steps necessary to implement them in the Sun ONE Identity Server. This chapter contains the following sections:

- Implementing the Authentication Service Provider Interface (SPI)
- Creating the AuthModule.jar File
- Creating the XML Service File
- Creating the Authentication Module Configuration File
- Creating the Localization Properties File
- Modifying the amAuth.xml File

Implementing the Authentication Service Provider Interface (SPI)

The Sun ONE Identity Server 6.1 authentication framework provides the Authentication Service Provider Interface (SPI) to invoke specific authentication modules. The Authentication SPI implements the Java™ Authentication and Authorization Service (JAAS) LoginModule API, and provides methods to access the Authentication Service and module configuration properties. The Authentication SPI is defined in the `com.sun.identity.authentication.spi` package. The Authentication SPI contains the abstract class, `AMLoginModule,` used for writing pluggable authentication modules. To write an authentication module, you extend the `AMLoginModule` class. Because `AMLoginModule` is an abstract class, it must be subclassed with the name of the authentication module. The class must also implement the `init, process`, and `getPrincipal` abstract methods in order to communicate with the module configuration properties files.

See the Sun ONE Identity Server Javadocs for details on implementing the Authentication SPI and the `AMLoginModule` class. Refer to Appendix C, "Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2" for details on mapping APIs.

Creating the AuthModule.jar File

Once the custom authentication module is written and compiled, create a .jar file for the module and put it in the *BaseDir*`/SUNWam/web-apps/services/lib` directory.

To create the `AuthModule.jar` file, issue the command:

`jar -cvf` *AuthModulename*`.`**`jar *.class`**

where the `*.class` file corresponds to the name of the authentication module you created that extended the `com.sun.identity.authentication.spi.AMLoginModule` class.

Creating the XML Service File

Each authentication service must have its own service XML file. This file specifies the user and administrator configurable attributes and allows them to be managed using the Sun ONE Identity Server administration console. The name of the XML service file typically follows the format `amAuth`*ModuleName*`.xml` (for example, `amAuthSafeWord.xml` or `amAuthLDAP.xml`) and is located in the *BaseDir*`/SUNWam/config/xml` directory.

Creating the Authentication Module Configuration File

The authentication module configuration file specifies the authentication module credentials required by an identity (either user, service or application) in order to authenticate to the module. Typically, the settings in this file define the user authentication screens. The name of the authentication module configuration file typically follows the format *ModuleName*`.xml` (for example, `SafeWord.xml` or `LDAP.xml`) and is located in the *BaseDir*`/SUNWam/web-apps/services/config/auth/default` directory.

In Sun ONE Identity Server 5.1 implementation, this file was a `.properties` file (for example, `LDAP.properties`). In Sun ONE Identity Server 6.1 implementation, this file is now a `.xml` file (for example, `LDAP.xml`). After converting LDAP data, the conversion tool will have created an *ImportDir*`/xml/component/iwtAuth`*Module_Name*`.xml` file. The import tool does not import the `iwtAuth`*Module_Name*`.xml.` Because the authentication module must be rewritten to the Authentication SPI, it is recommended that you create a new *ModuleName*`.xml` file for the custom authentication module although the `iwtAuth`*Module_Name*`.xml` can be useful to look through as you develop the *ModuleName*`.xml` file.

Creating the Localization Properties File

The localization properties file defines the language-specific screen text for the attribute names for the module. The name of the localization properties file typically follows the format `amAuth`*ModuleName*`.properties` (for example, `amAuthSafeWord.properties` or `amAuthLDAP.properties`) and is located in the *BaseDir*`/SUNWam/locale` directory under sub-directory appropriate to the locale (for example, `en_US`, or `FR`).

Modifying the amAuth.xml File

The `amAuth.xml` file defines the “parent” core authentication service. This file is located in the *BaseDir*`/SUNWam/config/xml` directory. Whenever a new authentication service is created, attributes in this file must be modified in order to recognize an authentication module.

1. Make a backup copy of the `amAuth.xml` file.
2. Add a value defining in the custom authentication module to the `iplanet-am-auth-authenticators` attribute list in the `amAuth.xml` file. The value contains a text string specifying the full class name of the module, including package. For example,

   ```
   <Value>com.sun.identity.authentication.modules.Modulename.MODULENAME
   </Value>
   ```

3. Add a value defining the custom authentication module to the `iplanet-am-auth-allowed-modules` attribute in the `amAuth.xml` file. The value contains a text string specifying the name of the module to display in Admin console for the module and the name defined for the module in the `iplanet-am-auth-authenticators` attribute list. For example,

   ```
   <ChoiceValue i18nKey="Module">MODULENAME</ChoiceValue>
   ```

4. Save the updated `amAuth.xml` file.
5. Type the following command to delete the old core service file.

   ```
   BaseDir/SUNWam/bin/amadmin --runAsDN
   uid=amadmin,ou=People,default_org,root_suffix --password password
   --deleteService iplanetAMAuthService
   ```

6. Type the following command to load the updated core service file.

   ```
   BaseDir/SUNWam/bin/amadmin --runAsDN
   uid=amadmin,ou=People,default_org,root_suffix --password password --schema
   amAuth.xml
   ```

Chapter 4

Migrating The Rewriter

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

Migration of the Rewriter involves converting gateway attributes to rules using the export, conversion, and import tools.

This chapter contains the following sections:

- Overview of the Rewriter
- Converting Gateway Attributes to Rules
- Using the Export Tool
- Using the Conversion Tool
- Using the Import Tool
- Writing Data Migration Modules

Overview of the Rewriter

The Rewriter serves two functions. The first function involves gateway-like components which act as a reverse proxy. In this scenario, the Rewriter prepends each URL with the gateway’s URL. For example, in a gateway scenario, an intranet page with the following hyperlink

```
<a href=”http://mymachine.myintranet.com/mypage.html”>
```

needs to be translated to

```
<a
href=”https://gateway.siroe.com/http://mymachine.myintranet.com/mypage.htm
l”>
```

so that when the user clicks a link associated with this anchor, the browser contacts the gateway which then fetches the content of `mypage.html` from `mymachine.myintranet.com`. In this way, the user can browse machines through the gateway that are not normally accessible from the intranet.

This chapter discusses the migration of this functionality of the Rewriter.

The second function of the Rewriter is to convert all relative URLs to absolute URLs. Migration of this functionality is not necessary.

Converting Gateway Attributes to Rules

To migrate iPlanet™ Portal Server 3.0 (Service Pack 3a, 4 or 5) gateway attributes to Sun™ ONE Portal Server 6.2 rules, you need to run the export, conversion, and import tools. The migration tools migrate the gateway attributes to rules in the Sun ONE Portal Server 6.2 Rewriter service ruleset called `idsruleset`. Importing the `idsruleset` does not remove the existing `default_ruleset`.

After the import tool has successfully imported `idsruleset`, the tool updates the Sun ONE Portal Server 6.2 display profiles for the URLScraper and XML providers:

```
<Provider name=”URLScraperProvider”
class=”com.sun.portal.providers.urlscraper.URLScraperProvider”>
```

and

```
<Provider name=”XMLProvider”
class=”com.sun.portal.providers.xml.XMLProvider”>
```

to include:

```
<Properties>
  <String name=”urlScraperRulesetID” value=”idsruleset”/>
</Properties>
```

The updates to the display profile only occur for provider definitions defined at the root suffix. For example, the default installation root suffix is `o=isp`.

Using the Export Tool

The export tool, `exportps`, stores data from the Sun ONE Portal Server 3.0 system to an export directory. To migrate gateway attributes to rules, you need to begin by running the export tool to gather the LDAP data and store it to disk.

To Run the Export Tool

1. On the Sun ONE Portal Server 3.0 system, change to the migration tool directory:

   `cd` *BaseDir*<b>`/SUNWps/migration/bin/`</b>

2. Run the `exportps` command:

   `./exportps [-a] [-p` *LDAP passphrase*<b>`] [`</b>*ExportDir*<b>`]`</b>

   Table 4-1 describes the options available for exporting LDAP data. This two-column table lists the options in the first column and their descriptions in the second column.

Table 4-1 Options Available for Exporting LDAP Data

Option	Description
`-a`	Runs all modules without a menu. To choose to export only LDAP data from the export menu, do not use the `-a` option.
`-p` *LDAP passphrase*	Sets the LDAP passphrase in order to avoid the LDAP passphrase prompt. If you do not use the `-p` *LDAP passphrase* option with the `-a` option, the tool will prompt you for the LDAP passphrase.
ExportDir	Specifies the directory where the information is to be saved. If you do not use the *ExportDir* option, the tool prompts you for the export directory.

3. The system displays a message similar to the following:

   ```
   Which directory should be created to store the Portal Server system?
   [/tmp/psExport]
   ```

   You see this prompt only if you do not specify the output directory on the command line. You can choose the default directory `/tmp/psExport` or enter the export directory.

   Type the directory and press Enter.

4. The system displays a message similar to the following:

   ```
   Delete the directory /tmp/psExport?
   ```

   You see this message only if there is already exported data in the `ExportDir` directory. If you have not already exported the LDAP database, you may keep the export directory without risking any conflicts in the data. If you have already exported the LDAP database, you should delete the directory to avoid conflicts.

5. The system displays a message similar to the following:

   ```
   Found iPS version 3.0sp5

   Begin export process at  Tue Jul  9 10:10:04 PDT 2003

   Error file: /tmp/psExport/logs/error.7644

   Report file: /tmp/psExport/logs/report.7644

   Metrics file: /tmp/psExport/logs/export_metrics.7644


   Export Menu:

   1) LDAP Database

   2) Desktop

   3) Certificate Databases

   4) All of the above

   5) Exit

   Select one of the listed options to export:
   ```

   You see the export menu only if you do not specify the `-a` option.

6. Type **`1`** and press Enter to export LDAP data only.

7. The system displays the following message:

   ```
   Enter the LDAP admin passphrase :
   ```

   You see this prompt only if you select a full export using the `-a` option and you do not specify the `-p` *LDAP passphrase* option.

   Type the LDAP admin passphrase and press Enter.

   You see output similar to the following:

   ```
   Dumping the ldap database
     organizationalPerson
     user
     role
     domain
     application
   Dumping xml
   .....................


   Successful completion of export process at  Tue Jul  9 10:21:25 PDT 
   2003
   ```

8. Change to the export directory you specified in Step 2 or Step 3. For example:

   `cd` *ExportDir*

   Substitute the export directory for *ExportDir*. If you used the default export directory, for example, use `/tmp/psExport`.

   Type **`ls`** to see the directories created by the export tool.

   Table 4-2 shows the four directories created by the export tool. This two-column table lists the directories in the first column and their descriptions in the second column.

Table 4-2 Directories Created by the Export Tool When Exporting LDAP Data

Directory	Description
ExportDir/`config`	Contains exported configuration data
ExportDir/`ldif`	Contains exported LDIF files
ExportDir/`logs`	Contains `error.PID`, `export_metrics.PID`, and `report.PID` files (where `PID` is the process ID)
ExportDir/`xml`	Contains exported XML data

After the export has completed, or if you run into problems, check the *ExportDir*/`logs` directory for the `report.PID`, `error.PID`, and `export_metrics.PID` files (where `PID` is the process ID). The `report.PID` file contains actions the export tool has or has not taken. The `error.PID` file contains warnings or errors so that you can correct any problems. The `export_metrics.PID` file contains metrics detailing when various elements of the export tool started, stopped, and the total time it took to export the data. You can look at the `stdout` header to know which log and report files to examine for the `exportps` command you are running.

9. Examine the exported data and manually change, before converting the data, all instances of the name of the Sun ONE Portal Server 3.0 system to the name of the Sun ONE Portal Server 6.2 system. If the port numbers on the two systems are different, you must change these manually as well.

If you are performing a single-system migration, server names will be the same, but port numbers will be different. You need to change port numbers manually before converting the data.

The migration tools do not modify server names and port numbers.

For example, in the following extract of the exported data in the *ExportDir*/config/platform.config file, you need to change manually all instances of host1.siroe.com, which is the name of the Sun ONE Portal Server 3.0 system, to the name of the Sun ONE Portal Server 6.2 system if the systems use different names. If the systems use different port numbers, you need to change them as well.

```
ips.defaultDomain=siroe.com

ips.server.protocol=http

ips.server.host=host1.siroe.com

ips.server.port=8080

ips.profile.host=host1.siroe.com

ips.profile.port=8080

ips.gateway.protocol=https

ips.gateway.host=host1.siroe.com

ips.gateway.port=443

ips.gateway.trust_all_server_certs=false

ips.virtualhost=host1.siroe.com 192.18.66.42

ips.naming.url=http://host1.siroe.com:8080/namingservice

ips.notification.url=http://host1.siroe.com:8080/
notificationservice
```

Using the Conversion Tool

The conversion tool, convertps, enables you to select which data to convert from the data exported from a valid installation of Sun ONE Portal Server 3.0. Because the conversion tool runs on the Sun ONE Portal Server 6.2 system, you need to move the export directory from the Sun ONE Portal Server 3.0 system to the Sun ONE Portal Server 6.2 system.

To Run the Conversion Tool

1. On the Sun ONE Portal Server 3.0 system, change to the `export` directory containing the data output by the export tool. For example:

   ```
   cd ExportDir
   ```

   If you used the default directory for exporting the LDAP data, you would type:

   ```
   cd /tmp
   ```

2. Save the export directory using the `tar` command. For example:

   ```
   tar cvf export.tar psExport
   ```

3. Use an FTP program to transfer the `export.tar` file to the Sun ONE Portal Server 6.2 system.

4. Extract the files from `export.tar` using the `tar` command. For example:

   ```
   tar xvf export.tar
   ```

5. On the Sun ONE Portal Server 6.2 system, change to the migration tool directory:

   ```
   cd BaseDir/SUNWps/migration/bin/
   ```

6. Run the `convertps` command:

   ```
   ./convertps [-a] [-f] [-i ExportDir] [-o ImportDir]
   ```

   Table 4-3 describes the options available for the `convertps` command. This two-column table lists the options in the first column and their descriptions in the second column.

Table 4-3 Options Available for Converting LDAP Data

Option	Description
`-a`	Runs all modules without a menu. To choose to convert only gateway rules to Rewriter rules from the conversion menu, do not use the `-a` option.

Table 4-3 Options Available for Converting LDAP Data *(Continued)*

Option	Description
`-f`	Converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server roles and places users under the organization. This option retains the multiple role to user assignment similar to Sun ONE Portal Server 3.0, but does not retain the hierarchical role functionality. All roles are created under the organization and are not prioritized. The display profile documents are prioritized and merged accordingly. To see if the roles have been migrated to roles, view the roles under the organization in the admin console. If you do not use the `-f` option, the conversion tool converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server suborganizations and places users under the suborganization. Without the `-f` option, the conversion tool retains the hierarchical functionality and customizations from Sun ONE Portal Server 3.0. The disadvantage is that it is difficult to move users from one suborganization to another. To see if the roles have been migrated to suborganizations, view the hierarchy in the admin console.
`-i` *ExportDir*	Specifies the input directory for the conversion. The input directory is the directory which contains the exported data created by the export tool. The conversion tool searches for export data in `/tmp/psExport` unless you specify a different input directory using the `-i` *ExportDir* option. If you specify a directory which does not contain data exported with the export tool, the conversion tool notifies you that the directory does not have export data and prompts you to enter a valid export directory.
`-o` *ImportDir*	Specifies the output directory for the conversion tool. The output directory is the directory used by the import tool. If you choose an import directory which already contains converted data, the conversion tool notifies you that modifying an existing migration may render import data unusable and prompts you to delete the import directory.

7. The system displays the following messages:

   ```
   Found Portal Server version 6.2
   ```

   ```
   Enter Identity Server Internal LDAP Authentication User password:
   ```

   Type in a valid password.

   ```
   Which directory should be created to store the converted data?
   [/tmp/psImport]
   ```

   You see this prompt only if you do not specify the output directory using the `-o` *ImportDir* option. You can choose the default directory or enter another import directory.

   ```
   Import directory /tmp/psImport already exists.
   ```

   ```
   Modifying an existing migration may render import data unusable.
   ```

   ```
   Delete the directory /tmp/psImport?
   ```

   You see the message about an existing import directory only if the import directory already contains converted data. In this example, option 2 is chosen to convert Gateway rules to Rewriter rules. In Chapter 2, only LDAP data was migrated. Since the migration of gateway rules to Rewriter rules has not yet been performed, the import directory need not be deleted.

8. After you choose the output directory, or if you specified the output directory using the `-o` *ImportDir* option on the command line, you see output similar to the following:

```
Begin conversion process at Mon Jul 15 10:42:46 PDT 2003
Error file: /tmp/psImport/logs/error.18106
Report file: /tmp/psImport/logs/report.18106
Metric file: /tmp/psImport/logs/convert_metrics.18106

Conversion Menu
1) LDAP Database
2) Gateway Rules to Rewriter Rules
3) Desktop
4) Certificate Databases
5) All of the above
6) Exit
Select one of the listed options to convert:
```

You see the conversion menu only if you do not specify the `-a` option.

9. Type **2** and press Enter to convert only the gateway rules to Rewriter rules.

   After you select an option, you see output that includes the option you selected, the end of the Gateway Rules to Rewriter Rules migration process, and the successful completion of the conversion process similar to the following:

   ```
   Warning - Template data not found in /tmp/psExport


   *****

   Begin gateway rule conversion process at Mon Jul 15 11:24:52 PDT
   2003

     Converting gateway rules data

   End gateway rule conversion process at Mon Jul 15 11:24:54 PDT 2003

   Successful completion of conversion process at Mon Jul 15 11:24:55
   PDT 2003
   ```

   You see the warnings about the templates not being found because only LDAP data was exported. If you export Desktop data in addition to LDAP data or if you choose to export all data using `exportps -a`, you will not see these warnings.

10. Change to the import directory making sure to substitute the import directory that you selected in Step 7. For example:

    cd *ImportDir*

    Substitute the import directory for *ImportDir*. If you used the default import directory, for example, use `/tmp/psImport`.

11. Type **ls** to see the directories created by the conversion tool.

    Table 4-4 shows the two directories created by the conversion tool when converting gateway rules to Rewriter rules. This two-column table lists the directories in the first column and their descriptions in the second column.

Table 4-4 Directories Created by the Conversion Tool When Converting Gateway Rules to Rewriter Rules

Directory	Description
ImportDir/rewriter/rules	Contains the rules.xml file which is the new ruleset definition.
ImportDir/logs	Contains error.PID, convert_metrics.PID, and report.PID files (where PID is the process ID).

After the conversion has completed, or if you run into problems, check the *ImportDir*/logs directory for the report.PID, error.PID, and convert_metrics.PID files (where PID is the process ID). The report.PID file contains actions the conversion tool has or has not taken and also alerts you to any customizations which you need to convert manually. The error.PID file contains warnings or errors so that you can correct any problems. The convert_metrics.PID file contains metrics detailing when various elements of the conversion tool started, stopped, and the total time it took to convert the data. You can look at the stdout header to know which log and report files to examine for the convertps command you are running.

12. If you run the convertps tool again in order to convert data other than Gateway rules to Rewriter rules, the tool will prompt you to delete the import directory. Type no if you wish to keep the Rewriter rules you have already converted.

Using the Import Tool

The import tool, importps, enables you to import the data exported and converted using the export and conversion tools to a Sun ONE Portal Server 6.2 system. The import tool searches for a directory containing valid data created by the conversion tool.

To Run the Import Tool

1. On the Sun ONE Portal Server 6.2 system, change to the import tool directory. For example:

   cd *BaseDir***/SUNWps/migration/bin**

2. Run the `importps` command:

   `./importps [-a] [-k] [-m] [`*ImportDir*`]`

   Table 4-5 describes the options available for importing gateway rules to Rewriter rules. This two-column table lists the options in the first column and their descriptions in the second column.

Table 4-5 Options Available for Importing Rewriter Rules

Option	Description
`-a`	Runs all modules without a menu. To choose to import only Rewriter rules from the import menu, do not use the `-a` option.
`-k`	Specifies that the tool does not overwrite existing users. When you run `importps` without the `-k` option, if it encounters a conflict with an existing user, it will delete the existing user and import the new user entry defined in *ImportDir*`/ldap/user.ldif`. If you run `importps -k`, if it encounters a conflict with an existing user, it leaves the existing user and places the rejected user entry in *ImportDir*`/ldap/rejected_users.ldif`. If the user has a display profile document in the *ImportDir*`/dp/user/` directory, the display profile file may change. The `-k` option does not affect importing user display profile documents. The `-k` option only affects whether an existing LDAP user's LDIF record is updated. The updates to the LDIF record include everything except display profile updates.
`-m`	Merges display profile documents. When you run `importps` without the `-m` option, it overwrites any existing display profile documents it finds. This means that the root suffix, organization, suborganization, and user level display profile documents, if they exist, are overwritten. If you run `importps -m`, when the tool encounters a conflict with an existing display profile document it will merge it.
ImportDir	Specifies the directory where the converted data is located. If the directory does not exist or if the directory does not contain valid converted data, the import tool prompts you to enter a directory containing the converted data to be imported. If you do not specify an import directory, the import tool will search `/tmp/psImport`.

3. The system displays a message similar to the following:

```
Found Portal Server version 6.2

Enter Identity Server Internal LDAP Authentication User password:
```

Type in a valid password.

```
Enter Appserver Administrator password.
```

You see this prompt only if you are migrating onto a Sun ONE Application Server web container.

```
Begin import process at Mon Jul 15 13:42:42 PDT 2003

Error file: /tmp/psImport/logs/importerror.18302

Report file: /tmp/psImport/logs/importreport.18302

Metrics file: /tmp/psImport/logs/import_metrics.18302


Import Menu:

1) LDAP Database

2) Rewriter Rules

3) Desktop

4) Certificate Databases

5) All of the above

6) Exit

Select one of the listed options to import:
```

You see the import menu only if you do not specify the `-a` option.

4. Type **2** and press Enter to import only Rewriter rules.

   The system displays messages similar to the following:

   ```
   Importing Rewriter Rules

   SUCCESS!

   Updating URLScraperProvider urlScraperRulesetID

   SUCCESS!

   Updating XMLProvider urlScraperRulesetID

   SUCCESS!

   Redeploying portal web application

   Deploying to instance host1.siroe.com...

   Successful completion of import process at Mon Jul 15 13:56:51 PDT
   2003
   ```

   After the import has completed, or if you run into problems, check the *ImportDir*/`logs` directory for the `importreport.PID`, `importerror.PID`, and `import_metrics.PID` files (where `PID` is the process ID). The `importreport.PID` file contains actions the import tool has or has not taken and also alerts you to any customizations which you need to import manually. The `importerror.PID` file contains warnings or errors so that you can correct any problems. The `import_metrics.PID` file contains metrics detailing when various elements of the import tool started, stopped, and the total time it took to import the data. You can look at the `stdout` header to know which log and report files to examine for the `importps` command you are running.

Writing Data Migration Modules

See Appendix B, “Sun ONE Portal Server 3.0 Data Migration Module Author's Guide” for information on writing your own data migration modules.

Chapter 5

Migrating The Desktop

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

This chapter provides procedures to migrate the Sun ONE Portal Server 3.0 Desktop to the Sun™ ONE Portal Server 6.2 Desktop. In particular, migration of the Desktop includes:

- Migration of templates, JSPs, and resource bundles
- Conversion of provider and channel components to services

This chapter also discusses customizations that the migration tools do not migrate automatically and the steps you need to take to migrate these customizations.

This chapter contains the following sections:

Desktop Pre-Migration Issues and Tasks

This section describes manual tasks that you may need to carry out before running the export tool.

Ensuring Files are Migrated

The installation of Sun ONE Portal Server 6.2 creates the *BaseDir*/`SUNWps/migration/modules/desktop/migrate_files.txt` file containing the default location of the notes provider text file. If you have moved files to non-default locations (for example, locations other than the `public_html` and templates directories), or if you have created a provider similar to the notes provider, you must add file system directories or files to the `migrate_files.txt` file before running the export tool in order to have them migrated. Files and resources listed in the `migrate_files.txt` are not translated. They are migrated "as is."

Using the Export Tool

The export tool, `exportps`, stores data from the Sun ONE Portal Server 3.0 system to an export directory. To migrate the Sun ONE Portal Server 3.0 Desktop to Sun ONE Portal Server 6.2, you need to begin by running the export tool to gather the various Desktop data and store it to disk.

To Run the Export Tool

1. On the Sun ONE Portal Server 3.0 system, change to the migration tool directory:

   cd *BaseDir*`/SUNWps/migration/bin/`

2. Run the `exportps` command:

   ./exportps [-a] [-p *LDAP passphrase*] [*ExportDir*]

   Table 5-1 describes the options available for exporting the Sun ONE Portal Server 3.0 Desktop. This two-column table lists the options in the first column and their descriptions in the second column.

Table 5-1 Options Available for Exporting the Sun ONE Portal Server 3.0 Desktop

Option	Description
`-a`	Runs all modules without a menu. To choose to export only the Desktop from the export menu, do not use the `-a` option.
`-p` *LDAP passphrase*	Sets the LDAP passphrase in order to avoid the LDAP passphrase prompt. If you do not use the `-p` *LDAP passphrase* option with the `-a` option, the tool will prompt you for the LDAP passphrase.
ExportDir	Specifies the directory where the information is to be saved. If you do not use the *ExportDir* option, the tool prompts you for the export directory.

3. The system displays a message similar to the following:

   ```
   Which directory should be created to store the Portal Server system?
   [/tmp/psExport]
   ```

   You see this prompt only if you do not specify the output directory on the command line. You can choose the default directory `/tmp/psExport` or enter the export directory.

   Type the directory and press Enter.

4. The system displays a message similar to the following:

   ```
   Delete the directory /tmp/psExport?
   ```

   You see this message only if there is already exported data in the `ExportDir` directory. If you have not already exported the Desktop, you may keep the export directory without risking any conflicts in the data. If you have already exported the Desktop, you should delete the directory to avoid conflicts.

5. The system displays a message similar to the following:

   ```
   Found iPS version 3.0sp5

   Begin export process at  Tue Jul 23 15:28:42 PDT 2003

   Error file: /tmp/psExport/logs/error.20370

   Report file: /tmp/psExport/logs/report.20370

   Metrics file: /tmp/psExport/logs/export_metrics.20370


   Export Menu:

   1) LDAP Database

   2) Desktop

   3) Certificate Databases

   4) All of the above

   5) Exit

   Select one of the listed options to export:
   ```

   You see the export menu only if you do not specify the `-a` option.

6. Type **2** and press Enter to export the Desktop only.

7. The system displays the following message:

   ```
   Enter the LDAP admin passphrase :
   ```

   You see this prompt only if you select a full export using the `-a` option and you do not specify the `-p` *LDAP passphrase* option.

   Type the LDAP admin passphrase and press Enter.

   You see output similar to the following:

   ```
   Copying templates and resource bundles

   .......

   Successful completion of export process at  Tue Jul 23 15:29:41 PDT
   2003
   ```

8. Change to the export directory you specified in Step 2 or Step 3. For example:

   cd *ExportDir*

   Substitute the export directory for *ExportDir*. If you used the default export directory, for example, use `/tmp/psExport`.

   Type **`ls`** to see the directories and files created by the export tool.

   Table 5-2 shows the directories and files created by the export tool. This two-column table lists the directories and files in the first column and their descriptions in the second column.

Table 5-2 Directories and Files Created by the Export Tool When Exporting the Desktop

Directory	Description
ExportDir/`logs`	Contains `error.PID`, `export_metrics.PID`, and `report.PID` files (where `PID` is the process ID)
ExportDir/`templates.tar`	Contains exported Desktop templates, resource bundles, web server root documents, modified and new authentication templates.

   After the export has completed, or if you run into problems, check the *ExportDir*/`logs` directory for the `report.PID`, `error.PID`, and `export_metrics.PID` files (where `PID` is the process ID). The `report.PID` file contains actions the export tool has or has not taken. The `error.PID` file contains warnings or errors so that you can correct any problems. The `export_metrics.PID` file contains metrics detailing when various elements of the export tool started, stopped, and the total time it took to export the data. You can look at the `stdout` header to know which log and report files to examine for the `exportps` command you are running.

9. Examine the exported data and manually change, before converting the data, all instances of the name of the Sun ONE Portal Server 3.0 system to the name of the Sun ONE Portal Server 6.2 system. If the port numbers on the two systems are different, you must change these manually as well.

   If you are performing a single-system migration, server names will be the same, but port numbers will be different. You need to change port numbers manually before converting the data.

   The migration tools do not modify server names and port numbers.

For example, in the following extract of the exported data in the *ExportDir*/`config/platform.config` file, you need to change manually all instances of `host1.siroe.com`, which is the name of the Sun ONE Portal Server 3.0 system, to the name of the Sun ONE Portal Server 6.2 system if the systems use different names. If the systems use different port numbers, you need to change them as well.

```
ips.defaultDomain=siroe.com

ips.server.protocol=http

ips.server.host=host1.siroe.com

ips.server.port=8080

ips.profile.host=host1.siroe.com

ips.profile.port=8080

ips.gateway.protocol=https

ips.gateway.host=host1.siroe.com

ips.gateway.port=443

ips.gateway.trust_all_server_certs=false

ips.virtualhost=host1.siroe.com 192.18.66.42

ips.naming.url=http://host1.siroe.com:8080/namingservice

ips.notification.url=http://host1.siroe.com:8080/
notificationservice
```

10. If you have customized Sun ONE Portal Server 3.0 `iwtLogin` provider templates, you should modify the `display*.html` templates in Sun ONE Portal Server 6.2 before migrating your templates. The migrated `iwtLoginProvider` channel uses the Sun ONE Portal Server 6.2 `Login` templates.

    If you do not modify the `display*.html` templates in Sun ONE Portal Server 6.2 before migrating the templates, you will need to update the `Login` templates with your Sun ONE Portal Server 3.0 customizations.

11. Because the Sun ONE Portal Server 3.0 `iwtLoginProvider` templates are not being migrated, the Sun ONE Portal Server 6.2 templates must be manually edited.

    a. **`cd /etc/opt/SUNWps/desktop/default/Login`**

b. In any of the `display*.html` files, add `https://server_name:port/` to the following:

```
<FORM ACTION=”https://server_name:port/amserver/login” ...>
```

```
<A HREF=”https://server_name:port/amserver/login ...”>
```

Using the Conversion Tool

The conversion tool, `convertps`, enables you to select which data to convert from the data exported from a valid installation of Sun ONE Portal Server 3.0. Because the conversion tool runs on the Sun ONE Portal Server 6.2 system, you need to move the export directory from the Sun ONE Portal Server 3.0 system to the Sun ONE Portal Server 6.2 system.

To Run the Conversion Tool

1. On the Sun ONE Portal Server 3.0 system, change to the directory above the `export` directory containing the data output by the export tool.

   For example, if you used the default directory (`/tmp/psExport`) for exporting the Desktop, you would type:

   ```
   cd /tmp
   ```

2. Save the export directory using the `tar` command. For example:

   ```
   tar cvf export.tar psExport
   ```

3. Use an FTP program to transfer the `export.tar` file to the Sun ONE Portal Server 6.2 system.

4. Extract the files from `export.tar` using the `tar` command. For example:

   ```
   tar xvf export.tar
   ```

5. On the Sun ONE Portal Server 6.2 system, change to the migration tool directory:

   cd *BaseDir***/SUNWps/migration/bin**

6. Run the `convertps` command:

   `./convertps [-a] [-f] [-i` *ExportDir***`] [-o`** *ImportDir***`]`**

   Table 5-3 describes the options available for the `convertps` command. This two-column table lists the options in the first column and their descriptions in the second column.

Table 5-3 Options Available for Converting the Desktop

Option	Description
`-a`	Runs all modules without a menu. To choose to convert only the Desktop from the conversion menu, do not use the `-a` option.
`-f`	Converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server roles and places users under the organization. This option retains the multiple role to user assignment similar to Sun ONE Portal Server 3.0, but does not retain the hierarchical role functionality. All roles are created under the organization and are not prioritized. The display profile documents are prioritized and merged accordingly. To see if the roles have been migrated to roles, view the roles under the organization in the admin console. If you do not use the `-f` option, the conversion tool converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server suborganizations and places users under the suborganization. Without the `-f` option, the conversion tool retains the hierarchical functionality and customizations from Sun ONE Portal Server 3.0. The disadvantage is that it is difficult to move users from one suborganization to another. To see if the roles have been migrated to suborganizations, view the hierarchy in the admin console.
`-i` *ExportDir*	Specifies the input directory for the conversion. The input directory is the directory which contains the exported data created by the export tool. The conversion tool searches for export data in `/tmp/psExport` unless you specify a different input directory using the `-i` *ExportDir* option. If you specify a directory which does not contain data exported with the export tool, the conversion tool notifies you that the directory does not have export data and prompts you to enter a valid export directory.
`-o` *ImportDir*	Specifies the output directory for the conversion tool. The output directory is the directory used by the import tool. If you choose an import directory which already contains converted data, the conversion tool notifies you that modifying an existing migration may render import data unusable and prompts you to delete the import directory.

7. The system displays the following messages:

```
Found Portal Server version 6.2
```

```
Enter Identity Server Internal LDAP Authentication User password:
```

Type in a valid password.

```
Which directory should be created to store the converted data?
[/tmp/psImport]
```

You see this prompt only if you do not specify the output directory using the `-o` *ImportDir* option. You can choose the default directory or enter another import directory.

```
Import directory /tmp/psImport already exists.
```

```
If you do not wish to overwrite data within this directory, please
exit this migration process and rename the directory.


Delete the directory /tmp/psImport?
```

You see the message about an existing import directory only if the import directory already contains converted data. If you have already converted the Desktop, you should exit this migration process by pressing `Ctrl-C` and rename the directory before converting the data.

8. After you choose the output directory, you see output similar to the following:

```
Begin conversion process at Wed Jul 24 15:19:34 PDT 2003
Error file: /tmp/psImport/logs/error.28128
Report file: /tmp/psImport/logs/report.28128
Metrics file: /tmp/psImport/logs/convert_metrics.28128

Conversion Menu
1) LDAP Database
2) Gateway Rules to Rewriter Rules
3) Desktop
4) Certificate Databases
5) All of the above
6) Exit
Select one of the listed options to convert:
```

You see the conversion menu only if you do not specify the -a option.

9. Type **3** and press Enter to convert only the Desktop.

   The system displays messages similar to the following:

   ```
   *** Extracting templates ***


   *****

   Begin desktop conversion process at Wed Jul 24 15:36:06 PDT 2003

   *** Converting templates ***

   *** Converting jsp ***

   *** Converting soft links ***

   *** Converting resource bundles ***

   *** Converting static web content ***

   End desktop conversion process at Wed Jul 24 15:38:13 PDT 2003

   Successful completion of conversion process at Wed Jul 24 15:38:13
   PDT 2003
   ```

10. Change to the import directory making sure to substitute the import directory that you selected in Step 7. For example:

    cd *ImportDir*

    Substitute the import directory for *ImportDir*. If you used the default import directory, for example, use `/tmp/psImport`.

11. Type **ls** to see the directories and files created by the conversion tool.

    Table 5-4 shows the directories and files created by the conversion tool when converting the Desktop. This two-column table lists the directories and files in the first column and their descriptions in the second column.

Table 5-4 Directories and Files Created by the Conversion Tool When Converting the Desktop

Directory	Description
ImportDir/`desktop.tar`	Contains Desktop data.
ImportDir/`locale.tar`	Contains resource bundles.
ImportDir/`public_html.tar`	Contains static web data.
ImportDir/`auth.tar`	Contains new or modified authentication modules.
ImportDir/`logs`	Contains `error.PID`, `convert_metrics.PID`, and `report.PID` files (where `PID` is the process ID).

After the conversion has completed, or if you run into problems, check the *ImportDir*/`logs` directory for the `report.PID`, `error.PID`, and `convert_metrics.PID` files (where `PID` is the process ID). The `report.PID` file contains actions the conversion tool has or has not taken and also alerts you to any customizations which you need to convert manually. The `error.PID` file contains warnings or errors so that you can correct any problems. The `convert_metrics.PID` file contains metrics detailing when various elements of the conversion tool started, stopped, and the total time it took to convert the data. You can look at the `stdout` header to know which log and report files to examine for the `convertps` command you are running.

12. If you run the `convertps` tool again in order to convert data other than the Desktop, the tool will prompt you to delete the import directory. Type no if you wish to keep the Desktop data you have already converted.

Using the Import Tool

The import tool, `importps`, enables you to import the data exported and converted using the export and conversion tools to a Sun ONE Portal Server 6.2 system. The import tool searches for a directory containing valid data created by the conversion tool.

To Run the Import Tool

1. On the Sun ONE Portal Server 6.2 system, change to the import tool directory. For example:

   ```
   cd BaseDir/SUNWps/migration/bin
   ```

2. Run the `importps` command:

   ```
   ./importps [-a] [-k] [-m] [ImportDir]
   ```

   Table 5-5 describes the options available for importing the Desktop. This two-column table lists the options in the first column and their descriptions in the second column.

Table 5-5 Options Available for Importing the Desktop

Option	Description
`-a`	Runs all modules without a menu. To choose to import only the Desktop from the import menu, do not use the `-a` option.
`-k`	Specifies that the tool does not overwrite existing users. When you run `importps` without the `-k` option, if it encounters a conflict with an existing user, it will delete the existing user and import the new user entry defined in *ImportDir*`/ldap/user.ldif`. If you run `importps -k`, if it encounters a conflict with an existing user, it leaves the existing user and places the rejected user entry in *ImportDir*`/ldap/rejected_users.ldif`. If the user has a display profile document in the *ImportDir*`/dp/user/` directory, the display profile file may change. The `-k` option does not affect importing user display profile documents. The `-k` option only affects whether an existing LDAP user's LDIF record is updated. The updates to the LDIF record include everything except display profile updates.
`-m`	Merges display profile documents. When you run `importps` without the `-m` option, it overwrites any existing display profile documents it finds. This means that the root suffix, organization, suborganization, and user level display profile documents, if they exist, are overwritten. If you run `importps -m`, when the tool encounters a conflict with an existing display profile document it will merge it.

Table 5-5 Options Available for Importing the Desktop *(Continued)*

Option	Description
ImportDir	Specifies the directory where the converted data is located. If the directory does not exist or if the directory does not contain valid converted data, the import tool prompts you to enter a directory containing the converted data to be imported. If you do not specify an import directory, the import tool will search `/tmp/psImport`.

3. The system displays messages similar to the following:

   ```
   Found Portal Server version 6.2

   Enter Identity Server Internal LDAP Authentication User password:
   ```

   Type in a valid password.

   ```
   Enter Appserver Administrator password.
   ```

   You see this prompt only if you are migrating onto a Sun ONE Application Server web container.

   ```
   Begin import process at Thu Jul 25 16:20:40 PDT 2003

   Error file: /tmp/psImport/logs/importerror.2560

   Report file: /tmp/psImport/logs/importreport.2560

   Metrics file: /tmp/psImport/logs/import_metrics.2560


   Import Menu:

   1) LDAP Database

   2) Rewriter Rules

   3) Desktop

   4) Certificate Databases

   5) All of the above

   6) Exit
   ```

   You see the import menu only if you do not specify the `-a` option.

4. Type **3** and press Enter to import only the Desktop.

   The system displays messages similar to the following:

   ```
   Extracting desktop templates

   Extracting locale resource bundles

   Extracting static webserver data

   Redeploying portal web application

   Deploying to instance host1.siroe.com...

   Successful completion of import process at Thu Jul 25 16:25:30 PDT
   2003
   ```

   After the import has completed, or if you run into problems, check the *ImportDir*/`logs` directory for the `importreport.PID`, `importerror.PID`, and `import_metrics.PID` files (where `PID` is the process ID). The `importreport.PID` file contains actions the import tool has or has not taken and also alerts you to any customizations which you need to import manually. The `importerror.PID` file contains warnings or errors so that you can correct any problems. The `import_metrics.PID` file contains metrics detailing when various elements of the import tool started, stopped, and the total time it took to import the data. You can look at the `stdout` header to know which log and report files to examine for the `importps` command you are running.

Desktop Post-Migration Issues and Tasks

Depending on customizations you have made to the Sun ONE Portal Server 3.0 Desktop and depending on your personal preferences, you may need to perform manual steps to your Sun ONE Portal Server 6.2 system after migrating the Desktop.

This section describes post migration issues and tasks you may need to perform after migrating the Desktop.

Desktop Links and Colors

If you are using a tabbed Desktop, after you migrate the Desktop from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2, the links and colors on the Desktop in Sun ONE Portal Server 6.2 are different from the links and colors on the Desktop in Sun ONE Portal Server 3.0. This section describes these differences.

Links

In Sun ONE Portal Server 3.0, the Desktop links are:

```
Home | Options | Contents | Layout | Help | Log Out
```

After migrating the Sun ONE Portal Server 3.0 Desktop to Sun ONE Portal Server 6.2, if using the tabbed Desktop, the Desktop has the following links:

```
Home | Tabs | Options | Theme | Help | Log Out
```

Under each tab, there is a link for `Content` and `Layout`.

Colors

If you are using a tabbed Desktop, after you migrate from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2, the background is navy blue rather than black. This is because the Sun ONE Portal Server 6.2 menubar template for the tabbed Desktop is used. There is no one-to-one mapping between the Sun ONE Portal Server 3.0 `iwtDesktop` and the Sun ONE Portal Server 6.2 `TemplateTabContainer`, `TemplateTabContainerProvider`, and each `TemplateTabCustomTableContainerProvider`. Therefore the customization of the tabbed Desktop `menubar.template` and each tab subcontainer is a manual process after migration. After migration, each tab's `Content` and `Layout` links should be manually removed.

Background Color of the Channel Title Bar Does Not Line up with the Channel Border

Channels without action icons such as Help, Edit, and so on, do not align correctly with the channel content in Netscape 4.x. To resolve this problem, for each template container, modify `providerWrapper.template` by adding a non-breaking space before the provider commands tag. For example, instead of:

```
<TD BGCOLOR="#666699" NOWRAP>
  <P ALIGN=RIGHT>
    [tag:provider_cmds]
</TD>
```

you should use:

```
<TD BGCOLOR="#666699" NOWRAP>
  <P ALIGN=RIGHT>&nbsp;
```

```
    [tag:provider_cmds]
</TD>
```

If this is a tabbed Desktop, modify `TemplateTabCustomTableContainerProvider/providerWrapper.template` for each template type. You may need to modify other containers depending on your deployment.

Duplicate Portal Server / Sun ONE Portal Server Banner

It is possible, after importing with the `-m` option, that you will see duplicate Portal Server and Sun ONE Portal Server banners with Desktop links. This can happen when the `Samples` tab is looking for a `SamplesTemplatePanelContainer` directory in the XML documents. When it cannot find this directory it then uses the provider directory which is the `TemplateTabContainerProvider`. The HTML in this directory includes the additional banner. To prevent this from happening, you can simply copy an existing panel container, such as `MyFrontPageTemplatePanelContainer`, to `SamplesTemplatePanelContainer`. This will resolve this issue.

RSS Channel Does Not Display Content

The RSS channel may not display content if the channel's display profile URL attribute value retrieves content from an external system (not the portal server itself).

To resolve this issue, take the following steps:

1. On the remote host, add the entry `Content-Type=text/xml File Suffix=rdf,xml` to the web server's configuration `mime.types`, file. For example:

   ```
   type=text/xml                    exts=rdf,xml
   ```

2. Restart the web server on the remote host. For example:

   BaseDir`/SUNWips/bin/ipsserver start`

   or

   ```
   /etc/init.d/ipsserver start
   ```

3. On the portal server host, add the following rule to the `idsruleset`:

   ```
   <TagText tag="url" attributePatterns="" />
   ```

JavaScript Error

When using the Sun ONE Portal Server 6.2 menubar.template a JavaScript™ error is thrown because the Logout link expects to find the JavaScript function closePopup.

Online Help Is Not Localized

After a successful data migration, the channel help URLs will not have the locale in the URL. For example, the URLs will be similar to:

```
http://host/portal/docs/online_help/...
```

To localize the online help, update the display profile documents and provide the correct <ConditionalProperties> definition. For example, for the locale en_US use the <ConditionalProperties> definition to define the helpURL as:

```
<ConditionalProperties condition="locale" value="en">

    <ConditionalProperties condition="locale" value="US">

        <String name="helpURL"
            value="online_help/user_help/desktop/desktopTOC.html"
            advanced="true"/>

    </ConditionalProperties>

</ConditionalProperties>
```

After defining the helpURL, the Desktop Channel Help URL will be similar to:

```
http://host/portal/docs/en_US/online_help/...
```

Writing Data Migration Modules

See Appendix B, "Sun ONE Portal Server 3.0 Data Migration Module Author's Guide" for information on writing your own data migration modules.

Migrating Custom Channels and Custom Providers

This section discusses the steps you must take to migrate custom channels and custom providers from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2. In addition to following the steps for migrating the Desktop discussed in this chapter, manual changes to provider code are necessary for your custom providers to work in Sun ONE Portal Server 6.2. You will need to read the *Sun ONE Portal Server 6.0 Desktop Customization Guide* and the Javadocs for the Sun ONE Portal Server 6.2 Desktop in order to understand what you need to do.

Overview of the Process of Migrating Custom Providers

1. Run the export tool on the Sun ONE Portal Server 3.0 system.
2. Run the conversion tool on the Sun ONE Portal Server 6.2 system.
3. Run the import tool on the Sun ONE Portal Server 6.2 system.
4. Copy the custom Java™ files to a working directory where you can compile against the new install jar files.
5. Follow the steps in the section on “Making Manual Changes to Provider Code” on page 106 to rework code manually.
6. Compile using javac or Forte™ for Java, after importing the appropriate jar.
7. Add compiled provider classes to the classes directory.
8. Add any utility jar files to the classes directory.

You do not have to restart the server because the class loader will pick up the new classes.

Research and Planning

This section discusses research and planning recommendations that you should follow before migrating custom channels and custom providers.

Reading Documentation

Take the time to read the Sun ONE Portal Server 6.2 documentation, Sun ONE Portal Server 6.2 APIs, Sun ONE Identity Server 6.1, and Sun ONE Identity Server SDK and API documentation.

Training Your Staff

The documentation for Sun ONE Portal Server 6.2 and Sun ONE Identity Server 6.1 is extensive and serves as a great starting point to help you migrate to Sun ONE Portal Server 6.2. You may also wish to take advantage of any training available for Sun ONE Portal Server 6.2 and Sun ONE Identity Server 6.1.

Defining the Features of Sun ONE Portal Server 6.2 You Plan to Use

After reading the available documentation and attending any subsequent training, the next recommended step is to take into account all the new features you want your providers to take advantage of as you plan to migrate them. For instance, the non-tabbed Desktop is migrated to a `TemplateTableContainer` and the tabbed Desktop is migrated to a `TemplateTabContainer`.

Taking Inventory of Custom Provider Code Used in Your Sun ONE Portal Server 3.0 Deployment

You should create an inventory of custom provider code used in your Sun ONE Portal Server 3.0 deployment. This inventory is useful when initiating a project plan for migrating custom provider code and should account for ascertaining the answers to some preliminary questions like:

- Why was the code written?
- What does the code accomplish?
- Who is available, as a knowledge resource, for questions about the code?
- Where is the latest source code?
- Who is the resource to make the necessary modifications?
- Was the code written to address any discrepancies in the Sun ONE Portal Server 3.0 APIs that the new and expanded Sun ONE Portal Server 6.2 APIs might address?

Initiating a Project Plan for Ensuring Custom Provider Conversion

In order to accomplish conversion of custom providers from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2, it is recommended that a project plan be developed that indicates milestones, tasks, and resources necessary to smooth the transition of these providers.

Preparing Your Test Environment

In order to migrate custom providers, you should set up a test environment of Sun ONE Portal Server 6.2. This can be done on the same system as your Sun ONE Portal Server 3.0 test environment or on a clean system.

Running the Migration Tools

Once your Sun ONE Portal Server 6.2 system is set up, you should run the migration tools as documented in this guide. This will migrate the provider's LDAP data (including custom provider attributes), provider templates, and so on to the Sun ONE Portal Server 6.2 environment.

Validating Custom Provider Data Migration

After migrating from Sun ONE Portal Server 3.0, you should see the following items already migrated for your custom providers by the tools:

- XML files. Sun ONE Portal Server 3.0 channel and provider definitions are migrated to Sun ONE Portal Server 6.2 as channel and provider definitions in the display profile documents. Sun ONE Portal Server 3.0 providers are migrated to the Sun ONE Portal Server 6.2 root suffix global display profile document. This is configurable in the admin console. See the *Sun ONE Portal Server* 6.2 *Administrator’s Guide* for information about the admin console.

- Template files. The templates have been migrated. They exist in `/etc/opt/SUNWps/desktop/default/`*originalProviderName*.

- Properties files. The properties files are migrated and deployed in *BaseDir*`/SUNWps/web-src/WEB-INF/classes`. The property files normally have the same name as the provider name. If a different name is used, you should use the `getResourceBundle(`*theSpecificName*`)` method in the provider code to get the resource bundle.

Making Manual Changes to Provider Code

This section discusses the steps you need to follow in order to change provider code manually so that your Sun ONE Portal Server 3.0 custom providers work in Sun ONE Portal Server 6.2. You will need to read the Javadocs for the Sun ONE Portal Server 6.2 Desktop in order to make these correctly.

1. Move your source code to the Sun ONE Portal Server 6.2 system if it is different from the Sun ONE Portal Server 3.0 system. The classes directory is specified in the `/etc/opt/SUNWps/desktop/desktopconfig.properties` file. By default, the directory is *BaseDir*`/SUNWps/desktop/classes`.

2. Set your `CLASSPATH` environment variable. Before compilation you need to change your `CLASSPATH` setting to include the Desktop SDK `jar`, for example:

   `CLASSPATH:` *BaseDir*`/SUNWps/sdk/desktop/desktopsdk.jar`

   If you are using any Sun ONE Identity Server APIs directly, you may also need to include the Sun ONE Identity Server SDK.

   Include any other Java classes and `jar` files that your custom provider code relies on.

3. Change import statements by removing:

   - `session.Session`
   - `session.SessionException`
   - `util.Debug`

4. Use the new debug by making the following changes:

   - Comment out import.
   - Comment out static block.
   - Use `getProviderContext().debugMessage(“message goes here”)`.
   - Use `getProviderContext().isDebugMessageEnabled()` boolean check around the log statement

5. Change getStringProperty calls by removing the first string in the signature, component, by changing from:

   ```
   getStringProperty(component, tag, default)
   ```

   to:

   ```
   getStringProperty(tag, default)
   ```

   Another API which enables you to get the localized version of the string from the display profile is getStringProperty(tag, default, true);.

   You can also use methods such as getBooleanProperty and getIntegerProperty, which deal with Boolean and integer values.

6. Wrap getStringProperty calls with try / catch for ProviderContextException. getStringProperty throws ProviderException.

7. Change calls to getResourceBundle so they are wrapped in a try block that catches the ProviderException.

8. Remove the store() method. This has been eliminated because getStringProperty and others have been extended to encompass this functionality.

9. Changes to getContent(Map) are optional. See the provider API documentation.

10. Compile your code. Use the javac command line script to compile your code. Then, discover and fix any errors. There will most likely be errors that involve the specifics of your custom provider.

11. Verify your migrated custom providers by checking:

    - That the channel properties in the display profile have been migrated correctly.

    - That users can log in and the migrated channels based on the custom provider are available on the Desktop.

    - That the custom provider has no errors in normal operation (display of content, edit, help, and so on). You can check logs in /var/opt/SUNWam/debug/.

    - That your channel's attributes based on the custom provider can be administered via the Sun ONE Portal Server 6.2 admin console and that changing these attributes has no adverse affect for your users.

    - Desktop errors located in the /var/opt/SUNWam/debug/desktop.debug file and other errors located in files with corresponding names in that directory.

Common Errors Encountered and Recommended Fixes

The list below includes common errors that you may encounter when compiling your custom providers via the javac command-line script. It includes answers to these issues as known to date.

- A warning message indicates that Sun ONE Portal Server 6.2 - PAPI Class(es) not found.
  - Did you modify your import statements to account for changes to Portal Server 6.2 - PAPI? See Step 3 in the section on "Making Manual Changes to Provider Code" on page 106.
  - Did you update your CLASSPATH? See Step 2 in the section on "Making Manual Changes to Provider Code" on page 106.
- A warning message indicates a class or a method in the Sun ONE Portal Server 6.2 - PAPI has changed or been deprecated.
  - Check the Javadocs at: `http://`*your_install_url*`/portal/javadocs`
- A warning message indicates that a class is no longer available.
  - Check the Sun ONE Portal Server 6.2 documentation and sample providers to see how new providers are being implemented.
- A warning message indicates that a method signature is not found.
  - Check the PAPI for deprecated or removed methods. In some instances, the methods have moved to a different class because they have been categorized differently. In other cases, the signature, such as `getStringProperty` has changed. See, for example, Step 8 in the section on "Making Manual Changes to Provider Code" on page 106.

NOTE Because of changes, deletions, and modifications to APIs, check to see if any of the previous public APIs now deprecated apply to the error you have encountered.

Information About Channel Migration

For any channel, these are the other important components:

- The properties files, including custom provider properties files are migrated and located in *BaseDir*/`SUNWps/web-src/WEB-INF/classes`.
- The template files are migrated to `/etc/opt/SUNWps/desktop/default/`*originalProviderName*.
- The XML files are migrated. Sun ONE Portal Server 3.0 channel and provider definitions are migrated to Sun ONE Portal Server 6.2 as channel and provider definitions in the display profile documents. Depending on the customizations you have made, Sun ONE Portal Server 3.0 channels are generally migrated to organization, suborganization or role, and user level display profile documents. This includes available and selected channel lists for the Desktop container. You can access the display profile through the admin console.

Example Custom Provider: Manual Code Changes for Migration

```
//change package from: com.iplanet.portalserver.providers.quotations;
package com.sun.portal.providers.quotations;

// Use some basic Java classes
import java.lang.*;
import java.io.*;
import java.util.*;
import java.net.*;

// Use some basic Java Servlet classes
import javax.servlet.*;
import javax.servlet.http.*;

// Use some iPS classes
//change to correct package name:
//import com.iplanet.portalserver.desktop.util.*;
import com.sun.portal.desktop.util.*;

//import com.iplanet.portalserver.providers.*;
import com.sun.portal.providers.*;

//change:  import com.iplanet.portalserver.desktop.util.ParameterMap;
import com.sun.portal.desktop.util.ParameterMap;
//change: import com.iplanet.portalserver.desktop.util.Target;
import com.sun.portal.desktop.util.Target;
//change: import com.iplanet.portalserver.providers.ProfileProviderAdapter;
import com.sun.portal.providers.ProfileProviderAdapter;
//change: import com.iplanet.portalserver.providers.Provider;
import com.sun.portal.providers.Provider;
//change: import com.iplanet.portalserver.providers.ProviderException;
import com.sun.portal.providers.ProviderException;

//remove:  import com.iplanet.portalserver.session.Session;
```

```
//change package from: com.iplanet.portalserver.providers.quotations;
//remove:  import com.iplanet.portalserver.session.SessionException;
//remove:  import com.iplanet.portalserver.util.Debug;
//remove: import com.iplanet.portalserver.profile.*;

// The QuotationProvider Class
//   This is a sample desktop provider application used to display
//   famous/humorous quotations on the desktop.
//
//   It shows how to use the BaseProvider API to:
//     - display html content on the desktop
//     - allow the user to edit attributes
//
//   It also invokes the Session, Profile and Logging APIs in order to:
//     - get/set user-level attributes regarding quotation preferences
//     - provide logging


public class QuotationProvider extends ProfileProviderAdapter implements Provider {

   //remove:  private static Debug debug = Debug.getInstance(“iwtQuotationsProvider”);

   //remove:  static {
   // debug.setDebug();
   //}

    // The constructor - don’t need to do any special initialization
    // so just invoke the ProfileProviderAdapter’s constructor
    public QuotationProvider() {
        super();
    }


    // Generate the HTML that will be displayed in this content provider’s
    // area on the desktop
//getContent has a new signature, this method will be deprecated
    public StringBuffer getContent(Map m) throws ProviderException {

        StringBuffer content = new StringBuffer(); // Contains the HTML to be
                                                   //   displayed
        Hashtable quotesHash = new Hashtable();    // A Hash of quote vectors,
                                                   //   one vector per category
        BufferedReader quotesReader;               // Used to read the quotes
        String quote;                              // A single quotation
        Random randomGenerator = new Random();     // To randomly pick quotes

        // Read in the quotations into a Hashtable of Vectors, each containg
        // the quotations of a specific category. If we can’t read the
        // quotations file it’s okay - we just won’t display any quotes later.
        // The category and quotation are separated by a “|” in the quotations
        // file.

        try {
          //Change: all getStringProperty() calls to accept only the last two arguements,
the component arg was removed.
           // Change: String quotationFileName = getStringProperty(“iwtQuotationProvider,
“fileLocation”, “/”);
```

```
//change package from: com.iplanet.portalserver.providers.quotations;
           String quotationFileName = getStringProperty(“iwtQuotationProvider,
“fileLocation”, “/”);
           quotesReader = new BufferedReader(new FileReader(new File(quotationFileName)));
            while ((quote = quotesReader.readLine()) != null) {
                String type = quote.substring(0,quote.indexOf(‘|’));
                if (!quotesHash.containsKey(type)) {
                    quotesHash.put(type,new Vector());
                }

((Vector)(quotesHash.get(type))).addElement(quote.substring(quote.indexOf(‘|’)+1));
            }
        } catch (ProviderContextException pe) {

          }catch (Exception e) {
        }

        // Determine is user’s preference is to display the category along with
        // a quotation.
        boolean displayCategories = true;


         // Change: if (getStringProperty(“iwtQuotationProvider”, “displayCategories”,
“false”).equals(“false”)) {
        try {
        if (getStringProperty(“displayCategories”,”false”).equals(“false”)) {
        } catch (ProviderContextException pe) { }
            displayCategories = false;
        }

          // Get the possible quotation categories.
          Enumeration e =
           //Change: getListProperty(“iwtQuotationProvider”, “selectedCategories”, new
Vector()).elements();
           getListProperty(“selectedCategories”, new Vector()).elements();

          // Print a quotation for each category selected by the user.
          while (e.hasMoreElements()) {
            String category = (String)e.nextElement();

            // If the user wanted to display the category along with the
            // quotation then do so.
            if (displayCategories) {
               //Remove:  content = getTemplate(“category.template”);
               Hashtable catgoryTags = new Hashtable();
             try {
                  String category = getStringProperty(“category”, ““);
              } catch (ProviderContextException pe) { }
           //Remove: content.append(category);
                  contentTags.put(“quotations-category”, category);

              //remove:  content = swap.doSwap(content,contentTags);
              content.append( getTemplate(“category.template”,categoryTags) );

            }

            // If there are any quotations for this category, pick a random
//change package from: com.iplanet.portalserver.providers.quotations;
            // one and display it, otherwise note that we didn’t find one.
            if (quotesHash.containsKey(category)) {
                Vector quoteVector = (Vector)quotesHash.get(category);
                int whichQuoteIndex =
Math.abs(randomGenerator.nextInt())%(quoteVector).size();
                content.append(“<UL>”);
                content.append(“<LI>”);
                content.append((String)(quoteVector.elementAt(whichQuoteIndex)));
                content.append(“</LI>”);
                content.append(“</UL>”);
            } else {
                content.append(“<UL>”);
                content.append(“<LI>”);
                content.append(“(no quotes of this type found)”);
                content.append(“</LI>”);
                content.append(“</UL>”);
           //New:  add catch for provider exception around Resource Bundles.
               try {
                     errMsg =
getResourceBundle(“iwtQuotationProvider”).getString(“QuotationProvider-NoQuotesFound”);
           } catch (ProviderException pe) {
                       errMsg = “There are no quotes found.”;
                }
                //New: use new message logging functionality
               if (getProviderContext().isDebugMessageEnabled()) {
                      getProviderContext().debugMessage( errMsg );
                }
            }
        }

        // Now return the HTML we have constructed.
        return content;
    }

    // Generate the HTML that will be displayed to let the user set
    // his/her preferneces.
//optional: getEdit() method signature is different, customer may want to move to new
implementation
    public StringBuffer getEdit(Map m) throws ProviderException {
        // Contains the HTML to be displayed
        StringBuffer content = new StringBuffer();

        // Display things in a single column table - so things line up.
        content.append(“<P><TABLE>”);

        // Let the user choose which categories s/he wants.
        content.append(“<TR>”);
        content.append(“<TD VALIGN=\”top\” HALIGN=\”left\”>”);
        content.append(“Types of Quotes to Display”);
        content.append(“</TD>”);

        // Get the user’s selected categories
        Vector selectedCategories = new Vector();
        Enumeration e2 =
           getListProperty(“selectedCategories”, new Vector()).elements();
```

```
//change package from: com.iplanet.portalserver.providers.quotations;
        while (e2.hasMoreElements()) {
            selectedCategories.addElement((String)e2.nextElement());
        }

        // List the possible categories, marking those that have already
        // been selected by the user.
        content.append(“<TD VALIGN=\”top\” HALIGN=\”left\”>”);
        Enumeration e =
           getListProperty(“possibleCategories”, new Vector()).elements();

        while (e.hasMoreElements()) {
            String category = (String)e.nextElement();
            content.append(“<INPUT TYPE=\”checkbox\” NAME=\”category-”);
            content.append(category);
            content.append(“\” VALUE=\””);
            content.append(category);
            content.append(“\””);
            if (selectedCategories.contains(category)) {
                content.append(“ CHECKED”);
            }
            content.append(“>”);
            content.append(category);
            content.append(“<BR>”);
        }
        content.append(“</TD>”);
        content.append(“</TR>”);

        // Let the user decide if they want to display a category with
        // each quotation.
        content.append(“<TR>”);
        content.append(“<TD VALIGN=\”top\” HALIGN=\”left\”>”);
        content.append(“<INPUT TYPE=\”checkboX\” NAME=\”display_categories\”
VALUE=\”yes\””);
        try {
        if (getStringProperty(“displayCategories”,”true”).equals(“true”)) {
            content.append(“ CHECKED”);
        }
        } catch (ProviderContextException pe) { }
        content.append(“>Display the Category Along With the Quotation”);
        content.append(“</TD>”);
        content.append(“</TR>”);

        content.append(“</TABLE>”);

        return content;
    }

    // Handle the submittal of the edit form.
//optional: processEdit() method signature is different, customer may want to move to new
implementation
    public URL processEdit(Map m) throws ProviderException {
        Hashtable req = new Hashtable(m);

        Enumeration parameterNames = req.keys();

        Vector categorySelections = new Vector(); // Categories user picked
```

```
//change package from: com.iplanet.portalserver.providers.quotations;
        boolean displayCategories = false;        // Display category too ?

        // Check all the passed parameters and add the selected categories
        // as well as check to see if user opted to display categories.
        while (parameterNames.hasMoreElements()) {
            String parameter = (String)parameterNames.nextElement();
            if (parameter.equals(“display_categories”)) {
                displayCategories = true;
            } else if (parameter.startsWith(“category-”)) {
                String []category = (String [])req.get(parameter);
                categorySelections.addElement(category[0]);
            }
        }

        // Set the list of selected categories for later storing in the
        // profile database.
        setListProperty(“selectedCategories”, categorySelections);

        // Set whether the user wants to display categories or not for later
        // storing in the profile database.
        if (displayCategories) {
           setStringProperty(“displayCategories”, “true”);
        } else {
           setStringProperty(“displayCategories”, “false”);
        }

        // Done processing the edit form - now store the user’s preferences.
        //Remove: store() method is obsolete
        super.processEdit(parameterNames); ?

        //} catch (ProfileException pe) { }
        return null;
    }
}
```

Chapter 6

Migrating Certificates

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

This chapter provides procedures to migrate Sun ONE Portal Server 3.0 certificates to the Sun™ ONE Portal Server 6.2 system. If the name of the Sun ONE Portal Server 6.2 system is different from the Sun ONE Portal Server 3.0 system, you will need new certificates. Migrating certificates involves using the export, conversion, and import tools. The migration tools do not migrate gateway certificates. This chapter contains the following sections:

- Using the Export Tool
- Using the Conversion Tool
- Using the Import Tool
- Migrating SSL Certificates on a Multi-Instance Installation
- Migrating Open Mode SSL Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2

Using the Export Tool

The export tool, `exportps`, stores data from the Sun ONE Portal Server 3.0 system to an export directory. To migrate Sun ONE Portal Server 3.0 certificates, you need to begin by running the export tool to store certificate data to disk.

To Run the Export Tool

1. On the Sun ONE Portal Server 3.0 system, change to the migration tool directory:

   cd *BaseDir***/SUNWps/migration/bin/**

2. Run the `exportps` command:

   ./exportps [-a] [-p *LDAP passphrase***] [***ExportDir***]**

   Table 6-1 describes the options available for exporting Sun ONE Portal Server 3.0 certificate databases. This two-column table lists the options in the first column and their descriptions in the second column.

Table 6-1 Options Available for Exporting Sun ONE Portal Server 3.0 certificate databases

Option	Description
`-a`	Runs all modules without a menu. To choose to export only certificate databases from the export menu, do not use the `-a` option.
`-p` *LDAP passphrase*	Sets the LDAP passphrase in order to avoid the LDAP passphrase prompt. If you do not use the `-p` *LDAP passphrase* option with the `-a` option, the tool will prompt you for the LDAP passphrase.
ExportDir	Specifies the directory where the information is to be saved. If you do not use the *ExportDir* option, the tool prompts you for the export directory.

3. The system displays a message similar to the following:

   ```
   Which directory should be created to store the Portal Server system?
   [/tmp/psExport]
   ```

   You see this prompt only if you do not specify the output directory on the command line. You can choose the default directory `/tmp/psExport` or enter the export directory.

   Type the directory and press Enter.

4. The system displays a message similar to the following:

   ```
   Export directory /tmp/psExport already exists.

   If you do not wish to overwrite data within this directory, please
   exit this migration process and rename the directory.


   Delete the directory /tmp/psExport?
   ```

   You see this message only if there is already exported data in the export directory. If you have not already exported the certificate databases, you may keep the export directory without risking any conflicts in the data. If you have already exported the certificate databases, you should delete the directory to avoid conflicts.

5. The system displays a message similar to the following:

   ```
   Found iPS version 3.0sp5

   Begin export process at  Wed Jul 31 12:50:12 PDT 2003

   Error file: /tmp/psExport/logs/error.7926

   Report file: /tmp/psExport/logs/report.7926

   Metrics file: /tmp/psExport/logs/export_metrics.7926


   Export Menu:

   1) LDAP Database

   2) Desktop

   3) Certificate Databases

   4) All of the above

   5) Exit

   Select one of the listed options to export:
   ```

   You see the export menu only if you do not specify the `-a` option.

6. Type **3** and press Enter to export the certificate databases only.

7. The system displays the following message:

```
Copying certificates


Successful completion of export process at  Wed Jul 31 12:56:27 PDT
2003
```

8. Change to the export directory you specified in Step 2 or Step 3. For example:

   cd *ExportDir*

   Substitute the export directory for *ExportDir*. If you used the default export directory, for example, use /tmp/psExport.

   Type **ls** to see the directories and files created by the export tool.

   Table 6-2 shows the directories and files created by the export tool. This two-column table lists the directories in the first column and their descriptions in the second column.

Table 6-2 Directories Created by the Export Tool When Exporting Certificate Databases

Directory	Description
ExportDir/logs	Contains error.PID, export_metrics.PID, and report.PID files (where PID is the process ID)
ExportDir/webcert	Contains exported web server certificates.
ExportDir/gatecert	Placeholder for Sun ONE Portal Server 6.2 Secure Remote Access Pack migration.

9. After the export has completed, or if you run into problems, check the *ExportDir*/logs directory for the report.PID, error.PID, and export_metrics.PID files (where PID is the process ID). The report.PID file contains actions the export tool has or has not taken. The error.PID file contains warnings or errors so that you can correct any problems. The export_metrics.PID file contains metrics detailing when various elements of the export tool started, stopped, and the total time it took to export the data. You can look at the stdout header to know which log and report files to examine for the exportps command you are running.

10. Examine the exported data and manually change, before converting the data, all instances of the name of the Sun ONE Portal Server 3.0 system to the name of the Sun ONE Portal Server 6.2 system. If the port numbers on the two systems are different, you must change these manually as well.

    If you are performing a single-system migration, server names will be the same, but port numbers will be different. You need to change port numbers manually before converting the data.

    The migration tools do not modify server names and port numbers.

    For example, in the following extract of the exported data in the *ExportDir*`/config/platform.config` file, you need to change manually all instances of `host1.siroe.com`, which is the name of the Sun ONE Portal Server 3.0 system, to the name of the Sun ONE Portal Server 6.2 system if the systems use different names. If the systems use different port numbers, you need to change them as well.

```
ips.defaultDomain=siroe.com

ips.server.protocol=http

ips.server.host=host1.siroe.com

ips.server.port=8080

ips.profile.host=host1.siroe.com

ips.profile.port=8080

ips.gateway.protocol=https

ips.gateway.host=host1.siroe.com

ips.gateway.port=443

ips.gateway.trust_all_server_certs=false

ips.virtualhost=host1.siroe.com 192.18.66.42

ips.naming.url=http://host1.siroe.com:8080/namingservice

ips.notification.url=http://host1.siroe.com:8080/notificationservic
e
```

Using the Conversion Tool

The conversion tool, `convertps`, enables you to select which data to convert from the data exported from a valid installation of Sun ONE Portal Server 3.0. Because the conversion tool runs on the Sun ONE Portal Server 6.2 system, you need to move the export directory from the Sun ONE Portal Server 3.0 system to the Sun ONE Portal Server 6.2 system.

To Run the Conversion Tool

1. On the Sun ONE Portal Server 3.0 system, change to the directory above the `export` directory containing the data output by the export tool.

   For example, if you used the default directory (`/tmp/psExport`) for exporting the certificate databases, you would type:

   ```
   cd /tmp
   ```

2. Save the export directory using the `tar` command. For example:

   ```
   tar cvf export.tar psExport
   ```

3. Use an FTP program to transfer the `export.tar` file to the Sun ONE Portal Server 6.2 system.

4. Extract the files from `export.tar` using the `tar` command. For example:

   ```
   tar xvf export.tar
   ```

5. On the Sun ONE Portal Server 6.2 system, change to the migration tool directory:

   ```
   cd BaseDir/SUNWps/migration/bin
   ```

6. Run the `convertps` command:

   ```
   ./convertps [-a] [-f] [-i ExportDir] [-o ImportDir]
   ```

   Table 6-3 describes the options available for the `convertps` command. This two-column table lists the options in the first column and their descriptions in the second column.

Table 6-3 Options Available for Converting Certificate Databases

Option	Description
`-a`	Runs all modules without a menu. To choose to convert only certificate databases from the conversion menu, do not use the `-a` option.
`-f`	Converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server roles and places users under the organization. This option retains the multiple role to user assignment similar to Sun ONE Portal Server 3.0, but does not retain the hierarchical role functionality. All roles are created under the organization and are not prioritized. The display profile documents are prioritized and merged accordingly. To see if the roles have been migrated to roles, view the roles under the organization in the admin console. If you do not use the `-f` option, the conversion tool converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server suborganizations and places users under the suborganization. Without the `-f` option, the conversion tool retains the hierarchical functionality and customizations from Sun ONE Portal Server 3.0. The disadvantage is that it is difficult to move users from one suborganization to another. To see if the roles have been migrated to suborganizations, view the hierarchy in the admin console.
`-i` *ExportDir*	Specifies the input directory for the conversion. The input directory is the directory which contains the exported data created by the export tool. The conversion tool searches for export data in `/tmp/psExport` unless you specify a different input directory using the `-i` *ExportDir* option. If you specify a directory which does not contain data exported with the export tool, the conversion tool notifies you that the directory does not have export data and prompts you to enter a valid export directory.
`-o` *ImportDir*	Specifies the output directory for the conversion tool. The output directory is the directory used by the import tool. If you choose an import directory which already contains converted data, the conversion tool notifies you that modifying an existing migration may render import data unusable and prompts you to delete the import directory.

7. The system displays the following messages:

```
Found Portal Server version 6.2
```

```
Enter Identity Server Internal LDAP Authentication User password:
```

Type in a valid password.

```
Which directory should be created to store the converted data?
[/tmp/psImport]
```

You see this prompt only if you do not specify the output directory using the `-o` *ImportDir* option. You can choose the default directory or enter another import directory.

```
Import directory /tmp/psImport already exists.
```

```
If you do not wish to overwrite data within this directory, please
exit this migration process and rename the directory.
```

```
Delete the directory /tmp/psImport?
```

You see the message about an existing import directory only if the import directory already contains converted data. If you have already converted the certificate databases, you should exit this migration process by pressing `Ctrl-C` and rename the directory before converting the data.

8. After you choose the output directory, you see output similar to the following:

   ```
   Begin conversion process at Thu Aug  1 16:21:48 PDT 2003

   Error file: /tmp/psImport/logs/error.12111

   Report file: /tmp/psImport/logs/report.12111

   Metric file: /tmp/psImport/logs/convert_metrics.12111


   Conversion Menu

   1) LDAP Database

   2) Gateway Rules to Rewriter Rules

   3) Desktop

   4) Certificate Databases

   5) All of the above

   6) Exit

   Select one of the listed options to convert:
   ```

   You see the conversion menu only if you do not specify the `-a` option.

9. Type **4** and press Enter to convert only certificate databases.

   The system displays messages similar to the following:

   ```
   *** Extracting templates ***


   *****

   Begin certificate conversion process at Thu Aug  1 16:25:24 PDT 2003

   End certificate conversion process at Thu Aug  1 16:25:24 PDT 2003

   Successful completion of conversion process at Thu Aug  1 16:25:25
   PDT 2003
   ```

10. Change to the import directory making sure to substitute the import directory that you selected in Step 7. For example:

    ```
    cd ImportDir
    ```

    Substitute the import directory for *ImportDir*. If you used the default import directory, for example, use `/tmp/psImport`.

11. Type **ls** to see the directories created by the conversion tool.

    Table 6-4 shows the directories created by the conversion tool when converting certificate databases. This two-column table lists the directories in the first column and their descriptions in the second column.

Table 6-4 Directories Created by the Conversion Tool When Converting Certificate Databases

Directory	Description
ImportDir/logs	Contains error.PID, export_metrics.PID, and report.PID files (where PID is the process ID)
ImportDir/webcert	Contains converted web server certificates.
ImportDir/gatecert	Placeholder for Sun ONE Portal Server 6.2 Secure Remote Access Pack migration.

    After the conversion has completed, or if you run into problems, check the *ImportDir*/logs directory for the report.PID, error.PID, and convert_metrics.PID files (where PID is the process ID). The report.PID file contains actions the conversion tool has or has not taken and also alerts you to any customizations which you need to convert manually. The error.PID file contains warnings or errors so that you can correct any problems. The convert_metrics.PID file contains metrics detailing when various elements of the conversion tool started, stopped, and the total time it took to convert the data. You can look at the stdout header to know which log and report files to examine for the convertps command you are running.

12. If you run the convertps tool again in order to convert data other than certificate databases, the tool will prompt you to delete the import directory. Type no if you wish to keep the certificate databases you have already converted.

Using the Import Tool

The import tool, importps, enables you to import the data exported and converted using the export and conversion tools to a Sun ONE Portal Server 6.2 system. The import tool searches for a directory containing valid data created by the conversion tool.

To Run the Import Tool

1. On the Sun ONE Portal Server 6.2 system, change to the import tool directory. For example:

   cd *BaseDir*/**SUNWps/migration/bin**

2. Run the `importps` command:

   ./importps [-a] [-k] [-m] [*ImportDir***]**

   Table 6-5 describes the options available for importing certificate databases. This two-column table lists the options in the first column and their descriptions in the second column.

Table 6-5 Options Available for Importing Certificate Databases

Option	Description
`-a`	Runs all modules without a menu. To choose to import only the certificate databases from the import menu, do not use the `-a` option.
`-k`	Specifies that the tool does not overwrite existing users. When you run `importps` without the `-k` option, if it encounters a conflict with an existing user, it will delete the existing user and import the new user entry defined in *ImportDir*`/ldap/user.ldif`. If you run `importps -k`, if it encounters a conflict with an existing user, it leaves the existing user and places the rejected user entry in *ImportDir*`/ldap/rejected_users.ldif`. If the user has a display profile document in the *ImportDir*`/dp/user/` directory, the display profile file may change. The `-k` option does not affect importing user display profile documents. The `-k` option only affects whether an existing LDAP user's LDIF record is updated. The updates to the LDIF record include everything except display profile updates.
`-m`	Merges display profile documents. When you run `importps` without the `-m` option, it overwrites any existing display profile documents it finds. This means that the root suffix, organization, suborganization, and user level display profile documents, if they exist, are overwritten. If you run `importps -m`, when the tool encounters a conflict with an existing display profile document it will merge it.

Table 6-5 Options Available for Importing Certificate Databases *(Continued)*

Option	Description
ImportDir	Specifies the directory where the converted data is located. If the directory does not exist or if the directory does not contain valid converted data, the import tool prompts you to enter a directory containing the converted data to be imported. If you do not specify an import directory, the import tool will search `/tmp/psImport`.

3. The system displays messages similar to the following:

```
Found Portal Server version 6.2
Enter Identity Server Internal LDAP Authentication User password:
```

Type in a valid password.

```
Enter Appserver Administrator password.
```

You see this prompt only if you are migrating onto a Sun ONE Application Server web container.

```
Begin import process at Thu Aug  1 16:43:15 PDT 2003
Error file: /tmp/psImport/logs/importerror.12362
Report file: /tmp/psImport/logs/importreport.12362
Metric file: /tmp/psImport/logs/import_metrics.12362

Import Menu:
1) LDAP Database
2) Rewriter Rules
3) Desktop
4) Certificate Databases
5) All of the above
6) Exit
Select one of the listed options to import:
```

You see the import menu only if you do not specify the `-a` option.

4. Type **4** and press Enter to import only the Desktop.

   The system displays messages similar to the following:

   ```
   Importing Certificates

   Redeploying portal web application

   Deploying to instance host1.siroe.com...

   Successful completion of import process at Thu Aug  1 16:46:30 PDT
   2003
   ```

5. After the import has completed, or if you run into problems, check the *ImportDir*/`logs` directory for the `importreport.PID`, `importerror.PID`, and `import_metrics.PID` files (where `PID` is the process ID). The `importreport.PID` file contains actions the import tool has or has not taken and also alerts you to any customizations which you need to import manually. The `importerror.PID` file contains warnings or errors so that you can correct any problems. The `import_metrics.PID` file contains metrics detailing when various elements of the import tool started, stopped, and the total time it took to import the data. You can look at the `stdout` header to know which log and report files to examine for the `importps` command you are running.

Migrating SSL Certificates on a Multi-Instance Installation

For details on using SSL with Sun ONE Portal Server 6.2, see the *Sun ONE Portal Server* 6.2 *Administrator's Guide.*

To migrate SSL certificates on a multi-instance installation:

1. Create the instance.

2. Create the certificate database for the instance.

3. Change to the *BaseDir*/`SUNWam/servers/alias` directory and copy the `cert7.db` and `key3.db` over the newly created db files. For example:

   ```
   cp https-host1.siroe.com@8443-host1-cert7.db
   https-instance2-host1-cert7.db

   cp https-host1.siroe.com@8443-rose16-key3.db
   https-instance2-host1-key3.db
   ```

4. Restart the server.

Migrating Open Mode SSL Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2

This section contains the procedures for migrating open mode SSL from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2. The overall process for migrating SSL includes the following procedures:

- Obtaining the Certificate
- Creating and Initializing a New Sun ONE Portal Server 6.2 Instance
- Requesting a Certificate
- Installing the Certificate

Obtaining the Certificate

If migrating to a machine with the same server name:

1. Obtain a copy the backup of the certificate from the Sun ONE Portal Server 3.0 server.
2. Copy the certificate to the Sun ONE Portal Server 6.2 server.

If migrating to a machine with a different server name:

- Obtain a new certificate from a Certificate Authority as certificates cannot be transferred from one host to another.

Creating and Initializing a New Sun ONE Portal Server 6.2 Instance

As root, from the command line interface, perform the following steps on the Sun ONE Portal Server 6.2 server:

1. Go to the directory *BaseDir*`/SUNWps/bin`.

   `cd` *BaseDir***`/SUNWps/bin`**

2. Type the following to create a new Sun ONE Portal Server 6.2 instance.

   `./multiserverinstance`

3. When prompted by the `mutliserverinstance` script, enter the instance nickname and port number.

4. Open a browser window and go to `http://`*server*`:`*admin_port*. For example, **`http://host1.siroe.com:8088`**

5. When the popup window appears enter the Admin UserId and password.

   - The default value for the Admin UserId is `admin`.
   - The password is the passphrase entered during installation.

6. Select the new server instance name.

7. Choose manage.

8. If the Trust Database has not been initialized previously, enter a database password in both text boxes and click OK. If the Trust Database has been initialized previously do not perform this step.

   You will receive a popup confirmation that the database has been initialized.

Requesting a Certificate

If you already have a certificate, you can skip this section and proceed with Installing the Certificate.

1. Click on Request a certificate on the sidebar.
2. Select New Certificate or Renew Certificate.
3. Select a CA to send the request to.
4. Select Internal for Cypotgraphic module.
5. Enter the password name.
6. Enter in the appropriate information and select OK.

Installing the Certificate

1. Click Install Certificate.
2. Select the server on which to install the certificate.
3. Leave the certificate name blank.

4. If the certificate is in a file, select Message is in this file, or cut and paste the certificate into the text area.

5. Choose OK.

   A new screen appears with the certificate information.

6. Choose Add Server Certificate.

7. Select the Servers tab.

8. Select the server that you want to run SSL and select Manage.

9. Select the Preferences tab and select Add Listen Socket.

10. Select Edit listen Sockets on the sidebar.

11. Choose on from the Security drop down and Click OK.

12. Restart the servers using the following command.

    BaseDir`/SUNWam/bin/amserver startall`

13. Because the Sun ONE Portal Server 3.0 `iwtLoginProvider` templates are not being migrated, the Sun ONE Portal Server 6.2 templates must be manually edited.

    a. **`cd /etc/opt/SUNWps/desktop/default/Login`**

    b. In any of the `display*.html` files, add `https://`*server_name*`:`*port*`/` to the following:

```
<FORM ACTION=”https://server_name:port/amserver/UI/login” ...>
```

```
<A HREF=”https://server_name:port/amserver/UI/login ...”>
```

Chapter 7

Migrating the Gateway, Netlet, and NetFile

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

This chapter describes the migration of the Sun ONE Portal Server 3.0 gateway attributes, NetFile, and Netlet to Sun™ ONE Portal Server, Secure Remote Access 6.2.

This chapter contains the following sections:

- Overview
- Migrating Sun ONE Portal Server 3.0 Gateway Data
- Verifying the Migration
- Troubleshooting

Overview

Migrating Sun ONE Portal Server, Secure Remote Access-related data from 3.0 to 6.2 involves running the Sun ONE Portal Server 3.0 Data Migration Tool Suite provided with Sun™ ONE Portal Server 6.2 and the Sun ONE Portal Server, Secure Remote Access migration scripts. If you install the Sun ONE Portal Server, Secure Remote Access migration scripts after installing the Sun ONE Portal Server 3.0 Data Migration Tool Suite, the Sun ONE Portal Server, Secure Remote Access installation of the migration tools copies the srapmigration.jar (containing the class files required for the migration) into the *BaseDir*/SUNWps/migration/modules/lib directory and creates the *BaseDir*/SUNWps/migration/modules/srap and *BaseDir*/SUNWps/migration/modules/srap_dp directories and places the modules in them.

Table 7-1 shows the file names and descriptions of the modules in the *BaseDir*/SUNWps/migration/modules/srap directory. This two-column table lists the module file name in the first column and the module description in the second column.

Table 7-1 Module File Names in the *BaseDir*/SUNWps/migration/modules/srap Directory

Module File Name	Module Description
70srapexport	Copies gateway certificates to the export directory.
70srapconvert	Converts gateway data, NetFile, and Netlet.
70srapimport	Imports gateway certificates if detected, gateway data, NetFile, and Netlet into Sun ONE Portal Server, Secure Remote Access 6.2.

Table 7-2 shows the file names and descriptions of the modules in the *BaseDir*/SUNWps/migration/modules/srap_dp directory. This two-column table lists the module file name in the first column and the module description in the second column.

Table 7-2 Module File Names in the *BaseDir*/SUNWps/migration/modules/srap_dp Directory

Module File Name	Module Description
43srapconvert	Creates display profile data for Sun ONE Portal Server, Secure Remote Access 6.2.
43srapimport	Imports display profile data created with the 43srapconvert module into Sun ONE Portal Server, Secure Remote Access 6.2.

You need to run the Sun ONE Portal Server 3.0 Data Migration Tool Suite script convertps to migrate the NetFile- and Netlet-related information.

The migration tools required for gateway migration are installed when you choose the option Install SRA Migration Tools during installation of Sun ONE Portal Server, Secure Remote Access 6.2.

Table 7-3 shows the two migration scripts and their descriptions. This two-column table shows the name of the migration script in the left column and its description in the right column.

Table 7-3 Sun ONE Portal Server, Secure Remote Access 6.2 Migration Tools

Name of Script	Description
`exportsrap`	Exports all Sun ONE Portal Server 3.0 gateway information. Use this tool only on the Sun ONE Portal Server 3.0 gateway system.
`importsrap`	Imports all exported gateway information into Sun ONE Portal Server, Secure Remote Access 6.2 in the required format. Use this tool only on the Sun ONE Portal Server 6.2 gateway system.

Gateway migration involves migrating the CA certificates from the certificate database.

You need to install the Sun ONE Portal Server, Secure Remote Access 6.2 migration tools on the Sun ONE Portal Server 3.0 gateway machine and also on the Sun ONE Portal Server, Secure Remote Access 6.2 gateway system.

Migrating the Sun ONE Portal Server, Secure Remote Access 6.2 components involves the following configurations of Sun ONE Portal Server 3.0 portal server and the gateway:

Sun ONE Portal Server 3.0 Gateway and Server on the Same Machine

This scenario requires installation of:

- Sun ONE Portal Server 3.0 Data Migration Tool Suite and Sun ONE Portal Server, Secure Remote Access migration tools on the Sun ONE Portal Server 3.0 system.
- Sun ONE Portal Server, Secure Remote Access 6.2 migration tools on the Sun ONE Portal Server, Secure Remote Access 6.2 gateway system.
- Sun ONE Portal Server 3.0 Data Migration Tool Suite and Sun ONE Portal Server, Secure Remote Access migration 6.2 tools on the Sun ONE Portal Server 6.2 system.

Sun ONE Portal Server 3.0 Gateway and Server on Different Machines

This scenario requires installation of:

- Sun ONE Portal Server, Secure Remote Access migration tools on the Sun ONE Portal Server 3.0 gateway machine.

NOTE To install only the gateway migration packages on the Sun ONE Portal Server 3.0 gateway machine, you can use the pkgadd command to add the SUNWpsmig and SUNWpsgwm packages. See "To Add the Migration Tools to the Sun ONE Portal Server 3.0 Gateway Machine."

- Sun ONE Portal Server 3.0 Data Migration Tool Suite and Sun ONE Portal Server, Secure Remote Access migration tools on the Sun ONE Portal Server 3.0 system.
- Sun ONE Portal Server 3.0 Data Migration Tool Suite and Sun ONE Portal Server, Secure Remote Access 6.2 migration tools on the Sun ONE Portal Server 6.2 system.
- Sun ONE Portal Server, Secure Remote Access 6.2 migration tools on the Sun ONE Portal Server 6.2 gateway machine.

To Add the Migration Tools to the Sun ONE Portal Server 3.0 Gateway Machine

To install only the migration tools on the Sun ONE Portal Server 3.0 Gateway Machine:

1. Insert the Java Enterprise System CD into the Sun ONE Portal Server 3.0 Gateway Machine.

2. Change directories to the directory where the SUNWpsmig and SUNWpsgwm packages are located. The packages are located in the Product/portal_srv/Packages directory. The location of the Product directory is at the same level as the installer script and depends on the platform you are using. For example, if using Solaris SPARC:

```
cd /orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc/
Product/portal_srv/Packages
```

3. Use the pkgadd command to add the two gateway migration packages.

   ```
   pkgadd SUNWpsmig
   ```

   ```
   pkgadd SUNWpsgwm
   ```

   The migration tools are installed in *BaseDir*/SUNWps/migration/

Migrating Sun ONE Portal Server 3.0 Gateway Data

Migrating the gateway data involves running the exportsrap script on the Sun ONE Portal Server 3.0 gateway machine, and the importsrap script on the Sun ONE Portal Server, Secure Remote Access 6.2 gateway system.

Run exportsrap if the gateway is on a standalone machine. Otherwise SRA Migration is an option on the export menu when you run exportps.

NOTE If you have installed the Sun ONE Portal Server, Secure Remote Access 6.2 migration tools, Netlet and NetFile are migrated as part of the Sun ONE Portal Server 3.0 Data Migration tools.

Ensure that you run the Sun ONE Portal Server 3.0 Data Migration Tool Suite and the Sun ONE Portal Server, Secure Remote Access 6.2 migration scripts for a complete gateway migration.

Migrating Gateway Data From a Stand-Alone Gateway System

Migrating the Sun ONE Portal Server 3.0 gateway data involves several steps as follows:

1. As root on the gateway system, change to the directory containing the Sun ONE Portal Server, Secure Remote Access migration tools. For example:

   cd *BaseDir***/SUNWps/migration/bin**

2. Run the `exportsrap` script. For example:

   ```
   ./exportsrap
   ```

   You see the messages similar to the following on your screen:

   ```
   Which directory should be created to store the gateway certificates?
   [/tmp/psExport]
   ```

3. Press Enter to accept the default directory `/tmp/psExport` or type in the name of another directory. You see messages similar to the following.

   If the export directory already exists, you see a message similar to the following:

   ```
   Export directory /tmp/psExport already exists.


   Delete the directory /tmp/psExport?
   ```

4. Type **`yes`** and press Enter if you want to delete the directory or type **`no`** if you do not wish to delete the directory. You see messages similar to the following:

   ```
   Found iPS version 3.0sp5

   Begin export process at Tue Sep 17 01:26:41 GMT 2003

   Error file: /tmp/psExport/logs/error.13026

   Report file: /tmp/psExport/logs/report.13026

   Copying certificates

   Successful completion of export process at Tue Sep 17 01:26:42 GMT
   2003
   ```

5. On the Sun ONE Portal Server 3.0 system, change to the directory above the `export` directory containing the data output by the export tool.

   For example, if you used the default directory (`/tmp/psExport`) for exporting the Desktop, you would type:

   ```
   cd /tmp
   ```

6. Save the export directory using the `tar` command. For example:

   ```
   tar cvf export.tar psExport
   ```

7. Use an FTP program to transfer the `export.tar` file to the Sun ONE Portal Server 6.2 gateway system.

8. Extract the files from `export.tar` using the `tar` command. For example:

   ```
   tar xvf export.tar
   ```

9. On the Sun ONE Portal Server 6.2 system, change to the migration tool directory:

   cd *BaseDir***/SUNWps/migration/bin**

10. Run the `importsrap` command. For example:

    ```
    ./importsrap
    ```

11. You see output similar to the following on your screen:

    ```
    Which directory contains the exported gateway certificates?
    [/tmp/psExport]
    ```

12. Press Enter to accept the default directory `/tmp/psExport` or type in the name of the directory containing the exported data. You see messages similar to the following:

    ```
    Begin import process at Tue Sep 17 12:16:40 IST 2003

    Error file: /tmp/psExport/logs/error.14557

    Report file: /tmp/psExport/logs/report.14557

    Gateway platform configuration file created.
    ```

Migrating Gateway Data From a System Containing the Gateway and Server

This section provides instructions for running the three migration tools that make up the Sun ONE Portal Server 3.0 Data Migration Tool Suite.

To Run the Export Tool

1. On the Sun ONE Portal Server 3.0 system, change to the migration tool directory:

   cd *BaseDir***/SUNWps/migration/bin/**

2. Run the `exportps` command:

   ./exportps [-a**] [-p** *LDAP passphrase***] [***ExportDir***]**

   Table 7-4 describes the options available for exporting gateway data and gateway certificates. This two-column table lists the options in the first column and their descriptions in the second column.

Table 7-4 Options Available for Exporting Gateway Data and Gateway Certificates

Option	Description
`-a`	Runs all modules without a menu. If you do not use the `-a` option, the tool will display an export menu listing the exportable types of data. To choose to export only gateway data and gateway certificates from the export menu, do not use the `-a` option.
`-p` *LDAP passphrase*	Sets the LDAP passphrase in order to avoid the LDAP passphrase prompt. If you do not use the `-p` *LDAP passphrase* option with the `-a` option, the tool will prompt you for the LDAP passphrase.
ExportDir	Specifies the directory where the information is to be saved. If you do not use the *ExportDir* option, the tool prompts you for the export directory.

3. The system displays message similar to the following:

   ```
   Which directory should be created to store the Portal Server system?
   [/tmp/psExport]
   ```

   You see this prompt only if you do not specify the output directory on the command line. You can choose the default directory `/tmp/psExport` or enter the export directory.

   Type the directory and press Enter.

4. The system displays a message similar to the following:

   ```
   Delete the directory /tmp/psExport?
   ```

   You see this message only if there is already exported data in the `ExportDir` directory. If you have not already exported gateway data and certificates, you may keep the export directory without risking any conflicts in the data. If you have already exported gateway data and certificates, you should delete the directory to avoid conflicts.

5. The system displays a message similar to the following:

   ```
   Found iPS version 3.0sp5

   Begin export process at  Mon Sep  9 10:30:14 PDT 2003

   Error file: /tmp/psExport/logs/error.11390

   Report file: /tmp/psExport/logs/report.11390

   Metrics file: /tmp/psExport/logs/export_metrics.11390


   Export Menu:

   1) LDAP Database

   2) Desktop

   3) Certificate Databases

   4) SRA Migration

   5) All of the above

   6) Exit

   Select one of the listed options to export:
   ```

   You see the export menu only if you do not specify the `-a` option. The SRA Migration option exports only the certificate data. When exporting the LDAP database, you must also export the Desktop.

6. Type **4** and press Enter to export the gateway.

7. The system displays the following message:

   ```
   Enter the LDAP admin passphrase :
   ```

   You see this prompt only if you select a full export using the `-a` option and you do not specify the `-p` *LDAP passphrase* option.

   Type the LDAP admin passphrase and press Enter.

   You see output similar to the following:

   ```
   Enter the LDAP admin passphrase :

   Dumping the ldap database

     organizationalPerson

     user

     role

     domain

     application

   Dumping xml

   .....................



   Copying templates and resource bundles


   Copying certificates


   Exporting gateway certificates

   Successful completion of export process at  Mon Sep  9 10:33:34 PDT 
   2003
   ```

8. Change to the export directory you specified in Step 2 or Step 3. For example:

   cd *ExportDir*

   Substitute the export directory for *ExportDir*. If you used the default export directory, for example, use /tmp/psExport.

   Type **ls** to see the directories created by the export tool.

   Table 7-5 shows the directory created by the export tool when exporting gateway data and gateway certificates. This two-column table lists the directory in the first column and its description in the second column.

Table 7-5 Directory Created by the Export Tool When Exporting Gateway Data and Gateway Certificates

Directory	Description
ExportDir/srap	Contains the gatecert subdirectory where exported gateway certificates are stored.

   After the export has completed, or if you run into problems, check the *ExportDir*/logs directory for the report.PID, error.PID, and export_metrics.PID files (where PID is the process ID). The report.PID file contains actions the export tool has or has not taken. The error.PID file contains warnings or errors so that you can correct any problems. The export_metrics.PID file contains metrics detailing when various elements of the export tool started, stopped, and the total time it took to export the data. You can look at the stdout header to know which log and report files to examine for the exportps command you are running.

9. Examine the exported data and verify all instances of the name of the Sun ONE Portal Server 3.0 system have been changed to the name of the Sun ONE Portal Server 6.2 system. Although the tool should pick up these values from the Sun ONE Portal Server 6.2 packages, you must still verify these values and manually change, if necessary before converting the data. If the port numbers on the two systems are different, you must verify and change these manually if necessary as well.

   If you are performing a single-system migration, server names will be the same, but port numbers will be different. Again, you need to verify and change port numbers manually if necessary before converting the data.

The migration tools do not modify server names and port numbers.

For example, in the following extract of the exported data in the *ExportDir*/`config/platform.config` file, you need to change manually all instances of `host1.siroe.com`, which is the name of the Sun ONE Portal Server 3.0 system, to the name of the Sun ONE Portal Server 6.2 system if the systems use different names. If the systems use different port numbers, you need to change them as well.

```
ips.defaultDomain=siroe.com

ips.server.protocol=http

ips.server.host=host1.siroe.com

ips.server.port=8080

ips.profile.host=host1.siroe.com

ips.profile.port=8080

ips.gateway.protocol=https

ips.gateway.host=host1.siroe.com

ips.gateway.port=443

ips.gateway.trust_all_server_certs=false

ips.virtualhost=host1.siroe.com 192.18.66.42

ips.naming.url=http://host1.siroe.com:8080/namingservice

ips.notification.url=http://host1.siroe.com:8080/
notificationservice
```

To Run the Conversion Tool

1. On the Sun ONE Portal Server 3.0 system, change to the directory above the `export` directory containing the data output by the export tool.

   For example, if you used the default directory (`/tmp/psExport`) for exporting the gateway data and gateway certificates, you would type:

   `cd /tmp`

2. Save the export directory using the `tar` command. For example:

   `tar cvf export.tar psExport`

3. Use an FTP program to transfer the `export.tar` file to the Sun ONE Portal Server 6.2 system.

4. Extract the files from `export.tar` using the `tar` command. For example:

```
tar xvf export.tar
```

5. On the Sun ONE Portal Server 6.2 system, change to the migration tool directory:

```
cd BaseDir/SUNWps/migration/bin
```

6. Run the `convertps` command:

```
./convertps [-a] [-f] [-i ExportDir] [-o ImportDir]
```

Table 7-6 describes the options available for the `convertps` command. This two-column table lists the options in the first column and their descriptions in the second column.

Table 7-6 Options Available for Converting Gateway Data and Gateway Certificates

Option	Description
`-a`	Runs all modules without a menu. If you do not use the `-a` option, the tool will display a convert menu listing the convertible types of data. To choose to convert only gateway data and gateway certificates from the conversion menu, do not use the `-a` option.
`-f`	Converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server roles and places users under the organization. This option retains the multiple role to user assignment similar to Sun ONE Portal Server 3.0, but does not retain the hierarchical role functionality. All roles are created under the organization and are not prioritized. The display profile documents are prioritized and merged accordingly. To see if the roles have been migrated to roles, view the roles under the organization in the admin console. If you do not use the `-f` option, the conversion tool converts Sun ONE Portal Server 3.0 roles to Sun ONE Identity Server suborganizations and places users under the suborganization. Without the `-f` option, the conversion tool retains the hierarchical functionality and customizations from Sun ONE Portal Server 3.0. The disadvantage is that it is difficult to move users from one suborganization to another. To see if the roles have been migrated to suborganizations, view the hierarchy in the admin console. The Sun ONE Portal Server, Secure Remote Access 6.2 migration tools do not support flat role migration. If you do not use this option when migrating SRA data, you should also not use the `-f` option when migrating Desktop data also.

Table 7-6 Options Available for Converting Gateway Data and Gateway Certificates

Option	Description
`-i` *ExportDir*	Specifies the input directory for the conversion. The input directory is the directory which contains the exported data created by the export tool. The conversion tool searches for export data in `/tmp/psExport` unless you specify a different input directory using the `-i` *ExportDir* option. If you specify a directory which does not contain data exported with the export tool, the conversion tool notifies you that the directory does not have export data and prompts you to enter a valid export directory.
`-o` *ImportDir*	Specifies the output directory for the conversion tool. The output directory is the directory used by the import tool. If you choose an import directory which already contains converted data, the conversion tool notifies you that modifying an existing migration may render import data unusable and prompts you to delete the import directory.

7. The system displays the following messages:

```
Found Portal Server version 6.2
```

```
Enter Identity Server Internal LDAP Authentication User password:
```

Type in a valid password.

```
Which directory should be created to store the converted data?
[/tmp/psImport]
```

You see this prompt only if you do not specify the output directory using the `-o` *ImportDir* option. You can choose the default directory or enter another import directory.

```
Import directory /tmp/psImport already exists.
```

```
If you do not wish to overwrite data within this directory, please
exit this migration process and rename the directory.
```

```
Delete the directory /tmp/psImport?
```

You see the message about an existing import directory only if the import directory already contains converted data. If you have already converted the gateway data and gateway certificates, you should exit this migration process by pressing `Ctrl-C` and rename the directory before converting the data.

8. After you choose the output directory, you see output similar to the following:

```
Begin conversion process at Tue Sep 17 15:41:55 SGT 2003
Error file: /tmp/psImport/logs/error.27888
Report file: /tmp/psImport/logs/report.27888
Metric file: /tmp/psImport/logs/convert_metrics.27888

Conversion Menu
1) LDAP Database
2) SRA Components Desktop
3) Gateway Rules to Rewriter Rules
4) Desktop
5) Certificate Databases
6) SRA Migration
7) All of the above
8) Exit
Select one of the listed options to convert:
```

You see the conversion menu only if you do not specify the `-a` option.

9. Type **2** and press Enter to convert only secure remote access components.

   The system displays messages similar to the following:

   ```
   *** Extracting templates ***


   *****

   10.Begin SRA conversion process at Tue Sep 17 15:42:55 SGT 2003


   *** Creating SRA display profile ***

   Converting channel display profile.

   Converting domain desktop preferences.

   Converting role desktop preferences.

   Converting user desktop preferences.

   End SRA conversion process at Tue Sep 17 15:43:05 SGT 2003

   Successful completion of conversion process at Mon Sep  9 12:20:01
   PDT 2003
   ```

10. Change to the import directory making sure to substitute the import directory that you selected in Step 7. For example:

    cd *ImportDir*

    Substitute the import directory for *ImportDir*. If you used the default import directory, for example, use `/tmp/psImport`.

11. Type **`ls`** to see the directories created by the conversion tool.

    Table 7-7 shows the directories created by the conversion tool when migrating the gateway. This two-column table lists the directories in the first column and their descriptions in the second column.

Table 7-7 Directories Created by the Conversion Tool When Converting Gateway Data

Directory	Description
ImportDir/`logs`	Contains `error.PID`, `export_metrics.PID`, and `report.PID` files (where `PID` is the process ID)
ImportDir/`srap`	Contains subdirectories for gateway-related desktop data such as gateway certificates, Netlet, NetFile's user, role, and organization attributes, and desktop display profile XML files.

After the conversion has completed, or if you run into problems, check the *ImportDir*/`logs` directory for the `report.PID`, `error.PID`, and `convert_metrics.PID` files (where `PID` is the process ID). The `report.PID` file contains actions the conversion tool has or has not taken and also alerts you to any customizations which you need to convert manually. The `error.PID` file contains warnings or errors so that you can correct any problems. The `convert_metrics.PID` file contains metrics detailing when various elements of the conversion tool started, stopped, and the total time it took to convert the data. You can look at the `stdout` header to know which log and report files to examine for the `convertps` command you are running.

If you run the `convertps` tool again in order to convert data other than certificate databases, the tool will prompt you to delete the import directory. Type no if you wish to keep the certificate databases you have already converted.

To Run the Conversion Tool to Convert Display Profile Data

1. On the Sun ONE Portal Server 6.2 system, change to the migration tool directory:

   ```
   cd BaseDir/SUNWps/migration/bin
   ```

2. Run the `convertps` command:

   ```
   ./convertps [-a] [-f] [-i ExportDir] [-o ImportDir]
   ```

3. The system displays the following messages:

```
Found Portal Server version 6.2
```

```
Enter Identity Server Internal LDAP Authentication User password:
```

Type in a valid password.

```
Which directory should be created to store the converted data?
[/tmp/psImport]
```

You see this prompt only if you do not specify the output directory using the `-o` *ImportDir* option. You can choose the default directory or enter another import directory.

```
Import directory /tmp/psImport already exists.
```

```
If you do not wish to overwrite data within this directory, please
exit this migration process and rename the directory.
```

```
Delete the directory /tmp/psImport?
```

Do not delete the directory.

4. After you choose the output directory, you see output similar to the following:

```
Begin conversion process at Mon Sep  9 12:16:38 PDT 2003
Error file: /tmp/psImport/logs/error.19178
Report file: /tmp/psImport/logs/report.19178
Metric file: /tmp/psImport/logs/convert_metrics.19178

Conversion Menu
1) LDAP Database
2) SRA Components Desktop
3) Gateway Rules to Rewriter Rules
4) Desktop
5) Certificate Databases
6) SRA Migration
7) All of the above
8) Exit
Select one of the listed options to convert:
```

You see the conversion menu only if you do not specify the `-a` option.

5. Type **6** and press Enter to perform only NetFile, Netlet, and gateway conversion. The system displays messages similar to the following:

```
*** Extracting templates ***

10.Begin SRA conversion process at Tue Sep 17 15:45:05 SGT 2003

Migrating SrapAttrs

Outputfile : /tmp/psImport/srap/gateway/ldif/user.ldif Inputfile :
/tmp/psExport/ldif/user.ldifdoConvert : 7

Converting LDIF entries corr. to users from
file:/tmp/psExport/ldif/user.ldif.....



Processing dn :
uid=authentication,ou=People,o=AdminRole,o=siroe.com,o=isp , org -
,o=AdminRole,o=siroe.com,o=isp

Processing dn :
uid=gateway-jughead.siroe.com,ou=People,o=defaultRole,o=siroe.com,o
=isp , org - ,o=defaultRole,o=siroe.com,o=isp

Processing dn :
uid=gateway-default,ou=People,o=defaultRole,o=siroe.com,o=isp , org
- ,o=defaultRole,o=siroe.com,o=isp

Processing dn :
uid=gateway-archie.siroe.com,ou=People,o=defaultRole,o=siroe.com,o=
isp , org - ,o=defaultRole,o=siroe.com,o=isp

/etc/opt/SUNWips/platform.conf (No such file or directory)

Processing dn : uid=root,ou=People,o=AdminRole,o=siroe.com,o=isp ,
org - ,o=AdminRole,o=siroe.com,o=isp

Processing dn : uid=vicky,ou=People,o=AdminRole,o=siroe.com,o=isp ,
org - ,o=AdminRole,o=siroe.com,o=isp

Processing dn :
uid=netfileuser1,ou=People,o=defaultRole,o=siroe.com,o=isp , org -
,o=defaultRole,o=siroe.com,o=isp

Processing dn :
uid=netfileuser2,ou=People,o=defaultRole,o=siroe.com,o=isp , org -
,o=defaultRole,o=siroe.com,o=isp

Processing dn :
```

```
uid=netfileuser3,ou=People,o=defaultRole,o=siroe.com,o=isp , org -
,o=defaultRole,o=siroe.com,o=isp

Processing dn :
uid=netfileuser4,ou=People,o=defaultRole,o=siroe.com,o=isp , org -
,o=defaultRole,o=siroe.com,o=isp

Processing dn :
uid=netletuser1,ou=People,o=defaultRole,o=siroe.com,o=isp , org -
,o=defaultRole,o=siroe.com,o=isp

Processing dn :
uid=netletuser2,ou=People,o=defaultRole,o=siroe.com,o=isp , org -
,o=defaultRole,o=siroe.com,o=isp

Processing dn :
uid=netletuser3,ou=People,o=defaultRole,o=siroe.com,o=isp , org -
,o=defaultRole,o=siroe.com,o=isp

Processing dn :
uid=netletuser4,ou=People,o=defaultRole,o=siroe.com,o=isp , org -
,o=defaultRole,o=siroe.com,o=isp

Processing dn :
uid=netletuser,ou=People,o=defaultRole,o=siroe.com,o=isp , org -
,o=defaultRole,o=siroe.com,o=isp

Processed 15 entries

Output available in file /tmp/psImport/srap/gateway/ldif/user.ldif

Converting gateway data

Converting LDIF entries corr. to users from
file:/tmp/psExport/ldif/domain.ldif.....



Output available in file
/tmp/psImport/srap/gateway/xml/GatewayOrgAttrs.xml

Converting LDIF entries corr. to users from
file:/tmp/psExport/ldif/role.ldif.....



Output available in file
/tmp/psImport/srap/gateway/xml/GatewayRoleAttrs.xml

  End gateway conversion process at Tue Sep 17 15:45:18 SGT 2003
```

```
Converting netlet data

Converting LDIF entries corr. to users from
file:/tmp/psExport/ldif/domain.ldif.....


Output available in file
/tmp/psImport/srap/netlet/xml/NetletOrgAttrs.xml

Converting LDIF entries corr. to users from
file:/tmp/psExport/ldif/role.ldif.....


Output available in file
/tmp/psImport/srap/netlet/xml/NetletRoleAttrs.xml

End netlet conversion process at Tue Sep 17 15:45:19 SGT 2003

Converting netfile data

Converting LDIF entries corr. to users from
file:/tmp/psExport/ldif/domain.ldif.....


Output available in file
/tmp/psImport/srap/netfile/xml/NetfileOrgAttrs.xml

Converting LDIF entries corr. to users from
file:/tmp/psExport/ldif/role.ldif.....


Output available in file
/tmp/psImport/srap/netfile/xml/NetfileRoleAttrs.xml

End netfile conversion process at Tue Sep 17 15:45:20 SGT 2003

Certificates copied

Successful completion of conversion process at Tue Sep 17 15:45:21
SGT 2003
```

6. Change to the import directory making sure to substitute the import directory that you selected in Step 7. For example:

   cd *ImportDir*

   Substitute the import directory for *ImportDir*. If you used the default import directory, for example, use `/tmp/psImport`.

7. Type **ls** to see the directories created by the conversion tool.

   Table 7-8 shows the directories created by the conversion tool when converting gateway data, including NetFile, and Netlet. This two-column table lists the directories in the first column and their descriptions in the second column.

Table 7-8 Directories Created by the Conversion Tool When Converting Gateway Data

Directory	Description
ImportDir/logs	Contains error.PID, export_metrics.PID, and report.PID files (where PID is the process ID)
ImportDir/srap	Contains subdirectories for gateway-related desktop data such as gateway certificates, Netlet, NetFile's user, role, and organization attributes and desktop display profile XML files.

After the conversion has completed, or if you run into problems, check the *ImportDir*/logs directory for the report.PID, error.PID, and convert_metrics.PID files (where PID is the process ID). The report.PID file contains actions the conversion tool has or has not taken and also alerts you to any customizations which you need to convert manually. The error.PID file contains warnings or errors so that you can correct any problems. The convert_metrics.PID file contains metrics detailing when various elements of the conversion tool started, stopped, and the total time it took to convert the data. You can look at the stdout header to know which log and report files to examine for the convertps command you are running.

To Run the Import Tool

1. On the Sun ONE Portal Server 6.2 system, change to the import tool directory. For example:

   cd *BaseDir*/**SUNWps/migration/bin**

2. Run the importps command:

   ./importps [-a] [-k] [-m] [*ImportDir***]**

   Table 7-9 describes the options available for importing secure remote access data. This two-column table lists the options in the first column and their descriptions in the second column.

Table 7-9 Options Available for Importing Secure Remote Access Data

Option	Description
`-a`	Runs all modules without a menu. If you do not use the `-a` option, the tool will display an import menu listing the importable types of data. To choose to import only the secure remote access data from the import menu, do not use the `-a` option.
`-k`	Specifies that the tool does not overwrite existing users. When you run `importps` without the `-k` option, if it encounters a conflict with an existing user, it will delete the existing user and import the new user entry defined in *ImportDir*`/ldap/user.ldif`. If you run `importps -k`, if it encounters a conflict with an existing user, it leaves the existing user and places the rejected user entry in *ImportDir*`/ldap/rejected_users.ldif`. If the user has a display profile document in the *ImportDir*`/dp/user/` directory, the display profile file may change. The `-k` option does not affect importing user display profile documents. The `-k` option only affects whether an existing LDAP user's LDIF record is updated. The updates to the LDIF record include everything except display profile updates.
`-m`	Merges display profile documents. When you run `importps` without the `-m` option, it overwrites any existing display profile documents it finds. This means that the root suffix, organization, suborganization, and user level display profile documents, if they exist, are overwritten. If you run `importps -m`, when the tool encounters a conflict with an existing display profile document it will merge it.
ImportDir	Specifies the directory where the converted data is located. If the directory does not exist or if the directory does not contain valid converted data, the import tool prompts you to enter a directory containing the converted data to be imported. If you do not specify an import directory, the import tool will search `/tmp/psImport`.

3. The system displays a message similar to the following:

   ```
   Found Portal Server version 6.2

   Enter Identity Server Internal LDAP Authentication User password:
   ```

   Type in a valid password.

   ```
   Enter Appserver Administrator password.
   ```

   You see this prompt only if you are migrating onto a Sun ONE Application Server web container.

   ```
   Begin import process at Tue Sep 17 16:03:22 SGT 2003

   Error file: /tmp/psImport/logs/importerror.3127

   Report file: /tmp/psImport/logs/importreport.3127

   Metric file: /tmp/psImport/logs/import_metrics.3127


   Import Menu:

   1) LDAP Database

   2) SRA Components Desktop

   3) Rewriter Rules

   4) Desktop

   5) Certificate Databases

   6) SRA Migration

   7) All of the above

   8) Exit

   Select one of the listed options to import:
   ```

   You see the import menu only if you do not specify the -a option.

4. Type **2** and press Enter to import only the secure remote access components Desktop.

NOTE Do not select option 2 before option 1. If you have not yet migrated LDAP data, follow the instructions in Chapter 2, “Migrating LDAP Data” before selecting option 2 to import `SRA Components Desktop`

The system displays messages similar to the following:

```
Importing dp xml

Adding Netlet provider.

ERROR: Duplicate name found.  Use a different name.

[DEBUG] Line 1: modify -u uid=amAdmin,ou=People,o=siroe.com,o=isp -w
hanates

t -d o=siroe.com,o=isp -m
/tmp/psImport/srap/desktop/dp/initialOrganization.

xml


SUCCESS!

[DEBUG] Line 3: modify -u uid=amAdmin,ou=People,o=siroe.com,o=isp -w
passphrase -d o=siroe.com,o=isp -m
/tmp/psImport/srap/desktop/dp/initialOrganization.xml


SUCCESS!

[DEBUG] Line 5: modify -u uid=amAdmin,ou=People,o=siroe.com,o=isp -w
passphrase -d o=dog,o=isp -m
/tmp/psImport/srap/desktop/dp/initialOrganization.xml


SUCCESS!

[DEBUG] Line 7: modify -u uid=amAdmin,ou=People,o=siroe.com,o=isp -w
passphrase -d o=NetFile-Java2dom,o=isp -m
/tmp/psImport/srap/destop/dp/initialOrganization.xml
```

```
SUCCESS!

[DEBUG] Line 9: modify -u
uid=amAdmin,ou=People,o=siroe.com,o=isp -w passphrase -d
o=nfj1,o=isp -m
/tmp/psImport/srap/desktop/dp/initialOrganization.xml


SUCCESS!

[DEBUG] Line 11: modify -u uid=amAdmin,ou=People,o=siroe.com,o=isp
-w passphrase -d o=nfj11,o=isp -m
/tmp/psImport/srap/desktop/dp/initialOrganization.xml


SUCCESS!

[DEBUG] Line 13: modify -u uid=amAdmin,ou=People,o=siroe.com,o=isp
-w passphrase -d o=vrush,o=isp -m
/tmp/psImport/srap/desktop/dp/initialOrganization.xml


SUCCESS!

[DEBUG] Line 15: modify -u uid=amAdmin,ou=People,o=siroe.com,o=isp
-w passphrase -d o=netlet,o=isp -m
/tmp/psImport/srap/desktop/dp/initialOrganization.xml


SUCCESS!

[DEBUG] Line 17: modify -u uid=amAdmin,ou=People,o=siroe.com,o=isp
-w passphrase -d o=murali,o=isp -m
/tmp/psImport/srap/desktop/dp/initialOrganization.xml


SUCCESS!

[DEBUG] Line 19: modify -u uid=amAdmin,ou=People,o=siroe.com,o=isp
-w passphrase -d o=siroe.com,o=isp -m
/tmp/psImport/srap/desktopdp/tab/siroe.com/modify/NewsTemplatePanel
Container.xml
```

```
 SUCCESS!

Redeploying portal web application

/opt/SUNWam/servers/https-admserv
/opt/SUNWam/servers/https-host1.siroe.com

Deploying to instance host1.siroe.com...

Successful completion of import process at Mon Sep  17 16:05:22 SGT
2003
```

After the import has completed, or if you run into problems, check the *ImportDir*/`logs` directory for the `importreport.PID`, `importerror.PID`, and `import_metrics.PID` files (where `PID` is the process ID). The `importreport.PID` file contains actions the import tool has or has not taken and also alerts you to any customizations which you need to import manually. The `importerror.PID` file contains warnings or errors so that you can correct any problems. The `import_metrics.PID` file contains metrics detailing when various elements of the import tool started, stopped, and the total time it took to import the data. You can look at the `stdout` header to know which log and report files to examine for the `importps` command you are running.

To Run the Import Tool to Import Display Profile Data

1. On the Sun ONE Portal Server 6.2 system, change to the import tool directory. For example:

   `cd` *BaseDir***`/SUNWps/migration/bin`**

2. Run the `importps` command:

   `./importps [-a] [-k] [-m] [`*ImportDir***`]`**

   Table 7-10 describes the options available for importing secure remote access data. This two-column table lists the options in the first column and their descriptions in the second column.

Table 7-10 Options Available for Importing Secure Remote Access Data

Option	Description
`-a`	Runs all modules without a menu. To choose to import only the secure remote access data from the import menu, do not use the `-a` option.

Table 7-10 Options Available for Importing Secure Remote Access Data *(Continued)*

Option	Description
`-k`	Specifies that the tool does not overwrite existing users. When you run `importps` without the `-k` option, if it encounters a conflict with an existing user, it will delete the existing user and import the new user entry defined in *ImportDir*`/ldap/user.ldif`. If you run `importps -k`, if it encounters a conflict with an existing user, it leaves the existing user and places the rejected user entry in *ImportDir*`/ldap/rejected_users.ldif`. If the user has a display profile document in the *ImportDir*`/dp/user/` directory, the display profile file may change. The `-k` option does not affect importing user display profile documents. The `-k` option only affects whether an existing LDAP user's LDIF record is updated. The updates to the LDIF record include everything except display profile updates.
`-m`	Merges display profile documents. When you run `importps` without the `-m` option, it overwrites any existing display profile documents it finds. This means that the root suffix, organization, suborganization, and user level display profile documents, if they exist, are overwritten. If you run `importps -m`, when the tool encounters a conflict with an existing display profile document it will merge it.
ImportDir	Specifies the directory where the converted data is located. If the directory does not exist or if the directory does not contain valid converted data, the import tool prompts you to enter a directory containing the converted data to be imported. If you do not specify an import directory, the import tool will search `/tmp/psImport`.

3. The system displays messages similar to the following:

   ```
   Found Portal Server version 6.2

   Enter Identity Server Internal LDAP Authentication User password:
   ```

   Type in a valid password.

   ```
   Enter Appserver Administrator password.
   ```

   You see this prompt only if you are migrating onto a Sun ONE Application Server web container.

   ```
   Begin import process at Thu Sep  19 13:08:09 IDT 2003

   Error file: /tmp/psImport/logs/importerror.19657

   Report file: /tmp/psImport/logs/importreport.19657

   Metric file: /tmp/psImport/logs/import_metrics.19657


   Import Menu:

   1) LDAP Database

   2) SRA Components Desktop

   3) Rewriter Rules

   4) Desktop

   5) Certificate Databases

   6) SRA Migration

   7) All of the above

   8) Exit

   Select one of the listed options to import:
   ```

   You see the import menu only if you do not specify the `-a` option.

4. Type **6** and press Enter to import only the secure remote access data.

   The system displays messages similar to the following:

```
Success 0: Successfully completed.

Success 0: Successfully completed.

Lowering the priorities of the assigned Netlet policies.

Netlet data migrated.

Success 0: Successfully completed.

Error 10: Cannot process requests: null

Lowering the priorities of the assigned Netfile policies.

Adding netfile channel to desktop.

Success 0: Successfully completed.

Success 0: Successfully completed.

Success 0: Successfully completed.

Netfile data migrated.

Adding srapGatewayAccessService to required services list.

Info 112: Entering ldapAuthenticate method!

Success 0: No Exceptions! LDAP Authentication success!

Info 113: Calling SSO method!

Info 107: Calling XML PARSER

Info 108: XML file to
parse:/var/opt/SUNWps/tmp/migSUNWpsgws_requests.xml

Info 101: Processing /var/opt/SUNWps/tmp/migSUNWpsgws_requests.xml

Info 111: Requests generated by amadmin

Info 102: Done processing
/var/opt/SUNWps/tmp/migSUNWpsgws_requests.xml

Success 0: Successfully completed.

Importing gateway data.

modifying entry
uid=authentication,ou=People,o=AdminRole,o=siroe.com,o=isp
```

```
modifying entry
uid=authentication,ou=People,o=AdminRole,o=siroe.com,o=isp


modifying entry
uid=gateway-host1.siroe.com,ou=People,o=defaultRole,o=siroe.com,o=i
sp


modifying entry
uid=gateway-host1.siroe.com,ou=People,o=defaultRole,o=siroe.com,o=i
sp


modifying entry
uid=gateway-default,ou=People,o=defaultRole,o=siroe.com,o=isp


modifying entry
uid=gateway-default,ou=People,o=defaultRole,o=siroe.com,o=isp


modifying entry
uid=gateway-host1.siroe.com,ou=People,o=defaultRole,o=siroe.com,o=i
sp


modifying entry
uid=gateway-host1.siroe.com,ou=People,o=defaultRole,o=siroe.com,o=i
sp


modifying entry uid=root,ou=People,o=AdminRole,o=siroe.com,o=isp


modifying entry uid=root,ou=People,o=AdminRole,o=siroe.com,o=isp


modifying entry uid=root,ou=People,o=AdminRole,o=siroe.com,o=isp
modifying entry uid=root,ou=People,o=AdminRole,o=siroe.com,o=isp


modifying entry uid=vicky,ou=People,o=AdminRole,o=siroe.com,o=isp


modifying entry uid=vicky,ou=People,o=AdminRole,o=siroe.com,o=isp


modifying entry
uid=netfileuser1,ou=People,o=defaultRole,o=siroe.com,o=isp

Success 0: Successfully completed.

Error 10: Cannot process requests: null

Success 0: Successfully completed.

Gateway data migrated.

Redeploying portal web application

/opt/SUNWam/servers/https-admserv
/opt/SUNWam/servers/https-host1.siroe.com

Deploying to instance host1.siroe.com...

Successful completion of import process at Thu Sep 19 18:41:47 IST
2003
```

After the import has completed, or if you run into problems, check the *ImportDir*/`logs` directory for the `importreport.PID`, `importerror.PID`, and `import_metrics.PID` files (where `PID` is the process ID). The `importreport.PID` file contains actions the import tool has or has not taken and also alerts you to any customizations which you need to import manually. The `importerror.PID` file contains warnings or errors so that you can correct any problems. The `import_metrics.PID` file contains metrics detailing when various elements of the import tool started, stopped, and the total time it took to import the data. You can look at the `stdout` header to know which log and report files to examine for the `importps` command you are running.

Verifying the Migration

After the gateway migration is complete, verify that

- The certificates are migrated under the instance name `migrated` (`/etc/opt/SUNWps/platform.conf.migrated`).
- The Sun ONE Portal Server 3.0 gateway profile is migrated as the gateway profile instance `migrated`.
- The gateway, NetFile, and Netlet services are assigned to the newly created or migrated organizations, assuming that these packages have been installed.

  These services are assigned to all organizations, suborganizations, and users, but the users who did not have access will be put under a role restricting their access to these services.
- The policy to enable the gateway, NetFile, and Netlet services is assigned to all new organizations and suborganizations.

Troubleshooting

Deleting Extra NetFile Links from the Desktop

After migration, there will be two sets of entries in the applications channel. They are the old Desktop migrated entries for NetFile lite and NetFile which you must remove because they point to non-existent URLs. There are also links to NetFile Java1 and NetFile Java2 which are not selected by default.

Delete the two NetFile links that appear on the Desktop after migration. Use the following URLs to access NetFile Java1 and NetFile Java2:

For NetFile Java1:

```
https://gateway_machine:port/http://server:port/portal/
NetFileApplet?Refer=java1
```

For NetFile Java2:

```
https://gateway_machine:port/http://server:port/portal/
NetFileApplet?Refer=java2
```

Adding the MIME Type Entries to the web.xml File

The data migration process removes the MIME type entries from the `web.xml` file. To restore Netlet and NetFile functionality after the data migration, the Mime type entries need to be added to the `web.xml` file in the web container directory, for example, *websphere-install-dir*`/WebSphere/AppServer/installedApps/portal.ear/portal.war/WEB-INF/web.xml`.

1. Add the following lines after the `session-config` tag in the `web.xml` file:

```
<mime-mapping>
    <extension>jar</extension>
    <mime-type>application/octet-stream</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>cab</extension>
    <mime-type>application/octet-stream</mime-type>
</mime-mapping>
```

2. Restart the Sun ONE Portal Server and the gateway.

Chapter 8

Migrating Sun™ ONE Compass Server

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

Sun™ ONE Portal Server 6.2 includes an integrated Search Engine that provides the functionality that Sun™ ONE Compass Server provided for Sun ONE Portal Server 3.0. Sun ONE Compass Server migration uses a separate set of tools from the Sun ONE Portal Server 3.0 Data Migration Tool Suite. The Sun ONE Compass Server migration tools include an export tool and an import tool that migrate Sun™ ONE Compass Server 3.01C, Service Pack 1 or later to Sun ONE Portal Server 6.2. Unlike the Sun ONE Portal Server 3.0 Data Migration Tool Suite, the Sun ONE Compass Server migration tools do not include a conversion tool because the conversion step is not necessary for Sun ONE Compass Server migration.

This chapter discusses migration from Sun ONE Compass Server 3.01C, Service Pack 1 or later to Sun ONE Portal Server 6.2. If you are running an earlier version of Sun ONE Compass Server, you must upgrade to Sun ONE Compass Server 3.01C, Service Pack 1 or later in order to migrate Sun ONE Compass Server to Sun ONE Portal Server 6.2. The Sun ONE Compass Server migration tools are *BaseDir*`/SUNWips/migration/compass/export.pl` and *BaseDir*`/SUNWips/migration/compass/import.pl`.

This chapter contains the following sections:

- What You Need to Know Before Beginning the Sun ONE Compass Server Migration Process
- Migrating the Personal Interest Profile (PIP)

- Using the Sun ONE Compass Server Export Tool
- Using the Sun ONE Compass Server Import Tool

What You Need to Know Before Beginning the Sun ONE Compass Server Migration Process

Use the following information to help plan the Sun ONE Compass Server migration process.

- The Sun ONE Compass Server migration tools migrate the Sun ONE Compass Server 3.01C, Service Pack 1 or later instance to the Sun ONE Portal Server 6.2 system.
- Sun ONE Compass Server migration tools are separate from the Sun ONE Portal Server 3.0 Data Migration Tool Suite.
- The Sun ONE Compass Server migration tools enable you to migrate only one instance of Sun ONE Compass Server 3.01C, Service Pack 1 or later at a time. Note that you can run the Sun ONE Compass Server migration tools once for the Sun™ ONE Web Server installation and the tools will migrate all instances of Sun ONE Compass Server 3.01C, Service Pack 1 or later within Sun ONE Web Server.
- You need to copy or move the exported data from the Sun ONE Compass Server 3.01C, Service Pack 1 or later system to the Sun ONE Portal Server 6.2 system.
- The existing Sun ONE Compass Server provider is not migrated because the Search provider in Sun ONE Portal Server 6.2 is more tightly integrated with Sun ONE Portal Server 6.2 than Sun ONE Compass Server 3.01C, Service Pack 1 or later was with Sun ONE Portal Server 3.0.
- You must run the Sun ONE Compass Server migration tools as root. For non-root configuration, perform the Compass migration as root and see the *Sun ONE Portal Server* 6.2 *Installation Guide* for information on non-root configuration.
- You must install Sun ONE Portal Server 6.2 on a new system before migrating Sun ONE Compass Server 3.01C, Service Pack 1 or later to Sun ONE Portal Server 6.2.

- You must set up and properly configure the Search Engine on the Sun ONE Portal Server 6.2 system either with or without new documents before migrating Sun ONE Compass Server 3.01C, Service Pack 1 or later to Sun ONE Portal Server 6.2.
- The Sun ONE Compass Server migration tools add migrated documents to the Search Engine on the Sun ONE Portal Server 6.2 system.
- The Sun ONE Compass Server migration tools replace new Search Engine configuration data with the migrated Sun ONE Compass Server 3.01C, Service Pack 1 or later configuration data.
- The Sun ONE Compass Server migration tools copy over all files under the robot directory.
- The Sun ONE Compass Server migration tools copy over Sun ONE Compass Server 3.01C, Service Pack 1 or later taxonomy and reindexes it.
- The Sun ONE Compass Server migration tools import Sun ONE Compass Server 3.01C, Service Pack 1 or later documents which they dump in advance and reindex.

Migrating the Personal Interest Profile (PIP)

If you plan to migrate the Personal Interest Profile (PIP) to the new iPlanet Portal Server 6.2 subscription service, read the following additional steps. Some planning is required before the migration process.

The user entries in the old system do not necessarily reside in the same directory server as the new iPlanet Portal Server 6.2. Therefore, the user entries might not be the same after the migration. The administrator planning the migration should plan the mapping in advance. The import process for the PIP is a separate step to reflect this requirement. This is especially true for Sun ONE Compass Server 3.01b where user entries are possibly stored in the local LDAP database which is no longer supported.

To ease this transition, the `export.pl` tool includes parameters to look up the new entries in the Sun ONE Directory Server where iPlanet Portal Server 6.2 is used. For example:

./export.pl [-h *hostname***] [-p** *port***] [-D** *bind_dn***] [-w** *passphrase***] [-b** *base_dn***]**

Table 8-1 describes the parameters used with the Sun ONE Compass Server export tool. This two-column table lists the parameters in the first column and their descriptions in the second column.

Table 8-1 Parmeters Used With Compass Server Export Tool

Parameter	Description
`-h` *hostname*	Specifies the host name.
`-p` *port*	Specifies the port.
`-D` *bind_dn*	Specifies the admin ID in the directory server.
`-w` *passphrase*	Specifies the passphrase.
`-b` *base_dn*	Specifies the Sun ONE Portal Server install base.

The migration program will get the old user id from the existing system. Then it uses above information, if supplied, which is in the directory server used by the Sun ONE Portal Server 6.2, and tries to map the old user id to the new user id.

The lookup assumes the uid is the same before and after migration. This implies that user entries are migrated to the new Sun ONE Directory Server and set up as Portal users. See Sun ONE Directory Server, Sun ONE Identity Server, and Sun ONE Portal Server documentation for this process.

For entries without the same uid, or if you do not use the lookup feature, you might need to inspect the generated `PKMigrated.ldif` file manually and resolve the mapping before importing it.

The migration is not expecting the user entries in the Directory Server with the `MySubscription` service. Therefore add the corresponding `objectClasses`. In the case that users already have this service assigned, the following line should be removed in the `PKMigrated.ldif` file:

```
add: objectClass
```

```
objectClass: sunPortalPKSubscriptionsPerson
```

Using the Sun ONE Compass Server Export Tool

You use the Sun ONE Compass Server export tool to export data from Sun ONE Compass Server 3.01C, Service Pack 1 or later to the Sun ONE Portal Server 6.2 system. You can run the Sun ONE Compass Server export tool once for each instance of Sun ONE Web Server containing the Sun ONE Compass Server 3.01C, Service Pack 1 or later plug-in or you can run the export tool once for the Sun ONE Web Server installation, which will pick up and export all Sun ONE Compass Server 3.01C, Service Pack 1 or later instances within Sun ONE Web Server.

To Run the Sun ONE Compass Server Export Tool

1. On the Sun ONE Portal Server 6.2 system, change to the Sun ONE Compass Server export tool directory:

   ```
   cd BaseDir/SUNWps/migration/compass/
   ```

2. Run the `export.pl` script:

   ```
   perl ./export.pl [-o ExportDir] [-i Target_InstanceDir]
   ```

   or

   ```
   ./export.pl [-o ExportDir] [-i Target_InstanceDir]
   ```

   Table 8-2 describes the parameters used with the Sun ONE Compass Server export tool. This two-column table lists the parameters in the first column and their descriptions in the second column.

Table 8-2 Parameters used with the Sun ONE Compass Server export tool

Parameter	Description
`-o` *ExportDir*	Specifies the directory where the exported data is to be saved.
`-i` *Target_InstanceDir*	Specifies the directory for the target instance to be exported.

If you do not specify any parameters, the Sun ONE Compass Server export tool will export all instances to the current directory.

3. The system displays messages similar to the following:

```
Exporting instance https-<instance>...

Exporting database...

Dump --> tmp/migration60.soif OK

Export Config files...

Exporting instance https-<instance>

Exporting database...

Dump --> tmp/migration60.soif OK

Export Config files...

Exporting instance https-<instance>

Exporting database...

Dump --> tmp/migration60.soif OK

Export Config files...

Number of instance(s) exported: 3
```

You only need to run the Sun ONE Compass Server export tool once for all Sun ONE Compass Server 3.01C, Service Pack 1 or later instances for each Sun ONE Web Server installation. For stand-alone installations of Sun ONE Web Server, you can simply copy over the Sun ONE Compass Server export tool. The Sun ONE Compass Server export tool does not use the default directories of the Sun ONE Portal Server 3.0 Data Migration Tool Suite because of the potential large size of the document files.

The Sun ONE Compass Server export tool can take two input parameters, one for the Sun ONE Web Server installation directory and one for the export directory. The Sun ONE Compass Server export tool dumps all documents into `/tmp/migration60.soif` for each Sun ONE Compass Server 3.01C, Service Pack 1 or later instance. The default character set for the data SOIF file generated by the Sun ONE Compass Server export tool is ISO-8859-1. Other character sets require manual changes in the data file. The default character set for Sun ONE Portal Server 6.2 is UTF-8.

The Sun ONE Compass Server export tool saves the host name of the Sun ONE Compass Server 3.01C, Service Pack 1 or later, CSID, and other information in `/tmp/migration60.info` for each Sun ONE Compass Server 3.01C, Service Pack 1 or later instance and uses this information to create new corresponding instances for the Search Engine. You can modify this by hand prior to running the Sun ONE Compass Server import tool to fine tune the new setup.

NOTE Although the file names generated by the export tool are named with a 6.0 naming convention, the exported data can be used on a Portal Server 6.2 implementation as it is compatible with both Sun ONE Portal Server 6.0, 6.1, and 6.2 releases.

Table 8-3 shows manual options for fine tuning exported data prior to running the import tool. This two-column table lists the options in the first column and their descriptions in the second column.

Table 8-3 Options for Fine Tuning Exported Data

Option	Descriptions
`DoMigrate`	Specifies whether or not to import this instance
`OldServerRoot`	Enables the import tool to identify the old Sun ONE Compass Server 3.01C, Service Pack 1 or later configuration
`OldCharSet` (optional)	The Sun ONE Compass Server 3.01C, Service Pack 1 or later instance's character set. It is normally ISO-8859-1, but it may be a different. This option enables the import tool to convert the data to UTF-8 correctly.
`NewServerPort` (optional)	Reserved for future use
`NewServerUser` (optional)	Reserved for future use
`NewServerID` (optional)	Reserved for future use
`NewServerHost` (optional)	Reserved for future use

Using the Sun ONE Compass Server Import Tool

In order to import the exported data from Sun ONE Compass Server 3.01C, Service Pack 1 or later into the Sun ONE Portal Server 6.2 system, you run the Sun ONE Compass Server import tool which uploads Sun ONE Compass Server 3.01C, Service Pack 1 or later data to the Sun ONE Portal Server 6.2 system.

NOTE Stop the iPlanet Portal Server before running the `import.pl` script. Due to schema changes the search database can get corrupted if the Portal Server is not stopped before running the `import.pl` script.

To Run the Sun ONE Compass Server Import Tool

1. On the Sun ONE Portal Server 6.2 system, change to the directory containing the Sun ONE Compass Server import tool:

   cd *BaseDir*/**SUNWps/migration/compass**

2. Run the `import.pl` script:

   perl ./import.pl [**-o** *Target_InstanceDir*] [**-i** *Source_InstanceDir*] [**-p** *Portal_InstallDir*]

   or

   ./import.pl [**-o** *Target_InstanceDir*] [**-i** *Source_InstanceDir*] [**-p** *Portal_InstallDir*]

   An example might look like:

   perl ./import.pl -o /var/opt/SUNWps/https-*instance***/portal -i /***nfs*/*ExportDir*/**https-***old-instance* **-p /opt**

   or

   ./import.pl -o /var/opt/SUNWps/https-*instance***/portal -i /***nfs*/*ExportDir*/**https-***old-instance* **-p /opt**

   Table 8-4 describes the parameters used with the Sun ONE Compass Server import tool. This two-column table lists the parameters in the first column and their descriptions in the second column.

Table 8-4 Parameters used with the Sun ONE Compass Server import tool

Parameter	Description
`-o` *Target_InstanceDir*	Specifies the directory of the Sun ONE Portal Server 6.2 Search root.
`-i` *Source_InstanceDir*	Specifies the directory of the Sun ONE Compass Server 3.01C, Service Pack 1 or later instance.

Table 8-4 Parameters used with the Sun ONE Compass Server import tool

Parameter	**Description**
`-p` *Portal_InstallDir*	Specifies the directory of the Sun ONE Portal Server 6.2 installation.

3. The system displays the following message:

```
Migration Completed
```

Chapter 9

Changes in Sun ONE Portal Server APIs that Affect Migration

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

This chapter provides information on the API changes between Sun ONE Portal Server 3.0 and Sun™ ONE Portal Server 6.2.

This chapter contains the following sections:

- Changes From Sun ONE Portal Server 3.0 That Affect Application Development
- Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken
- Migrating the Sun ONE Portal Server 3.0 Profile to Sun ONE Identity Server User Management Classes and Service Management Classes
- Migrating Sun ONE Portal Server 3.0 Logging to Sun ONE Identity Server Logging APIs
- Migrating Sun ONE Portal Server 3.0 Debug to Sun ONE Identity Server Debug APIs
- Migrating Sun ONE Portal Server 3.0 Provider APIs to Sun ONE Portal Server 6.2 Provider APIs

Changes From Sun ONE Portal Server 3.0 That Affect Application Development

The following changes to Portal Server 3.0 affect portal developers:

- New data model - Sun ONE Portal Server 6.2 uses Sun™ ONE Directory Server as its storage mechanism. This replaces the Portal Server 3.0 internal role tree directory.
- The Portal Server 3.0 APIs for session, profile, authentication, and so on, have been replaced by the Sun ONE Identity Server APIs. See the *Sun ONE Portal Server 6.0 Developer's Guide* for more information.
- If you migrate from Portal Server 3.0 to Sun ONE Portal Server 6.2, the migration process also includes modifying source code to use the new Portal Server APIs. See Appendix C, "Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2" for API mappings.
- The Portal Server 3.0 Provider API (PAPI) has been updated as follows:
    - New `ProviderContext` interface.
    - New APIs for container hierarchy.
    - Compile compatibility for some providers.
    - Improved internationalization so that:
        - The display profile supports localized strings.
        - Character set encoding methods are determined using local setting and client type.

Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken

For details on API changes needed for migrating Sun ONE Portal Server 3.0 session toSun ONE Identity Server SSOToken, see Table C-1 in Appendix C, "Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2."

Migrating the Sun ONE Portal Server 3.0 Profile to Sun ONE Identity Server User Management Classes and Service Management Classes

For details on API changes needed for migrating Sun ONE Portal Server 3.0 profile to Sun ONE Identity Serveruser management classes and service management classes, see Table C-2 in Appendix C, “Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2.”

Migrating Sun ONE Portal Server 3.0 Logging to Sun ONE Identity Server Logging APIs

For details on API changes needed for migrating Sun ONE Portal Server 3.0 logging to Sun ONE Identity Server logging APIs, see Table C-3 in Appendix C, “Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2.”

Migrating Sun ONE Portal Server 3.0 Debug to Sun ONE Identity Server Debug APIs

For details on API changes needed for migrating Sun ONE Portal Server 3.0 debug to Sun ONE Identity Server debug APIs, see Table C-4 in Appendix C, “Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2.”

Migrating Sun ONE Portal Server 3.0 Provider APIs to Sun ONE Portal Server 6.2 Provider APIs

The Desktop SDK for Sun ONE Portal Server 3.0 included the following interfaces and classes:

- `Provider`
- `ProviderAdapter`
- `ProfileProviderAdapter`

- `DesktopURL`
- `InvalidEditFormDataException`
- `AsciiFormInputException`
- `LoadPropertiesException`
- `ProfileProviderException`
- `ProviderException`
- `ProviderWrapperException`
- `UnknownEditTypeException`

This section outlines the Desktop API changes between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2. For the Sun ONE Portal Server 3.0 Desktop SDK classes and methods that have changed syntactically or semantically, see Table C-5 in Appendix C, "Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2."

By far, the largest changes are in the provider interface. These changes fall into two general categories:

- Changes to facilitate the formalization of container channels
- Changes to separate access to services out of the PAPI

In Sun ONE Portal Server 3.0, there is a single implicit container for all channels. This container cannot be modified or replaced. All leaf channels can assume a single container with a particular implementation. In Sun ONE Portal Server 6.2, the idea of a container channel is formalized. The Sun ONE Portal Server 6.2 Desktop consists of a number of container channels, all with different implementations and features. In the Sun ONE Portal Server 3.0 provider interface, there is a subset of methods that assume a single container. For example, the Sun ONE Portal Server 3.0 container called the `Provider.isDetached()` method on a leaf channel to determine if the provider should be displayed in a detached window. The information describing whether the channel is detached or not is logically connected to the channel. This does not scale to an environment in which the channel's detached status will be different for every container. Specifically, the detached flag is actually a property of the container channel, not the leaf channel.

Additionally, having an `isDetached()` method in the provider interface assumes that all containers support this functionality. This is not the case. A tab container, for example, would not support detachability.

It is useful to split channel properties into two categories:

- Type 1- Properties that are the same regardless of the container in which the channel is displayed.
- Type 2: Properties that describe a relationship between a container and leaf channel.

`Provider.isDetached()` is a type 2 property since it describes a relationship between a container and leaf channel rather than being the same regardless of the container in which the channel is displayed. In Sun ONE Portal Server 6.2, all type 2 properties are owned by the container channel. In other words, they are properties of the container channel. For example, if a container supports detachability for its leaf channels, it must store a property that is a mapping between leaf channel name and detached status, for every leaf channel it contains.

The methods to access type 2 properties are therefore removed from the provider interface.

In Sun ONE Portal Server 6.2, the underlying services layer changed from using the proprietary Sun ONE Portal Server 3.0 services to Sun ONE Identity Server. Because the Sun ONE Portal Server 3.0 provider interface is tightly integrated with the Sun ONE Portal Server 3.0 service APIs, migration is required in Sun ONE Portal Server 6.2. In an attempt to prevent this problem in the future, Sun ONE Portal Server 6.2 separates service access into an agnostic interface called `ProviderContext`. The idea is that provider developers access services through `ProviderContext` and are therefore immune to changes in the underlying services implementation.

There is a subset of methods in the provider interface that are specific to the syntax and semantics of the Sun ONE Portal Server 3.0 service APIs. These have been changed to use standard JDK or Portal Server APIs. An example of this is the `Provider.init()` method. In Sun ONE Portal Server 3.0, this method took a Sun ONE Portal Server 3.0 session object as an argument. In Sun ONE Portal Server 6.2, this method takes a servlet request object from the Servlets 2.2 SDK.

Table C-5 in Appendix C, “Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2.” maps the Sun ONE Portal Server 3.0 classes and methods that have changed syntactically or semantically to the Sun ONE Portal Server 6.2 equivalents. Note that classes and methods where there is no syntactic or semantic change are excluded from the table.

Chapter 10

Enabling Logon to a Different Organization or Suborganization

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

Migration from Sun ONE Portal Server 3.0 to Sun™ ONE Portal Server 6.2 sets the core authentication service organization URL mapping in the following manner:

Organization: *organization name*, for example:

```
siroe.com
```

Suborganization: *suborganization*, *o=organization name*, for example:

```
eng,o=siroe.com
```

Role: not applicable.

After a successful migration, if you want to access the Desktop for the default organization then use:

```
http://host:port/amserver/UI/login
```

This URI will display the default organization's core authentication service authentication menu.

You can also use:

`http://`*host:port/Deploy_URI*`/dt` (for example, `http://`*host*`:`*port*`/portal/dt`)

This URI will also display the default organization's core authentication service authentication menu. *Deploy_URI* is the value entered during installation.

If Sun ONE Portal Server 3.0 has more than one domain, then the organization URL is important. For example, if your Sun ONE Portal Server 3.0 deployment uses the domains `siroe.com` and `sesta.com` then you can authenticate and see the correct Desktop after migration by using:

`http://`*host*`:`*port*`/amserver/UI/login?org=siroe.com`

or

`http://`*host*`:`*port*`/amserver/UI/login?org=sesta.com`

The URL for accessing, for example, suborganization `eng` for organization `siroe.com` would be:

`http://`*host*`:`*port*`/amserver/UI/login?org=eng,o=siroe.com`

See Chapter 9 of the *Sun ONE Identity Server Administrator's Guide* for customizations of the organization URL mapping attribute.

Part II

Upgrading from Sun Java System Portal Server 6.0 to Sun ONE Portal Server 6.2

Chapter 11

Planning the Upgrade from Sun Java System Portal Server 6.0 to Sun Java System Portal Server 6.2

This chapter provides an overview and discussion on how to plan for upgrading users, roles, organizations, components, templates, providers, channels, and properties files from Sun™ ONE Portal Server 6.0 to Sun™ ONE Portal Server 6.2.

This chapter contains the following sections:

- Overview of Upgrade Process
- Upgrading Certificates
- Upgrading Multiple Nodes

Overview of Upgrade Process

The Sun ONE Portal Server 6.2 product uses the Sun ONE Identity Server 6.1 service as its LDAP data store and management interface. The Sun ONE Portal Server 6.0 product used the Sun ONE Identity Server 5.1 service as its LDAP data store and management interface. Although the Sun ONE Identity Server 5.1 and the Sun ONE Identity Server 6.1 products are similar in structure and provide many of the same services, changes to the schema (the LDAP mechanism used to store and define service configuration data) from Sun ONE Identity Server 5.1 to Sun ONE Identity Server 6.1 require you to migrate the data in order to upgrade from the Sun ONE Portal Server 6.0 software to the Sun ONE Portal Server 6.2 software.

NOTE The Sun ONE Portal Server 6.2 product can be deployed into a variety of web containers including the Sun™ ONE Web Server, the Sun™ ONE Application Server, the IBM Websphere Application Server, and the BEA Weblogic Application Server; however, when upgrading your Sun ONE Portal Server 6.0 installation into a Sun ONE Portal Server 6.2 installation you must deploy it into the same brand of web container.

These are the high-level steps to upgrade from Sun ONE Portal Server 6.0 to Sun ONE Portal Server 6.2. (Subsequent chapters provide step-by-step instructions for performing the actual processes.)

1. Before you start the Sun ONE Portal Server upgrade process, shut down all instances on all nodes running the Sun ONE Portal Server software in your system. The upgrade process involves changes that do not allow the old Sun ONE Portal Server software version to be run at the same time as the new Sun ONE Portal Server software version.

2. Upgrade the Sun ONE Identity Server 5.1 software to the Sun ONE Identity Server 6.1 version on all nodes running the Sun ONE Portal Server software in your system. See Chapter 12, “Upgrading from Sun ONE Identity Server 5.1 to Sun™ ONE Identity Server 6.1” on page 191.

3. Upgrade each instance of the Sun ONE Portal Server software from the 6.0 release to the 6.2 release. See Chapter 14, “Upgrading Sun™ ONE Portal Server Software from Version 6.0 to Version 6.2” on page 225.

4. Restart your systems and allow user traffic.

Upgrading Certificates

If the name of the Sun ONE Portal Server 6.2 system is the same as the name of the Sun ONE Portal Server 6.0 system from which you are upgrading, you can use your existing certificates. If the names are not the same, you will need new certificates. For details on using certificates with Sun ONE Portal Server 6.2, see the *Sun ONE Portal Server* 6.2 *Administrator’s Guide* sections on configuring SSL.

Upgrading Multiple Nodes

Use the following information to help plan the upgrade process when upgrading multiple nodes:

- Your Sun ONE Portal Server 6.0 system can consist of one or more nodes running the Sun ONE Portal Server software, all of which share a single logical Sun™ ONE Directory Server for its LDAP data store. The upgrade process involves changes that do not allow the old Sun ONE Portal Server software version to be run at the same time as the new Sun ONE Portal Server software version. Therefore, before you start the upgrade process you must completely shut down all instances running the Sun ONE Portal Server software in your system.
- Once the LDAP data has been migrated and the primary or first Sun ONE Portal Server node has been upgraded, you may restore traffic to your Sun ONE Portal Server system. Subsequent nodes may be upgraded while the primary Sun ONE Portal Server node is in operation.

Chapter 12

Upgrading from Sun ONE Identity Server 5.1 to Sun™ ONE Identity Server 6.1

NOTE All instances of the Sun™ ONE Identity Server 5.1 product refer to what was formerly known as the iPlanet™ Directory Server Access Management Edition 5.1 product.

This chapter provides an overview and a description of the process required to upgrade Sun ONE Identity Server 5.1 software to Sun ONE Identity Server 6.1 software in order to upgrade your Sun™ ONE Portal Server system from the 6.0 software release to the 6.2 software release.

This chapter contains the following sections:

- Overview Sun ONE Identity Server Upgrade Process
- Backing Up Existing Installation
- Removing Sun ONE Portal Server 6.0 Packages and Backing Up Templates
- Upgrading the Web Container
- Uninstalling Sun ONE Identity Server 5.1 Components and Schema
- Upgrading the Existing Directory Server Installation with the Sun ONE Identity Server 6.1 Schema
- Retaining the Top Level Entry of the Directory Information Tree (DIT)
- Installing Sun ONE Identity Server 6.1 Components on the Existing Directory Server Installation

- Migrating Sun ONE Identity Server 5.1 Services, Policies, Authentication Entries, and Top Level Organization Data

NOTE You must completely shut down all instances running the Sun ONE Portal Server software in your system before you start the upgrade process.

To shut down a Sun ONE Portal Server system using Sun™ ONE Web Server as its web container, use the `/etc/init.d/amserver stop` command.

To shut down a Sun ONE Portal Server system using an application server as its web container, use the appropriate application server stop command.

Overview Sun ONE Identity Server Upgrade Process

The Sun ONE Identity Server 5.1 (5.1a for application servers) to Sun ONE Identity Server 6.1 upgrade process involves the following high-level steps:

1. Backing up the existing installation of the Sun ONE Identity Server 5.1 and the web container.
2. Running Sun ONE Portal Server 6.2 preupgrade script to remove the Sun ONE Portal Server 6.0 packages and backup any Sun ONE Portal Server modified templates.
3. Upgrading the web container.
4. Uninstalling Sun ONE Identity Server 5.1 product components and the schema, but not the Sun™ ONE Directory Server 5.1 installation.
5. Upgrading the existing Sun ONE Directory Server with the Sun ONE Identity Server 6.1 schema changes.
6. Retaining the top level entry of your Directory Information Tree (DIT)
7. Installing Sun ONE Identity Server 6.1 component software on the existing Directory Server 5.1 installation.

8. Migrating Sun ONE Identity Server 5.1 services, policies, authentication entries, and top level organization data to the Sun ONE Identity Server 6.1 format.

NOTE The steps in the process must be performed in the order they are listed here.

It is expected that the person performing the upgrade procedure is familiar with Sun ONE Directory Server commands, schema semantics, DIT, and the Sun ONE Identity Server schema and DIT structures. In addition, familiarity with XML and Sun ONE Identity Server installation procedure is required. For further information on Sun ONE Directory Server commands or concepts, refer to the Sun ONE Directory Server documentation suite located at `http://docs.sun.com`

For further information on the process or migrating data from the Sun ONE Identity Server 5.1 to Sun ONE Identity Server 6.1, refer to the *Sun ONE Identity Server 6.1 Installation and Configuration Guide.*

Backing Up Existing Installation

The Sun ONE Identity Server 5.1 product stores user and service configuration on an associated directory server. Use the directory server `db2bak` command to back up the Sun ONE Identity Server 5.1 data. This command is available in the `slapd-`*HOSTNAME* directory within the base directory of the directory server. For example, if the directory server was installed to the default install directory (`/usr/ldap`) on the server `sesta`, the base directory would be `/usr/ldap/slapd-sesta`. You must also back up the web container (web server or application server) configuration data.

1. Create a backup of the database using the `db2bak` command and specify the backup directory in which to save the backup file. For example, to create a backup of the database on the server and store it in the `/export/upgrade_bak/DS5.1bak/db2bakup/config` directory, type the following:

```
/DS_BASEDIR/slapd-HOSTNAME/db2bak
/export/upgrade_bak/DS5.1bak/db2bakup
```

2. Create a backup copy of the configuration for the directory server. For example, to copy the configuration and store it in the `/export/upgrade_bak/DS5.1bak/config` directory, type the following:

```
cp -r /DS_BASEDIR/slapd-HOSTNAME/config/
/export/upgrade_bak/DS5.1bak/db2bakup
```

3. Run the `saveconfig` command to save the Admin Server configuration.

   ```
   /DS_BASEDIR/slapd-HOSTNAME/saveconfig
   ```

   This saves the Admin Server configuration to `aconfbak` directory.

4. Create a backup copy of the Admin Server configuration.

   ```
   cp -r /DS_BASEDIR/slapd-HOSTNAME/confbak
   /export/upgrade_bak/DS5.1bak
   ```

5. Create backup copies of the web container data for you installation. For example, on a web server deployment, copy the data in each of the following directories:

   ```
   /BASEDIR/SUNWam/web-apps/applications
   /BASEDIR/SUNWam/web-apps/services
   /BASEDIR/SUNWam/servers/alias
   /BASEDIR/SUNWam/config
   /BASEDIR/SUNWam/lib
   /BASEDIR/SUNWam/locale
   ```

   where *BASEDIR* is the directory in which Sun ONE Identity Server 5.1product was installed. For example, to backup the /*BASEDIR*/`SUNWam/web-apps/applications` directory to `/export/upgrade_bak/WS6bak`, type the following:

   ```
   cp -r /BASEDIR/SUNWam/web-apps/applications
   /export/upgrade_bak/WS6bak
   ```

NOTE If SSL is not enabled, the /*BASEDIR*/`SUNWam/servers/alias` directory may not exist.

6. Create backup copies of the logs, debug, and install files. Copy the data in each of the following directories:

   ```
   /var/opt/SUNWam/logs
   /var/opt/SUNWam/debug
   /var/opt/SUNWam/install
   ```

   For example, to backup the `/var/opt/SUNWam/logs` directory to `/export/upgrade_bak/WS6bak/logs`, type the following:

   ```
   cp -r /var/opt/SUNWam/logs /export/upgrade_bak/WS6bak/logs
   ```

7. For application server deployments, create backup copies of the application server's instance configuration directory. For example, to back up the default instance configuration directory on the Sun ONE Application Server (`/var/opt/SUNWappserver7/domains/domain1/server1/config`) to `/export/upgrade_bak/s1as7_bak/var_bak`, type the following:

   ```
   cp -r /var/opt/SUNWam/appserver7/domains/domain1/server1/config
   /export/upgrade_bak/s1as7_bak/var_bak
   ```

Removing Sun ONE Portal Server 6.0 Packages and Backing Up Templates

The Sun ONE Portal Server 6.2 software includes a script (`preupgrade`) that backups up any modified templates and removes all the Sun ONE Portal Server 6.0 packages. This script is located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of this script is located in `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools` directory.

1. Log in to the machine and become superuser.

   You will need root access to uninstall the Sun ONE Portal Server.

2. Change directories to where the `preupgrade` script is located. For example,

   ```
   cd /orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools
   ```

3. Type:

   ```
   ./preupgrade
   ```

   The script processes each Sun ONE Portal Server 6.0 installed package for user modifications, saves any modifications to `/tmp/ps_backup`, and then removes each Sun ONE Portal Server 6.0 package.

Upgrading the Web Container

Sun ONE Identity Server 6.1 and Sun ONE Portal Server 6.2 software can be deployed into a variety of web containers including the Sun™ ONE Web Server 6.1, the Sun™ ONE Application Server 7.0 Update 1, the IBM Websphere Application Server, and the BEA WebLogic Application Server; however, when upgrading your Sun ONE Portal Server 6.0 installation into a Sun ONE Portal Server 6.2 installation you must deploy it into the same brand of web container.

The Sun ONE Web Server 6.1 software and the Sun ONE Application Server 7 update 1 software is included with the Sun Java™ Enterprise System product. If you are upgrading to one of these web containers, use the Sun Java™ Enterprise System installer program to install the software.

Upgrading to Sun ONE Web Server 6.1 Software

You can upgrade to the Sun ONE Web Server 6.1 software using the Sun Java™ Enterprise System installer program located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of the installer is located in `/orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc` directory.

1. If you have not already done so, log in to the machine and become superuser.

   You will need root access to install the Sun ONE Web Server.

2. If you are not already working in the directory where you downloaded the product, use the cd command to change to that directory. For example,

   ```
   cd /orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc
   ```

3. If you used the `su` command to become root on your system, use the `xhost` command to grant access to your display. For example, use the following command to grant access to all users:

   ```
   xhost +
   ```

4. Type the following command to launch the installer:

   ```
   ./installer
   ```

5. At the Welcome screen, click Next.

6. Review the Software License Agreement and click Yes, Accept License to accept.

7. Specify the appropriate language support for the system and click Next.

8. On the Select Components tab, check the checkbox for the Sun ONE Web Server 6.1 component, uncheck all the other checkboxes, and click Next.

9. Check the Upgrade existing J2SE SDK radio button and click OK.

10. On the Shared Component Upgrades Required panel, review the list of shared components that must be upgraded and click Next.

11. Verify the target installation directories and click Next.

12. If the Checking System Requirements panel indicates that any patches are needed, cancel the install and add any required patches. Otherwise, click Next to continue.

13. On the Configuration Type Panel, select Minimal Configuration and click Next.

14. A series of configuration panels will display. Verify the settings on each of these panels and accept the settings by clicking Next. Make sure to specify the same admin server port and web server port as they were there for Sun ONE Identity Server 5.1 web server as well as the same user/admin passwords.

15. On the Ready to Install panel, click Next.

16. On the Installation Complete panel, select the appropriate button to view the Summary or the Install Log and click Close.

Upgrading to Sun ONE Application Server 7.0 Update 1 Software

Before upgrading to the Sun ONE Application Server 7.0 Update 1 software, you must manually make copies of the application server data in the `/etc/opt` and `/var/opt` directories and perform the uninstall using the Sun ONE Application Server 7.0 uninstaller. Once that is completed, you can upgrade using the Sun Java™ Enterprise System installer program located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of the installer is located in `/orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc` directory.

1. Create backup copies of the existing application server data from the following directories:

   ```
   /etc/opt/SUNWappserver7
   /var/opt/SUNWappserver7
   ```

   For example, to backup the `/etc/opt/SUNWappserver7` directory to `/export/upgrade_bak/s1as7_bak/etc_bak`, type the following command:

   ```
   cp -r /etc/opt/SUNWappserver7 /export/upgrade_bak/s1as7_bak/etc_bak
   ```

2. Use the `cd` command to change directories to the directory containing the Sun ONE Application Server 7.0 uninstaller.

   ```
   cd /opt/SUNWappserver7
   ```

3. Type the following command to launch the Sun ONE Application Server uninstaller:

   ```
   ./uninstall
   ```

   When the uninstall is complete, install the Sun ONE Application Server 7.0 Update 1 software.

4. If you are not already working in the directory where you downloaded the Sun Java Enterprise System product, use the `cd` command to change to that directory. For example,

   ```
   cd /orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc
   ```

NOTE Make sure that no appservd or imq processes are running.

5. If you used the `su` command to become root on your system, use the `xhost` command to grant access to your display. For example, use the following command to grant access to all users:

   ```
   xhost +
   ```

6. Type the following command to launch the installer:

   ```
   ./installer
   ```

7. At the Welcome screen, click Next.

8. Review the Software License Agreement and click Yes, Accept License to accept.

9. Specify the appropriate language suppport for the system and click Next.

10. On the Select Components tab, check the checkbox for the Sun ONE Application Server 7.0 component, uncheck all the other checkboxes, and click Next.

11. On the Shared Component Upgrades Required panel, review the list of shared components that must be upgraded and click Next.

12. Verify the target installation directories and click Next.

13. If the Checking System Requirements panel indicates that any patches are needed, cancel the install and add any required patches. Otherwise, click Next to continue.

14. On the Configuration Type Panel, select Minimal Configuration and click Next.

15. A series of configuration panels will display. Verify the settings on each of these panels and accept the settings by clicking Next. Accept the default locations.

16. On the Ready to Install panel, click Next.

17. On the Installation Complete panel, select the appropriate button to view the Summary or the Install Log and click Close.

18. When the installation completes, delete the content in the `domains.bin` and `/etc/opt/SUNWappserver7` directory and `/var/opt/SUNWappserver7/domains` directory. For example, to delete the content in the `/etc/opt/SUNWappserver7/domains.bin` directory, type the following command:

    ```
    rm -r /etc/opt/SUNWappserver7domains.bin
    ```

19. Copy back the backed up `/etc/opt` and `/var/opt` directories from Step 1.

    ```
    cp -r /export/upgrade_bak/s1as7_bak/etc_bak/* /
    etc/opt/SUNWappserver7/.
    cp -r /export/upgrade_bak/s1as7_bak/var_bak/* /
    var/opt/SUNWappserver7/.
    ```

20. Verify that you can access the Sun ONE Application Server 7.0 administration console. For example, use the following URL to access the administration console:

    `http://hostname:4848.`

Uninstalling Sun ONE Identity Server 5.1 Components and Schema

Use the Sun ONE Identity Server 5.1 uninstallation program to remove Policy and Management service and schema components of Sun ONE Identity Server 5.1 product, but DO NOT remove Sun ONE Directory Server 5.1 product.

1. Change directories to where the Sun ONE Identity Server 5.1 installation program is located. (The installation program is located in the `idsame` directory on the Sun ONE Portal Server CD or download image.)

2. Type **`ksh aminstall`**

3. Specify **`y`** to accept the license agreement.

4. At the “What would you like to do?“prompt, select option 1, “Remove existing components, then continue installation”.
5. At the “Please select one of the bundles to remove from ...” prompt, select option 1, “DSAME Management and Policy Services”.

   The program uninstalls the Sun ONE Identity Server 5.1 Policy and Management Services.
6. At the “Please select one of the bundles to remove from ...”prompt, select option 3, ”iPlanet Directory Server Configuration for DSAME”.

   This will remove the Sun ONE Identity Server 5.1 schema configuration for the Directory Server.
7. At the “Please select one of the bundles to remove from ...” prompt, select option 7 to exit the Sun ONE Identity Server installation program.
8. Check for the `SUNWamjdk` package after the uninstallation is complete using the following command:

   ```
   pkginfo |grep SUNWamjdk
   ```
9. If the `SUNWamjdk` package is present, remove it using the following command:

   ```
   pkgrm SUNWamjdk
   ```
10. If you had Sun ONE Identity Server 5.1 SP2 installed on your system, enter the following:

    ```
    patchrm 113626-01
    ```
11. For Sun ONE Application Server Enterprise Edition deployments, use the `pkgrm` command to remove the following additional packages:

    ```
    pkgrm SUNWamsas
    pkgrm SUNWamsac
    ```
12. For IBM Websphere application server deployments, use the `pkgrm` command to remove the following additional packages:

    ```
    pkgrm SUNWamwss
    pkgrm SUNWamwsc
    ```
13. For BEA WebLogic application server deployments, use the `pkgrm` command to remove the following additional packages:

    ```
    pkgrm SUNWamwls
    pkgrm SUNWamwlc
    ```

14. For application server deployments, go to the application server console, apply the changes, and stop and restart the application server instance.

    Use the mechanism appropriate for your application server. For more information on installing into an application server deployment, refer to the *Sun ONE Portal Server 6.2 Installation Guide* or the documentation specific to your application server.

    Because the Sun ONE Identity Server has been removed but the Portal Server is still installed, exceptions will be logged. You can ignore these exceptions.

15. On all deployments, stop and restart the Sun ONE Directory Server.

    ```
    /DS_BASEDIR/slapd-HOSTNAME/stop-slapd
    /DS_BASEDIR/slapd-HOSTNAME/start-slapd
    ```

Upgrading the Existing Directory Server Installation with the Sun ONE Identity Server 6.1 Schema

Use the `ldapmodify` command to upgrade the existing directory with the Sun ONE Identity Server 6.1 schema formats and data. There are two LDIF files with updated schema information (`sunone_schema2.ldif` and `ds_remote_schema.ldif`). These files are located on the Sun Java Enterprise System media. For example, the Solaris SPARC versions are located in `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools` directory.

1. Change directories to where the Sun ONE Identity Server 6.1 updated schema files are located.

2. Use the `ldapmodify` command from the directory server install directory to load the `sunone_schema2.ldif` file to directory server.

    ```
    /DS_BASEDIR/shared/bin/ldapmodify -h HOSTNAME -p 389 -D
    “cn=Directory Manager” -w PASSWORD -c -f sunone_schema2.ldif
    ```

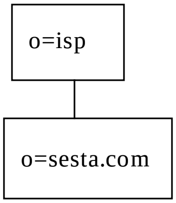
3. Use the `ldapmodify` command from the directory server install directory to load the `ds_remote_schema.ldif` file to directory server.

    ```
    /DS_BASEDIR/shared/bin/ldapmodify -h HOSTNAME -p 389 -D
    “cn=Directory Manager” -w PASSWORD -c -f ds_remote_schema.ldif
    ```

Retaining the Top Level Entry of the Directory Information Tree (DIT)

Before installing Sun ONE Identity Server 6.1 software you will need to verify that the Directory Server 5.1 installation that existed with iPlanet Directory Server Access Management Edition 5.1 is set up to retain the existing DIT structure and ensure proper installation. To retain the Sun ONE Identity Server 5.1 DIT structure, the top level entry must have the iplanet-am-service-status attribute set. Depending on your installation, this attribute may or may not be set at the top level of your DIT. For example, in the default installation of the Sun ONE Portal Server 6.0 release, the DIT might have a hierarchical structure with o=isp at the top of the structure and an organization such as o=sesta.com beneath it, as is shown in Figure 12-1.

Figure 12-1 Hierarchical Directory Structure



In this case, isp is not really an organization and as such does not have an iplanet-am-service-status attribute set. To retain the Sun ONE Identity Server 5.1 DIT structure, the top level entry, *isp*, must be updated before you install Sun ONE Identity Server 6.1. Use the ldapmodify command to load an LDIF file (top_entry.ldif) with the updated top level entry. This file is located on the Sun Java Enterprise System media. For example, the Solaris SPARC version is located in /orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools directory. If the Sun ONE Identity Server 5.1 Directory Server has an organization (flat DIT) as top level entry, then this change is not necessary before installing Sun ONE Identity Server 6.1.

To update the top level entry of your DIT:

1. Change directories to where the Sun ONE Identity Server 6.1 top_entry.ldif file is located.

2. Use a text editor to edit the LDIF file and set the ROOT_SUFFIX variable to the top level entry per your DIT.

3. Use the `ldapmodify` command from the directory server install directory to load the `top_entry.ldif` file to directory server.

   *`/DS_BASEDIR`***`/shared/bin/ldapmodify -h`** *`HOSTNAME`* **`-p 389 -D "cn=Directory Manager" -w`** *`PASSWORD`* **`-c -f top_entry.ldif`**

4. Load the entry and aci for amldapuser into directory server using the `ldapmodify` command. Use `amldapuser.ldif` and `amldapuser-aci.ldif` files.

NOTE The password set in the `amldapuser.ldif` file is 111111111. You may want to change it according to your DIT but make sure that it is different from Admin password.

---

   a. Type:

```
/usr/ldap/shared/bin/ldapmodify -h pikes -p 389 -D 'cn=directory
manager' -w iplanet3 -c -f
/net/storage/export/images/orion/s9orion/orion1of2_sparc.s9_orio
n/latest/Solaris_sparc/portal_svr/Tools/amldapuser.ldif
```

   b. Type:

```
/usr/ldap/shared/bin/ldapmodify -h pikes -p 389 -D 'cn=directory
manager' -w iplanet3 -c -f
/net/storage/export/images/orion/s9orion/orion1of2_sparc.s9_orio
n/latest/Solaris_sparc/portal_svr/Tools/amldapuser-aci.
ldif
```

Installing Sun ONE Identity Server 6.1 Components on the Existing Directory Server Installation

Install the Sun ONE Identity Server 6.1 software using the Sun Java™ Enterprise System installer program located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of the installer is located in `/orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc` directory.

1. If you have not already done so, log in to the machine and become superuser.

   You will need root access to install the Sun ONE Identity Server.

2. If you are not already working in the directory where you downloaded the Sun Java Enterprise System product, use the cd command to change to that directory. For example,

   ```
   cd /orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc
   ```

3. If you used the su command to become root on your system, use the xhost command to grant access to your display. For example, use the following command to grant access to all users:

   ```
   xhost +
   ```

4. Type the following command to launch the installer:

   ```
   ./installer
   ```

5. At the Welcome screen, click Next.
6. Review the Software License Agreement and click Yes, Accept License to accept.
7. Specify the appropriate lanquage suppport for the system and click Next.
8. On the Select Components tab, check the checkbox for the Sun ONE Identity Server 6.1 component, uncheck all the other checkboxes, and click Next.
9. A warning message displays indicating that the Sun ONE Identity Server requires the Sun ONE Directory Server be installed or that Sun ONE Identity Server be configured to access a remote Directory Server. Click Continue.
10. Check the Upgrade existing J2SE SDK radio button and click OK.
11. On the Shared Component Upgrades Required panel, review the list of shared components that must be upgraded. Accept the upgrading of the SUNWicu package and click Next.
12. Verify the target installation directories and click Next.
13. If the Checking System Requirements panel indicates that any patches are needed, cancel the install and add any required patches. Otherwise, click Next to continue.
14. On the Configuration Type Panel, select Custom Configuration and click Next.

15. A series of configuration panels will display depending on the type of Web Container you chose to upgrade to. Verify the settings on each of these panels and accept the settings by clicking Next. Unselect Sun ONE Directory Server 5.2 and Administration Server 5.2 and Root in Directory Server as o=isp and the amldapuser password is 111111111 (or the password used in Step 4 of "Retaining the Top Level Entry of the Directory Information Tree (DIT)."

    The following entries specified should match the values that were set for the original Sun ONE Identity Server 5.1 installation:

    directory root suffix
    directory manager password
    admin user
    admin password
    directory server host
    directory server port
    console deployment descriptor
    services deployment descriptor

    Refer to the `AMConfig.properties` file from the Sun ONE Identity Server 5.1 installation backup for any values of which you are not sure.

16. On the Identity Server (1 of 6) panel,

    a. Verify the settings for the Administrator User ID, Administrator Password, LDAP User ID.

    b. Enter the Identity Server Internal LDAP Authentication User Password and retype password to confirm.

    c. Enter the Password Encryption Key value:

       KmhUnWR1MYWDYW4xuqdF5nbm+CXIyOVt

    d. Click Next to continue.

17. On the Identity Server (2 of 6) panel, select the appropriate Web Container for your implementation and click Next.

18. On the Identity Server (3 of 6) panel, verify the settings for the Web Container and click Next.

19. On the Identity Server (4 of 6) panel, verify the configuration settings and click Next.

20. On the Identity Server (5 of 6) panel,

    a. Verify the Directory Server Host, the Identity Server Directory Root Suffix, and the Directory Manager DN. Directory Server Port: 389

   b. Enter 389 as the Directory Server Port and enter the correct Directory Manager Password.

   c. Click Next to continue.

21. On the Identity Server (6 of 6) panel, at the “Is Directory Server provisioned with user data?” query specify Yes.

22. Verify the Organization Marker Object Class, the Organization Naming Attribute, the User Marker Object Class, and the User Naming Attribute and click Next.

23. Click No in response to the following Info message:

    This Directory Server does not have Identity Server 6.1 DIT. Do you want the installation program to load the DIT into your Directory Server?

24. On the Ready to Install panel, click Next.

25. On the Installation Complete panel, select the appropriate button to view the Summary or the Install Log and click Close.

26. For application server deployments, go to the application server console, apply the changes, and stop and restart the application server instance.

    Use the mechanism appropriate for your application server. For more information on installing into an application server deployment, refer to the *Sun ONE Portal Server 6.2 Installation Guide* or the documentation specific to your application server.

Migrating Sun ONE Identity Server 5.1 Services, Policies, Authentication Entries, and Top Level Organization Data

Once Sun ONE Identity Server 6.1 is installed and the Sun ONE Directory Server schema is updated, the directory server data must be modified to Sun ONE Identity Server 6.1 format. In Sun ONE Identity Server 6.1, policy, authentication and console components have changed significantly from Sun ONE Identity Server 5.1 release and hence need to be migrated.

All the migration scripts needed for this are located on the Sun Java Enterprise System media. For example, the Solaris SPARC version is located in `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools` directory. The scripts contain additional information, which you must read before running the scripts. It will help you set some variables in each script or check the values of the variables.

1. Set the `LD_LIBRARY_PATH` environment variable to `/usr/iplanet/servers/lib/`. For example,

   `setenv LD_LIBRARY_PATH=/usr/iplanet/servers/lib/`

2. In each of the migration scripts, verify or specify that the path to the `LDAP_SEARCH` variable points to `/usr/iplanet/servers/shared/bin/ldapsearch` or a value appropriate for your deployment.

3. Run the `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools/update-schema.pl` script to migrate the schema data. At the following script prompts, verify or specify the appropriate values and press Enter:

   ```
   Enter Host name [DIRECTORY_SERVER]:<hostname>
   Enter Bind User name [cn=directory manager]:
   Enter Bind password:<password>
   Enter port number [389]:
   Enter root suffix [o=sun.com]:o=isp
   ```

   This script generates an output file, `60entries.ldif`.

4. Set the `LD_LIBRARY_PATH` environment variable to `/usr/ldap/lib:$LD_LIBRARY_PATH`. For example,

   `setenv LD_LIBRARY_PATH /usr/ldap/lib:$LD_LIBRARY_PATH`

5. Run the `ldapmodify` command on the output file generated in Step 3 to update the schema. For example, type

   DS_BASEDIR/**shared/bin/ldapmodify -h** *HOSTNAME* **-p 389 -D “cn=Directory Manager” -w** *PASSWORD* **-c -f 60entries.ldif**

6. Run the `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools/update-policies.pl` script to migrate the policy data. At the `update-policies.pl` script prompts, verify or specify the appropriate values and press Enter.

   This script creates an xml file for each organization containing policies. For example, if the organization is `o=sesta.com,o=sales`, then the script creates a file named `o=sesta.com-o=sales.xml`. These files will be used later.

7. Run the /orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools/update-auth.pl script to migrate the authentication data. At the update-auth.pl script prompts, verify or specify the appropriate values and press Enter. In addition to the standard prompts listed in Step 3, the following prompt also displays.

   ```
   In IS 6.x a new attribute iplanet-am-auth-cert-gw-cert-auth-enabled
   is added to the iPlanetAMAuthCertService. Please find the value of
   an attribute com.iplanet.authentication.modules.cert.gwAuthEnable
   in your IS 5.1 AMConfig.properties and enter it here. (enter no if
   the attribute is not there) [no]:
   ```

   The script generates an input file, 51auth-entries.dn, and an output file, 51to60auth-entries.ldif. These files will be used later.

TIP This script will only migrate the Sun ONE Identity Server provided authentication modules. Any customized authentication modules must implement the Authentication Service Provider Interface (SPI) (which provides the authentication framework) and need to be rewritten using com.sun.identity.authentication.spi.AMLoginModule class. In addition, you must also do the following:

- Create or update the XML service file for the authentication module. For example, amAuthLDAP.xml defines the LDAP service parameters. If this file existed in Sun ONE Identity Server 5.1 implementation, it is migrated automatically by the Sun ONE Identity Server 6.1 migration script.
- Create or update the authentication module configuration file. This file specifies the authentication module credentials by the defining the user authentication screens. In Sun ONE Identity Server 5.1, this file was a .properties file (for example, LDAP.properties). In Sun ONE Identity Server 6.1, this file is a .xml file (for example, LDAP.xml).
- Create or update the localization properties file. For example, amAuthLDAP.properties defines the LDAP properties.
- Update the iplanet-am-authenticators authentication attribute in the amAuth.xml file by adding a value that specifies the fully qualified class name of the custom authentication module. This can be done by manually editing the attribute in the amAuth.xml file and then importing the file.

Refer to Appendix F, “Authentication Framework Changes Between Sun ONE Portal Server 6.0 and Sun ONE Portal Server 6.2” for details on the changes to the authentication framework service that you need to know for migration purposes. For detailed information on writing and implementing custom authentication modules, refer to the *Sun ONE Identity Server 6.1 Programmer’s Guide*.

NOTE The authentication menu attribute provided by Sun ONE Identity Server 5.1 admin console is not supported in Sun ONE Identity Server 6.1 release. If you need to configure a selectable list of valid authentication modules, use the admin console to set each authentication module with the same level in the authentication level attribute. Users can access a page with all the authentication modules for a specific authentication level by specifying the authlevel in the login URL. For example, a user performs a login with the following syntax:

`http://`*HOSTNAME*`:`*PORT*`/`*DEPLOY_URI*`/UI/Login?authlevel=`*N*

Refer to the *Sun ONE Portal Server 6.2 Administrator's Guide* for information on configuring authentication modules.

8. Run the `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools/update-toporg-services.pl` script to migrate the top level organization data. At the `update-toporg-services.pl` script prompts, verify or specify the appropriate values and press Enter.

   This script generates an input file, `51toporg-template.dn` and an output file, `51to60toporg-template.ldif.`

9. Use Directory Server console to remove the following Sun ONE Identity Server 5.1 services:

NOTE If you added other services to Sun ONE Identity Server, don't remove them. If you have not added any additional services, you can remove the entire services branch under the top level entry.

- iPlanetAMAdminConsoleService
- iPlanetAMAuthService
- iPlanetAMAuthAnonymousService
- iPlanetAMAuthCertService
- iPlanetAMAuthLDAPService
- iPlanetAMAuthMembershipService
- iPlanetAMAuthNTService

- iPlanetAMAuthRadiusService
- iPlanetAMAuthSafewordService
- iPlanetAMAuthUnixService
- iPlanetAMClientDetectionService
- iPlanetAMDomainURLAccessService
- iPlanetAMEntrySpecificService
- iPlanetAMLoggingService
- iPlanetAMNamingService
- iPlanetAMPlatformService
- iPlanetAMPolicyService
- iPlanetAMSessionService
- iPlanetAMUserService
- iPlanetAMWebAgentService
- DAI

10. Use Directory Server console to remove the Sun ONE Identity Server 5.1 user's default login URL attribute (`iplanet-am-user-default-url`) from your user entries as this attribute is no longer available in Sun ONE Identity Server 6.1.

NOTE If you do not remove the `iplanet-am-user-default-url` attribute, you will get object class violation errors.

11. In the `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools/load-services.pl` script, verify or specify that the path to the `$PROD_DIR` variable points to `SUNWam`.

12. Run the `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools/load-services.pl` script to load the services data. At the `load-services.pl` script prompts, verify or specify the appropriate values and press Enter

13. Run the `ldapmodify` command on the output file generated in Step 7 to update the authentication data. For example, type

```
/DS_BASEDIR/shared/bin/ldapmodify -h HOSTNAME -p 389 -D
"cn=Directory Manager" -w PASSWORD -a -c -f 51to61auth-entries.ldif
```

14. Run the /orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools/update-services.pl script to update the services data. At the update-services.pl script prompts, verify or specify the appropriate values and press Enter.

    This script generates an an output file, 60services.ldif.

15. Run the ldapmodify command on the output file generated in Step 14 to update the services data. For example, type

    /DS_BASEDIR/**shared/bin/ldapmodify -h** *HOSTNAME* **-p 389 -D “cn=Directory Manager” -w** *PASSWORD* **-c -f 60services.ldif**

16. Replace the ROOT_SUFFIX in the liberty_services.ldif file.

    Sun ONE Identity Server 6.1 supports Liberty Alliance Federation Management. This file registers two services (iPlanetAMAuthenticationDomainConfigService and iPlanetAMProviderConfigService) to support the Sun ONE Identity Server 6.1 Liberty Alliance Federation Management.

17. Run the ldapmodify command to the liberty_services.ldif file from Step 16 to update the Liberty Services data in the schema. For example, type

    /DS_BASEDIR/**shared/bin/ldapmodify -h** *HOSTNAME* **-p 389 -D “cn=Directory Manager” -w** *PASSWORD* **- a -c -f liberty_services.ldif**

18. Run the /orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools/delete-policies.pl script to generate the policies data to delete. At the delete-policies.pl script prompts, verify or specify the appropriate values and press Enter.

    This script generates an an output file, delete-policies.ldif.

19. Run the ldapdelete command on the output file generated in Step 18 to delete the policies data. For example, type

    /DS_BASEDIR/**shared/bin/ldapdelete -h** *HOSTNAME* **-p 389 -D “cn=Directory Manager” -w** *PASSWORD* **-f delete-policies.ldif**

20. Manually set the amAdmin distinguished names in AMConfig.properties file. For example

    ```
    com.sun.identity.authentication.super.user=uid=amAdmin,ou=
    People,o=iplanet.com,o=isp
    ```

21. Run the `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools/scripts/UpgradeDIT` script to generate the policies data to delete. At the script prompts, verify or specify the appropriate values and press Enter.

    You will need to provide the directory manager and amAdmin passwords. Refer to the `AMConfig.properties` file from the Sun ONE Identity Server 5.1 installation.

22. Replace the ROOT_SUFFIX in the `auth_entries.ldif` file.

23. Run the `ldapmodify` command to the `auth_entries.ldif` file from Step 22 to update the Liberty Services data in the schema. For example, type

    ```
    /DS_BASEDIR/shared/bin/ldapmodify -h HOSTNAME -p 389 -D
    “cn=Directory Manager” -w PASSWORD - a -c -f auth_entries.ldif
    ```

24. Stop and restart the Sun ONE Directory Server.

    ```
    /DS_BASEDIR/slapd-HOSTNAME/stop-slapd
    /DS_BASEDIR/slapd-HOSTNAME/start-slapd
    ```

25. Stop and start the web container. For example, on a Sun ONE Web Server, type the following commands to change directories to the web server script location and run the start and stop scripts:

    ```
    cd /opt/SUNWwbsvr/https-HOSTNAME
    ./stop
    ./start
    ```

    For servers deployed on an application server, use the mechanism appropriate for your application server.

26. Log into the Sun ONE Identity Server 6.1 administration console.

NOTE If the error “Internal Authentication Error” displays, manually set the `login.configuration.provider` line in the `JDR_DIR/jre/lib/security/java.security` file as follows:

```
login.configuration.provider=com.sun.identity.
authentication.config.AMConfiguration
```

Restart the web container.

27. From the administration console, enable the user management service.

d. Navigate to the top-level organization for your implementation.

By default, Identity is selected in the location pane and Organizations is selected in the Navigation pane.

e. Select the Service Configuration tab.

f. Click the properties arrow icon next to Administration, check Enable User Management, and click Save.

28. Create referral policies at the top-level organization for your implementation and add rules for each service.

a. From the administration console, navigate to the top-level organization for your implementation.

b. Select Policies from the View menu in the navigation pane.

c. Click Create New.

d. For Type of Policy, select Referral.

e. For Name, type either `SubOrgReferral_`*`organization`* or `PeerOrgReferral_`*organization* where organization is the name of the organization for the referral and click Create.

f. Click Rules from the View menu data pane, select the rules for the policy and click Save.

g. Click Referrals from the View menu data pane and click Add.

h. Verify that the name of the peer or suborganization is selected for Value and click Create.

i. Click Save in the data pane to save the referral policy.

29. Register and create templates for the Policy Configuration service at the top-level organization for your implementation.

a. From the administration console, navigate to the top-level organization for your implementation.

b. Select the Services from the View menu.

c. Click Register.

d. Check Policy Configuration checkbox and then click Register.

e. Click the properties arrow icon next to the Policy Configuration service and click Create.

f. Enter the password for the user `amldapuser` and click Save.

30. Run the `amadmin` command for each of the output files generated in Step 6 to update the policies data. For example, if the output file was named `o=sesta.com-o=sales.xml`, type

    ```
    /BASEDIR/SUNWam/bin/amadmin -u
    “uid=amAdmin,ou=People,o=sesta.com,o=isp” -w PASSWORD -t
    o=sesta.com-o=sales.xml
    ```

31. From the administration console, check the users listed for each organization to verify that all the users have been migrated.

This completes the basic migration. For additional information on migrating iPlanet Directory Server Access Management Edition 5.1 data to Sun ONE Identity Server 6.1 formats, refer to Appendix A of the *Sun ONE Identity Server 6.1 Installation and Configuration Guide.*

Chapter 13

Upgrading from Sun ONE Identity Server 6.0 to Sun™ ONE Identity Server 6.1

This chapter provides an overview and a description of the process required to upgrade Sun ONE Identity Server 6.0 software to Sun ONE Identity Server 6.1 software in order to upgrade your Sun™ ONE Portal Server system from the 6.1 software release to the 6.2 software release.

This chapter contains the following sections:

- Overview of the Sun ONE Identity Server Upgrade Process
- Backing Up and Removing the Sun ONE Identity Server 6.0 Installation
- Installing Sun ONE Identity Server 6.1 Software
- Completing the Upgrade

NOTE You must completely shut down all instances running the Sun ONE Portal Server software in your system before you start the upgrade process.

To shut down a Sun ONE Portal Server system using Sun™ ONE Web Server as its web container, use the `/etc/init.d/amserver stop` command.

To shut down a Sun ONE Portal Server system using an application server as its web container, use the appropriate application server stop command.

Overview of the Sun ONE Identity Server Upgrade Process

The Sun ONE Identity Server 6.0 to Sun ONE Identity Server 6.1 upgrade process involves the following high-level steps:

1. Backing up and removing the Sun ONE Identity Server 6.0 installation.
2. Installing the Sun ONE Identity Server 6.1 software.
3. Completing the upgrade.

Backing Up and Removing the Sun ONE Identity Server 6.0 Installation

The Sun ONE Identity Server 6.1 software includes a script (`pre60to61upgrade`) that backs up the Sun ONE Identity Server 6.0 configuration and removes all the Sun ONE Identity Server 6.0 packages. This script is located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of this script is located in `/orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc/Product/identity_svr/Tools/` directory.

NOTE Back up the `AMConfig.properties` file before executing the `pre60to61upgrade` script.

1. Log in to the machine and become superuser.

   You will need root access to uninstall the Sun ONE Identity Server.

2. Change directories to where the `pre60to61upgrade script` is located. For example,

   ```
   cd /orion/pointproducts/Solaris_sparc/Product/identity_svr/Tools
   ```

3. Type:

   ```
   ./pre60to61upgrade
   ```

   A log of the pre60to61upgrade processing is output to `/var/sadm/install/logs/Sun_ONE_Identity_Server_upgrade_log.time_stamp`

4. Make sure that the Sun ONE Directory Server is running on the correct port. At the following prompts, verify the appropriate settings. Press Enter to accept the values.

```
Directory Server fully-qualified hostname.:
Directory Server port (389):
Top-Level Administrator DN:
Top-Level Administrator password:
Enter directory to store back up files (/opt):
```

Installing Sun ONE Identity Server 6.1 Software

Install the Sun ONE Identity Server 6.1 software using the Sun Java™ Enterprise System installer program located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of the installer is located in `/orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc` directory.

1. If you have not already done so, log in to the machine and become superuser.

   You will need root access to install the Sun ONE Identity Server.

2. If you are not already working in the directory where you downloaded the product, use the cd command to change to that directory. For example,

   ```
   cd /orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc
   ```

3. If you used the `su` command to become root on your system, use the `xhost` command to grant access to your display. For example, use the following command to grant access to all users:

   ```
   xhost +
   ```

4. Type the following command to launch the installer:

   ```
   ./installer
   ```

5. At the Welcome screen, click Next.

6. Review the Software License Agreement and click Yes, Accept License to accept.

7. Specify the appropriate language support for the system and click Next.

8. On the Select Components tab, check the checkbox for the Sun ONE Identity Server 6.1 component and for the type of Web Container you will be upgrading (Sun ONE Web Server, Sun ONE Application Server , BEA WebLogic, or IBM WebSphere), uncheck all the other checkboxes (Sun ONE Directory Server, Administration Server, App Server, Message Queue), and click Next.
9. A warning message displays indicating that the Sun ONE Identity Server requires the Sun ONE Directory Server be installed or that Sun ONE Identity Server be configured to access a remote Directory Server. Click Continue.
10. Check the Upgrade existing J2SE SDK radio button and click OK.
11. On the Shared Component Upgrades Required panel, review the list of shared components that must be upgraded and click Next.
12. Verify the target installation directories and click Next.
13. If the Checking System Requirements panel indicates that any patches are needed, cancel the install and add any required patches. Otherwise, click Next to continue.
14. On the Configuration Type Panel, select Custom Configuration and click Next.
15. A series of configuration panels will display depending on the type of Web Container you chose to upgrade to. Verify the settings on each of these panels and accept the settings by clicking Next. The following entries specified should match the values that were set for the original Sun ONE Identity Server 6.0 installation:

    directory root suffix
    directory manager password
    admin user
    admin password
    directory server host
    directory server port
    console deployment descriptor
    services deployment descriptor

    Refer to the `AMConfig.properties` file from the Sun ONE Identity Server 6.0 installation backup for any values of which you are not sure.
16. On the Identity Server (1 of 6) panel,
    a. Verify the settings for the Administrator User ID, Administrator Password, LDAP User ID.
    b. Enter the Identity Server Internal LDAP Authentication User Password and retype password to confirm.

c. Enter the Password Encryption Key value from the `AMConfig.properties` file from the Sun ONE Identity Server 6.0 installation.

NOTE The value of the encryption key for Sun ONE Identity Server 6.0 is static. The upgrade will fail if any other value is provided.

d. Click Next to continue.

17. On the Identity Server (2 of 6) panel, select the appropriate Web Container for your implementation and click Next.
18. On the Identity Server (3 of 6) panel, verify the settings for the Web Container and click Next.
19. On the Identity Server (4 of 6) panel, verify the configuration settings and click Next.
20. On the Identity Server (5 of 6) panel,

    a. Verify the Directory Server Host, the Identity Server Directory Root Suffix, and the Directory Manager DN. Directory Server Port: 389

    b. Enter 389 as the Directory Server Port and enter the correct Directory Manager Password.

    c. Click Next to continue.

21. On the Identity Server (6 of 6) panel, at the “Is Directory Server provisioned with user data?’ query specify Yes.
22. Verify the Organization Marker Object Class, the Organization Naming Attribute, the User Marker Object Class, and the User Naming Attribute and click Next.
23. Click No in response to the following Info message:

    This Directory Server does not have Identity Server 6.1 DIT. Do you want the installation program to load the DIT into your Directory Server?

24. On the Ready to Install panel, click Next.
25. On the Installation Complete panel, select the appropriate button to view the Summary or the Install Log and click Close.

Completing the Upgrade

The Sun ONE Identity Server 6.1 software includes a script (`Upgrade60DitTo61`) that upgrades the directory and schema information. When the Sun ONE Identity Server 6.1 software is installed, this script is copied into `/opt/SUNWam/migration/60to61/scripts/` directory. In addition to the running this script, you must also verify some deployment information, deploy the Portal WAR file, and configure some security, login, and authentication information.

Perform the following steps to complete the upgrade process.

1. Change directories to where the `Upgrade60DitTo61` script is located. For example,

   ```
   cd /opt/SUNWam/migration/60to61/scripts
   ```

2. Type:

   ```
   ./Upgrade60DitTo61
   ```

3. At the following prompts, verify the appropriate settings. Press Enter to accept the values.

   ```
   Directory Server fully-qualified hostname.:
   Directory Server port (389):
   Directory Manager DN ( cn=Directory Manager ):
   Directory Manager ( cn=Directory Manager ) password:
   Top-Level Administrator DN
   (uid=amAdmin,ou=People,dc=red,dc=iplanet,dc=com ):
   Top-Level Administrator
   (uid=amAdmin,ou=People,dc=red,dc=iplanet,dc=com ) password:
   ```

   The script displays the following message:

   ```
   IS_dit_version = 6.0
   Deleting DSS Service.
   Upgrading 6.0 schema.
   modifying entry cn=schema
   Create DIT succeeded.
   Before proceeding, you MUST restart the directory server.
   Hit return when ready.
   ```

4. In another terminal window, stop and restart the Sun ONE Directory Server using the following commands:

   ```
   /DS_BASEDIR/slapd-HOSTNAME/stop-slapd
   /DS_BASEDIR/slapd-HOSTNAME/start-slapd
   ```

5. In the first terminal window, enter return.

   The script continues processing the directory and schema information. When complete, the script displays the following message:

   ```
   Success 0: Sucessfully completed.
   Upgrade completed.
   ```

6. To make the upgrade changes take effect,

   a. Stop and restart the Sun ONE Directory Server again.

   b. Stop and start the web container. For example, on a Sun ONE Web Server, type the following commands to change directories to the web server script location and run the start and stop scripts:

      ```
      cd /opt/SUNWwbsvr/https-HOSTNAME
      ./stop
      ./start
      ```

This completes the basic upgrade process. Your Sun ONE Identity Server 6.1 should be running with the directory and schema information from your previous installation. Access the Sun ONE Identity Server Administration console to verify that the upgrade process completed successfully.

Chapter 14

Upgrading Sun™ ONE Portal Server Software from Version 6.0 to Version 6.2

This chapter provides an overview and discussion on how to upgrade your Sun™ ONE Portal Server system from the 6.0 software release to the 6.2 software release.

This chapter contains the following sections:

- The Upgrade Process
- Installing Sun ONE Portal Server 6.2 Software
- Restoring the Customized Files
- Upgrading Sun ONE Portal Server: Secure Remote Access Pack

The Upgrade Process

The Sun ONE Portal Server 6.0 to 6.2 upgrade process involves the following high-level steps:

1. Installing Sun ONE Portal Server 6.2 software.
2. Restoring the Sun ONE Portal Server 6.0 customized files.
3. Upgrading Sun ONE Portal Server: Secure Remote Access Pack 6.0 software.

NOTE	The steps in the process must be performed in the order they are listed here.

Installing Sun ONE Portal Server 6.2 Software

Install the Sun ONE Portal Server 6.2 software using the Sun Java™ Enterprise System installer program located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of the installer is located in `/orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc` directory.

1. If you have not already done so, log in to the machine and become superuser.

   You will need root access to install the Sun ONE Portal Server.

2. Change directories to where the installer is located. For example,

   ```
   cd /orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc
   ```

3. If you used the `su` command to become root on your system, use the `xhost` command to grant access to your display. For example, use the following command to grant access to all users:

   ```
   xhost +
   ```

4. Type the following command to launch the installer:

   ```
   ./installer
   ```

5. At the Welcome screen, click Next.

6. Review the Software License Agreement and click Yes, Accept License to accept.

7. Specify the appropriate language support for the system and click Next.

8. On the Select Components tab, uncheck any previously checked boxes and check only the Sun ONE Portal Server 6.2 checkbox.

9. A warning message displays indicating that the Sun ONE Portal Server requires the Sun ONE Application Server or Sun ONE Web Server be installed or that Sun ONE Portal Server be configured to an existing Web Container. Click Continue.

10. On the Shared Component Upgrades Required panel, review the list of shared components that must be upgraded and click Next.

11. Verify the target installation directories and click Next.

12. If the Checking System Requirements panel indicates that any patches are needed, cancel the install and add any required patches. Otherwise, click Next to continue.

13. On the Configuration Type Panel, select Minimal Configuration and click Next.

NOTE	You must install using the Minimal Configuration option to be able to restore the previous Portal Server configuration.

14. On the Ready to Install panel, click Next.
15. On the Installation Complete panel, select the appropriate button to view the Summary or the Install Log and click Close.

Restoring the Customized Files

The Sun ONE Portal Server 6.2 software includes a script (`postupgrade`) that restores user modified templates from the Sun ONE Portal Server 6.0 installation. This script is located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of this script is located in `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools` directory. In addition to the running the `postupgrade` script, you must also verify some deployment information, deploy the Portal WAR file, and configure some security, login, and authentication information.

Perform the following steps to complete the upgrade process.

1. Open the `/etc/opt/SUNWps/PSConfig.properties` file with a text editor and verify that it contains correct values for the following variables:

   ```
   DEPLOY_DIR=
   DEPLOY_DOCROOT=
   DEPLOY_DOMAIN=
   ```

   For example, on systems deployed in the default directory of a Sun ONE Web Server installation, the settings for `DEPLOY_DIR`, and `DEPLOY_DOCROOT` would be

   ```
   DEPLOY_DIR=/opt/SUNWwbsrvr
   DEPLOY_DOCROOT=/opt/SUNWwbsrvr/docs
   ```

2. Type the following command to deploy the Portal WAR file

   ```
   /opt/SUNWps/bin/deploy
   ```

a. When prompted into which instance to deploy, select the instance or select "." for all instances and press Enter to deploy the Portal WAR file.

b. When prompted for the deployment URI, enter `/portal`.

3. Change directories to where the postupgrade script is located. For example,

```
cd /orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools
```

4. Type:

```
./postupgrade
```

5. Update the Web Container's classpath in *web-container-install-root*/*instance-name*/`config/server.xml` to include the portlet jar files:

```
"/opt/SUNWps/lib/jdom.jar:/opt/SUNWps/lib/portlettl.jar:/opt/SUNWps/lib
/portlet.jar:/opt/SUNWps/lib/portletcontainercommon.jar:/opt/SUNWps/lib
/portletcommon.jar"
```

By updating the web container's classpath to include the portlet jar files, you can avoid the following exception when restarting the web container.

```
failure: WebModule[/portal]: WEB2671: Error configuring application listener of
class com.sun.portal.common.logging.impl.LoggerServletContextListenerImpl
java.lang.ClassNotFoundException:
com.sun.portal.common.logging.impl.LoggerServletContextListenerImpl
    at org.apache.catalina.loader.WebappClassLoader.loadClass
    (WebappClassLoader.java:1463)
    at org.apache.catalina.loader.WebappClassLoader.loadClass
    (WebappClassLoader.java:1299)
    at org.apache.catalina.core.StandardContext.listenerStart
    (StandardContext.java:3326)
    at org.apache.catalina.core.StandardContext.start
    (StandardContext.java:3720)
    at com.iplanet.ias.web.WebModule.start(WebModule.java:238)
    at org.apache.catalina.core.ContainerBase.start
    (ContainerBase.java:1133)
    at org.apache.catalina.core.StandardHost.start
    (StandardHost.java:652)
    at org.apache.catalina.core.ContainerBase.start
    (ContainerBase.java:1133)
    at org.apache.catalina.core.StandardEngine.start
    (StandardEngine.java:355)
    at org.apache.catalina.startup.Embedded.start
    (Embedded.java:995)
    at com.iplanet.ias.web.WebContainer.start
    (WebContainer.java:411)
    at com.iplanet.ias.web.WebContainer.startInstance
    (WebContainer.java:506)
    at com.iplanet.ias.server.J2EERunner.confPostInit
    (J2EERunner.java:161)
```

6. Run the `restoremodfiles` script to restore the 6.0 customized files that were backed up. For example, from the `/orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools` directory, type:

   `./restoremodfiles`

7. When prompted, enter the Sun ONE Identity Server administration password.

8. When prompted, enter the LDAP User password.

9. Restart the web container.

For example, to start the web container on a Sun ONE Portal Server system using a Sun ONE Web Server on server sesta, do the following:

a. Change directories to the directory containing the web server start and stop scripts.

```
cd /opt/SUNWwbsrvr/https-sesta
```

b. Type the following commands to stop and start the web server.

```
./stop
./start
```

To start the web container on a Sun ONE Portal Server system using an application server as its web container, use the appropriate application server start command.

10. Log in to the Sun ONE Identity Server Administration console.
11. Select the Service Configuration tab.
12. Select Core and then Organization Authentication Modules.
13. Click Add Membership and then Save.
14. From the Service Configuration tab, select Membership and specify amldapuser in the Password for Root User Bind.
15. From the Service Configuration tab, select Authentication Configuration and specify the Default Success URL.
16. Restart the web container.

You have now completed the upgrade process. Your Sun ONE Portal Server 6.2 system should be up and running with all of the previous existing data and customizations. You should access your Portal Desktop to verify that all the files have been upgraded successfully.

Localizing Online Help

After a successful upgrade, the channel help URLs will not have the locale in the URL. For example, the URLs will be similar to:

```
http://host/portal/docs/online_help/...
```

To localize the online help, update the display profile documents and provide the correct <ConditionalProperties> definition. For example, for the locale en_US use the <ConditionalProperties> definition to define the helpURL as:

```
<ConditionalProperties condition="locale" value="en">

   <ConditionalProperties condition="locale" value="US">

      <String name="helpURL"
         value="online_help/user_help/desktop/desktopTOC.html"
         advanced="true"/>

   </ConditionalProperties>

</ConditionalProperties>
```

After defining the `helpURL`, the Desktop Channel Help URL will be similar to:

```
http://host/portal/docs/en_US/online_help/...
```

Upgrading Sun ONE Portal Server: Secure Remote Access Pack

To complete the upgrade process for Sun ONE Portal Server: Secure Remote Access Pack installations, you need to do the following:

- Add a new key, `gateway.logging.password` in each of the `/etc/opt/SUNWps/platform.conf`. files with the value `amService-srapGateway` logging user's password.
- Change the value of `gateway.notification.url` from `/notification` to `notification`.
- Add the `netletproxy.port` and `rewriterproxy.port` properties and set to the appropriate values specified for `netletproxy` and `rewriterproxy`.
- The following entries in the `platform.conf` file are backward compatible; however, they will default to the 6.0 behavior if not modified:
    - `gateway.favicon`—Accepts the URL path to to which the Gateway will redirect requests for the favicon.ico file. Default is null.
    - `gateway.enable.customurl`—Indicates that there is a reverse proxy in front of the gateway. Default is false.
    - `gateway.enable.accelerator`—Indicates that there is SSL accelerator support on the gateway. Default is false.
    - `gateway.httpurl`—Specifies the custom http URL to be used as the gatewayURL by the rewriter when `gateway.enable.customurl` is enabled.

- gateway.httpsurl—Specifies the custom https URL to be used as the gatewayURL by the rewriter when gateway.enable.customurl is enabled.

You have now completed the upgrade process. Your Sun ONE Portal Server: Secure Remote Access Pack 6.2 system should be up and running with all of the previous existing data and customizations.

Part III

Upgrading from Portal Server 6.1 to Sun ONE Portal Server 6.2

Chapter 15

Planning the Upgrade from Sun Java System Portal Server 6.1 to Sun Java System Portal Server 6.2

This chapter provides an overview and discussion on how to plan for upgrading users, roles, organizations, components, templates, providers, channels, and properties files from Sun™ ONE Portal Server 6.1 to Sun™ ONE Portal Server 6.2.

This chapter contains the following sections:

- Overview of Upgrade Process
- Upgrading Certificates
- Upgrading Multiple Nodes

Overview of Upgrade Process

The Sun ONE Portal Server 6.2 product uses the Sun ONE Identity Server 6.1 service as its LDAP data store and management interface. The Sun ONE Portal Server 6.1 product used the Sun ONE Identity Server 6.0 service as its LDAP data store and management interface. Although the Sun ONE Identity Server 6.1 and the Sun ONE Identity Server 6.0 products are similar in structure and provide many of the same services, changes to the product require you to upgrade the Sun ONE Identity Server 6.0 to 6.1 before you can upgrade your Sun ONE Portal Server 6.1 software to the Sun ONE Portal Server 6.2 software.

NOTE The Sun ONE Portal Server 6.2 product can be deployed into a variety of web containers including the Sun™ ONE Web Server, the Sun™ ONE Application Server, the IBM Websphere Application Server, and the BEA Weblogic Application Server; however, when upgrading your Sun ONE Portal Server 6.1 installation into a Sun ONE Portal Server 6.2 installation you must deploy it into the same brand of web container.

These are the high-level steps to upgrade from Sun ONE Portal Server 6.1 to Sun ONE Portal Server 6.2. (Subsequent chapters provide step-by-step instructions for performing the actual process.)

1. Before you start the Sun ONE Portal Server upgrade process, shut down all instances on all nodes running the Sun ONE Portal Server software in your system. The upgrade process involves changes that do not allow the old Sun ONE Portal Server software version to be run at the same time as the new Sun ONE Portal Server software version.

2. Upgrade the Sun ONE Identity Server 6.0 software to the Sun ONE Identity Server 6.1 version on all nodes running the Sun ONE Portal Server software in your system. See Chapter 13, “Upgrading from Sun ONE Identity Server 6.0 to Sun™ ONE Identity Server 6.1” on page 217.

3. Upgrade each instance of the Sun ONE Portal Server software from the 6.1 release to the 6.2 release. See Chapter 16, “Upgrading Sun™ ONE Portal Server Software from Version 6.1 to Version 6.2” on page 239.

4. Restart your systems and allow user traffic.

Upgrading Certificates

If the name of the Sun ONE Portal Server 6.2 system is the same as the name of the Sun ONE Portal Server 6.1 system from which you are upgrading, you can use your existing certificates. If the names are not the same, you will need new certificates. For details on using certificates with Sun ONE Portal Server 6.2, see the *Sun ONE Portal Server* 6.2 *Administrator’s Guide* sections on configuring SSL.

Upgrading Multiple Nodes

Use the following information to help plan the upgrade process when upgrading multiple nodes:

- Your Sun ONE Portal Server 6.1 system can consist of one or more nodes running the Sun ONE Portal Server software, all of which share a single logical Sun™ ONE Directory Server for its LDAP data store. The upgrade process involves changes that do not allow the old Sun ONE Portal Server software version to be run at the same time as the new Sun ONE Portal Server software version. Therefore, before you start the upgrade process you must completely shut down all instances running the Sun ONE Portal Server software in your system.
- Once the LDAP data has been upgraded and the primary or first Sun ONE Portal Server node has been upgraded, you may restore traffic to your Sun ONE Portal Server system. Subsequent nodes may be upgraded while the primary Sun ONE Portal Server node is in operation.

Chapter 16

Upgrading Sun™ ONE Portal Server Software from Version 6.1 to Version 6.2

This chapter provides an overview and discussion on how to upgrade your Sun™ ONE Portal Server system from the 6.1 software release to the 6.2 software release.

This chapter contains the following sections:

- Overview of the Sun ONE Portal Server Upgrade Process
- Removing Sun ONE Portal Server 6.1 Packages and Backing Up Templates
- Installing Sun ONE Portal Server 6.2 Software
- Completing the Upgrade

Overview of the Sun ONE Portal Server Upgrade Process

The Sun ONE Portal Server 6.1 to 6.2 upgrade process involves the following high-level steps:

1. Running Sun ONE Portal Server 6.2 preupgrade script to remove the Sun ONE Portal Server 6.1 packages and backup any modified templates.
2. Installing the Sun ONE Portal Server 6.2 software using the Sun Java™ Enterprise System installer.
3. Completing the upgrade.

Removing Sun ONE Portal Server 6.1 Packages and Backing Up Templates

The Sun ONE Portal Server 6.2 software includes a script (`preupgrade`) that backs up any modified templates and removes all the Sun ONE Portal Server 6.1 packages. This script is located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of this script is located in `/orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc/Product/portal_svr/Tools` directory.

1. Log in to the machine and become superuser.

   You will need root access to uninstall the Sun ONE Portal Server.

2. Change directories to where the preupgrade script is located. For example,

   ```
   cd /orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools
   ```

3. Type:

   ```
   ./preupgrade
   ```

   The script processes each Sun ONE Portal Server 6.1 installed package for user modifications, saves any modifications to `/tmp/ps_backup`, and then removes each Sun ONE Portal Server 6.1 package.

Installing Sun ONE Portal Server 6.2 Software

Install the Sun ONE Portal Server 6.2 software using the Sun Java™ Enterprise System installer program located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of the installer is located in `/orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc` directory.

1. If you have not already done so, log in to the machine and become superuser.

   You will need root access to install the Sun ONE Portal Server.

2. Change directories to where the installer is located. For example,

   ```
   cd /orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc
   ```

3. If you used the `su` command to become root on your system, use the `xhost` command to grant access to your display. For example, use the following command to grant access to all users:

   ```
   xhost +
   ```

4. Type the following command to launch the installer:

   ```
   ./installer
   ```

5. At the Welcome screen, click Next.

6. Review the Software License Agreement and click Yes, Accept License to accept.

7. Specify the appropriate language support for the system and click Next.

8. On the Select Components tab, uncheck any previously checked boxes and check only the Sun ONE Portal Server 6.2 checkbox.

9. A warning message displays indicating that the Sun ONE Portal Server requires the Sun ONE Application Server or Sun ONE Web Server be installed or that Sun ONE Portal Server be configured to an existing Web Container. Click Continue.

10. On the Shared Component Upgrades Required panel, review the list of shared components that must be upgraded and click Next.

11. Verify the target installation directories and click Next.

12. If the Checking System Requirements panel indicates that any patches are needed, cancel the install and add any required patches. Otherwise, click Next to continue.

13. On the Configuration Type Panel, select Minimal Configuration and click Next.

NOTE You must install using the Minimal Configuration option to be able to restore the previous Portal Server configuration.

14. On the Ready to Install panel, click Next.

15. On the Installation Complete panel, select the appropriate button to view the Summary or the Install Log and click Close.

Completing the Upgrade

The Sun ONE Portal Server 6.2 software includes a script (`postupgrade`) that restores user modified templates from the Sun ONE Portal Server 6.1 installation. This script is located on the Sun Java™ Enterprise System media. For example, the Solaris SPARC version of this script is located in `/orion/s9orion/orion1of2_sparc.s9_orion/latest/Solaris_sparc/Product/portal_svr/Tools` directory. In addition to the running the `postupgrade` script, you must also verify some deployment information, deploy the Portal WAR file, and configure some security, login, and authentication information.

Perform the following steps to complete the upgrade process.

1. Open the `/etc/opt/SUNWps/PSConfig.properties` file with a text editor and verify that it contains correct values for the following variables:

   ```
   DEPLOY_DIR=
   DEPLOY_DOCROOT=
   DEPLOY_DOMAIN=
   ```

   For example, on systems deployed in the default directory of a Sun ONE Web Server installation, the settings for `DEPLOY_DIR`, and `DEPLOY_DOCROOT` would be

   ```
   DEPLOY_DIR=/opt/SUNWwbsrvr
   DEPLOY_DOCROOT=/opt/SUNWwbsrvr/docs
   ```

2. Type the following command to deploy the Portal WAR file

   ```
   /opt/SUNWps/bin/deploy
   ```

   a. When prompted into which instance to deploy, select the instance or select "." for all instances and press Enter to deploy the Portal WAR file.

   b. When prompted for the deployment URI, enter `/portal`.

3. Change directories to where the postupgrade script is located. For example,

   ```
   cd /orion/pointproducts/Solaris_sparc/Product/portal_svr/Tools
   ```

4. Type:

   ```
   ./postupgrade
   ```

5. When prompted, enter the Sun ONE Identity Server administration password.

6. When prompted, enter the LDAP User password.

Perform the following steps after running the `postupgrade` script:

1. Open the /*JDK_dir*/jre/lib/java.security file and verify that it contains the following line:

   ```
   login.configuration.provider=com.sun.identity.authentication.config.
   AMConfiguration
   ```

2. Restart the web container.

   For example, to start the web container on a Sun ONE Portal Server system using a Sun ONE Web Server on server sesta, do the following:

   a. Change directories to the directory containing the web server start and stop scripts.

      ```
      cd /opt/SUNWwbsrvr/https-sesta
      ```

   b. Type the following commands to stop and start the web server.

      ```
      ./stop
      ./start
      ```

      To start the web container on a Sun ONE Portal Server system using an application server as its web container, use the appropriate application server start command.

3. Log in to the Sun ONE Identity Server Administration console.
4. Select the Service Configuration tab.
5. Select Core and then Organization Authentication Modules.
6. Click Add Membership and then Save.
7. From the Service Configuration tab, select Membership and specify amldapuser in the Password for Root User Bind.
8. Click Save.
9. From the Service Configuration tab, select Core under Authentication Configuration and specify the Login Success URL as /portal/dt.
10. Click Save.
11. Restart the web container.

You have now completed the upgrade process. Your Sun ONE Portal Server 6.2 system should be up and running with all of the previous existing data and customizations. You should access your Portal Desktop to verify that all the files have been upgraded successfully.

Localizing Online Help

After a successful upgrade, the channel help URLs will not have the locale in the URL. For example, the URLs will be similar to:

```
http://host/portal/docs/online_help/...
```

To localize the online help, update the display profile documents and provide the correct <ConditionalProperties> definition. For example, for the locale en_US use the <ConditionalProperties> definition to define the helpURL as:

```
<ConditionalProperties condition="locale" value="en">

    <ConditionalProperties condition="locale" value="US">

        <String name="helpURL"
            value="online_help/user_help/desktop/desktopTOC.html"
            advanced="true"/>

    </ConditionalProperties>

</ConditionalProperties>
```

After defining the helpURL, the Desktop Channel Help URL will be similar to:

```
http://host/portal/docs/en_US/online_help/...
```

Part IV

Upgrading from Sun ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2

Chapter 17

Planning the Upgrade from Sun ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2

This chapter provides an overview and discussion on how to plan for upgrading users, roles, organizations, components, templates, providers, channels, and properties files from Sun™ ONE Portal Server 6.2 to Sun Java™ System Portal Server 6 2004Q2.

This chapter contains the following sections:

- Overview of Upgrade Process
- Installing Portal Server and Identity Server on Separate Machines
- Upgrading Multiple Nodes

Overview of Upgrade Process

The Sun Java System Portal Server 6 2004Q2 product uses the Sun Java System Identity Server 2004Q2 service as its LDAP data store and management interface. The Sun ONE Portal Server 6.2 product used the Sun ONE Identity Server 6.1 service as its LDAP data store and management interface. Although the Sun Java System Identity Server 2004Q2 and the Sun ONE Identity Server 6.1 products are similar in structure and provide many of the same services, changes to the product require you to upgrade the Sun ONE Identity Server 6.1 to Sun Java System Identity Server 2004Q2 before you can upgrade your Sun ONE Portal Server 6.2 software to the Sun Java System Portal Server 6 2004Q2 software.

NOTE The Sun ONE Portal Server 6.2 product can be deployed into a variety of web containers including the Sun™ ONE Web Server, the Sun™ ONE Application Server, the IBM Websphere Application Server, and the BEA Weblogic Application Server; however, when upgrading your Sun ONE Portal Server 6.2 installation into a Sun Java System Portal Server 6 2004Q2 installation you must deploy it into the same brand of web container. In addition, the IBM Websphere Application Server and the BEA Weblogic Application Server must also run on a separate machine than where Sun Java System Identity Server 2004Q2 is installed and configured. See "Installing Portal Server and Identity Server on Separate Machines" on page 249 for more information.

These are the high-level steps to upgrade from Sun ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2. (Subsequent chapters provide step-by-step instructions for performing the actual process.)

1. Before you upgrade, back up any web container customized files related to Portal Server 6.2, including:
    - Customized console JSP pages
    - Customized authentication JSP pages
    - JAR files for customized modules
    - Customized sample Portal Server desktop

CAUTION The SUN ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2 upgrade can take a considerable time, based on the complexity of the Portal Server 6.2 files that are deployed. Do not upgrade a Portal Server 6.2 system that is already deployed into production until a test deployment of that production system has first been upgraded successfully.

CAUTION If you have made extensive customizations to Portal Server 6.2 files, you should contain Sun technical support or professional services for assistance.

2. Before you start the Sun ONE Portal Server upgrade process, shut down all instances on all nodes running the Sun ONE Portal Server software in your system. The upgrade process involves changes that do not allow the old Sun ONE Portal Server software version to be run at the same time as the new Sun Java System Portal Server software version.

3. Upgrade all nodes running the Sun ONE Portal Server software in your system using the Java Enterprise System 2 installer to install shared components. See the *Java Enterprise System 2004Q2 Installation Guide* for more information.

4. Upgrade the Sun ONE Identity Server 6.1 software to use shared components for the Sun Java System Identity Server 2004Q2 version on all nodes running the Sun ONE Portal Server software in your system using the Java Enterprise System 2 installer. See the Java Enterprise System 2004Q2 Installation Guide for more information on upgrading Identity Server.

5. Apply to each node running the Sun ONE Portal Server 6.2 software patches specific to Sun Java System Portal Server 6 2004Q2. See Chapter 18, "Upgrading from Sun ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2" on page 251.

6. Restart your systems and allow user traffic.

Installing Portal Server and Identity Server on Separate Machines

For Java Enterprise System 2004Q2, Sun Java System Portal Server can be installed and configured to run with either:

- Sun Java System Identity Server is installed and configured in the same installation session on the same physical machine
- Sun Java System Identity Server is previously installed and configured on a separate machine

By default, Sun Java System Portal Server 6 2004Q2 and Sun Java System Identity Server 2004Q2, Sun Java System Directory Server 2004Q2, and a web container are installed on the same physical machine, with the following subcomponents selected for Sun Java System Identity Server:

- Identity Management and Policy Services Core
- Identity Server Administration Console
- Common Domain Services for Federation Management
- Identity Server SDK

You can install Sun Java System Portal server with a third-party web container (such as BEA WebLogic or IBM Websphere); however, Sun Java System Identity Server 2004Q2 supports third-party web containers differently. In this case, upgrading Portal Server for a third-party web container will require an additional machine and a reconfiguration of Portal Server and Identity Server software. See“Upgrading Portal Server on a Third-Party Web Container” on page 258 for more information on third-party web container support

Upgrading Multiple Nodes

Use the following information to help plan the upgrade process when upgrading multiple nodes:

- Your upgraded Sun Java System Portal Server 6 2004Q2 system can consist of one or more nodes running the Sun ONE Portal Server 6.2 software, all of which share a single logical Sun™ ONE Directory Server for its LDAP data store. The upgrade process to Sun Java Enterprise System 2004Q2 involves changes that do not allow the Sun ONE Portal Server 6.2 software version to be run at the same time as the new Sun Java System Portal Server2004Q2 software version. Therefore, before you start the upgrade process you must completely shut down all instances running the Sun ONE Portal Server 6.2 software in your system.
- Once the LDAP data has been upgraded and the primary or first Sun ONE Portal Server 6.2 node has been upgraded to Sun Java System Portal Server 6 2004Q2, you may restore traffic to your upgraded Sun Java System Portal Server 6 2004Q2 system. Subsequent nodes may be upgraded while the primary Sun Java System Portal Server 6 2004Q2 node is in operation.

Chapter 18

Upgrading from Sun ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2

This chapter contains procedures for upgrading from Sun ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2. It contains the following topics:

- "Accessing Patches" on page 251
- "Backing up Web Container Customized Files" on page 252
- "Upgrading the Sun Web Container Software" on page 253
- "Upgrading Identity Server" on page 253
- "Upgrading Portal Server" on page 256

Accessing Patches

Upgrading Portal Server is done using patches. Download the patches listed in the following table from SunSolve (patch revision should be the same as listed in the table or newer).

Table 18-1 Sun Java System Portal Server 2004Q2 Patches

Functional Area to Patch	Patch ID for Solaris SPARC	Patch ID for Solaris x86
Identity Server administration console	117769-01	117770-01
Portal Server configurator	118377-01	118378-01
Portal Server	116736-25	116737-25

Table 18-1 Sun Java System Portal Server 2004Q2 Patches

Functional Area to Patch	Patch ID for Solaris SPARC	Patch ID for Solaris x86
Secure Remote Access Support	116749-25	116750-25
Gateway	116738-23	116739-23
Rewriter Proxy	116742-23	116743-23
Netlet Proxy	116740-23	116741-23
Localization	117012-05	117013-05
	117105-05	117105-05

Backing up Web Container Customized Files

Before you upgrade, back up any web container customized files related to Portal Server 6.2, including:

- Customized console JSP pages
- Customized authentication JSP pages
- JAR files for customized modules
- Customized sample Portal Server desktop

CAUTION If you have made extensive customizations to Portal Server 6.2 files, you should contact Sun technical support or professional services for assistance.

CAUTION The SUN ONE Portal Server 6.2 to Sun Java System Portal Server 6 2004Q2 upgrade can take a considerable time, based on the complexity of the Portal Server 6.2 files that are deployed. Do not upgrade a Portal Server 6.2 system that is already deployed into production until a test deployment of that production system has first been upgraded successfully.

It is recommend that you make a list of your customizations so you can redo them after you upgrade and then verify that they work correctly. The following directories should be backed up:

- `/opt/SUNWps` (assuming a default installation location)
- `/etc/opt/SUNWps`
- `/var/opt/SUNWps`

Upgrading the Sun Web Container Software

The Java Enterprise System 2004Q2 release requires that the Identity Server instance be run on Sun's Web Server or Application Server (such as Web Server 6.1 SP2 or Application Server 7.0 Update 3) on the same system. If you are using an older version, you must upgrade the web container software before you can upgrade to Java Enterprise System 2004Q2 release.

For information about upgrading Sun's Web Server or Application Server software, refer to the respective web container documentation:

- For Web Server 6.1 SP2 see:

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

- For Application Server 7.0 Update 3, see:

  `http://docs.sun.com/coll/s1_asseu3_en`

Also, if you saved any customization files under "Backing up Web Container Customized Files" on page 252, you will need to redo the customizations after you upgrade the web container.

Upgrading Identity Server

Portal Server upgrade has a dependency on Identity Server. Prior to upgrading Portal Server, upgrade all systems running Identity Server to the Java Enterprise System 2004Q2 version.

Refer to the *Java Enterprise System 2003Q4 Installation Guide* for more a more detailed description of the Identity Server upgrade.

Using Web Server 6 2004Q2 as a Web Container

If you are using Sun Java System Web Server as a web container, you must run the install the Identity Server administration console patch.

1. Install Identity Server 2004Q2.

   See the *Sun Java Enterprise System 2004Q2 Installation Guide* for more information.

2. Run the following command to install the Identity Server administration console patch:

   ```
   > patchadd 117769-01
   ```

   Patch 117769-01 is intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system).

Backing up the Administration Console Help Files

The Portal Server help files that are used with the Identity Server administration console must be backed up before the Identity Server 6.1 software is upgraded and restored after the Identity Server 2004Q2 software is installed.

1. Copy the contents of the online help directory to a temporary directory, such as:

   ```
   cp -r /installation-directory/SUNWam/public_html/online_help/docs_en_US/ps /tmp
   ```

2. Run the Identity Server pre-upgrade script.

   See the *Sun Java Enterprise System 2004Q2 Installation Guide* for more information.

3. Install Identity Server 2004Q2.

   See the *Sun Java Enterprise System 2004Q2 Installation Guide* for more information.

4. Copy the contents of the temporary directory to the online help directory, such as:

```
cp -r /tmp/ps
/installation-directory/SUNWam/public_html/online_help/docs_en_US/ps
```

Enabling Client Detection

In order to enable client detection, change the Identity Server Client Detection global attributes as follows:

1. Access the Identity Server 2004Q2 console using the following URL:

   `http://host-name.domain-name:port/amconsole`

   where *host-name.domain-name:port* is the fully qualified host name and port of the web container you are using.

2. When the Identity Server login page appears, log in as `amadmin`.

3. On the console, click the Service Configuration tab.

   The console displays the Service Configuration options in the navigation frame.

4. In the navigation frame under Service Configuration, click Client Detection.

5. For Client Detection, set the following items in the data frame:

   a. Set the Client Detection Class global attribute to `com.sun.mobile.cdm.FEDIClientDetector`

   b. Click the Enable Client Detection check box.

6. Click Save.

Verifying the Upgrade

If you customized your Identity Server 6.1 installation, you must manually redo the customizations in your new Identity Server 2004Q2 installation.

Here are several ways to verify that the upgrade was successful:

- Access the Identity Server 2004Q2 console using the following URL:

  `http://host-name.domain-name:port/amconsole` where *host-name.domain-name*:*port* is the fully qualified host name and port of the web container you are using.

When the Identity Server login page appears, log in as amadmin. Click the "Service Configuration" tab. If the new Identity Server 2004Q2 services such as "Discovery Service" and "Liberty and Personal Profile Service" are available, the upgrade of Identity Server on the specific web container should be successful.

- Review the status of the upgrade by checking the following log files in the `/var/sadm/install/logs` directory:
  - Pre-upgrade script (`pre61to62upgrade`):

    `Sun_Java_System_Identity_Server_upgrade_log.`*timestamp*
  - Sun Java Enterprise System installer:

    `Java_Shared_Component_Install.`*timestamp*
    `Java_Enterprise_System_install.A`*timestamp*
    `Java_Enterprise_System_install.B`*timestamp*
    `Java_Enterprise_System_Summary_Report_install.`*timestamp*
  - Post-upgrade script (`Upgrade61DitTo62`):

    `Sun_Java_System_Identity_Server_upgrade_dit_log.`*timestamp*

Upgrading Portal Server

This section contains procedures for upgrading Portal Server. It also contains upgrade procedures for products associated with Portal Server. Follow the procedures in the following sections as appropriate:

- "Upgrading Portal Server on a Sun Web Container" on page 257
- "Upgrading Portal Server on a Third-Party Web Container" on page 258
- "Upgrading the Gateway" on page 264
- "Upgrading the Rewriter Proxy" on page 265
- "Upgrading the Netlet Proxy" on page 265
- "Upgrading Localization" on page 266

Upgrading Portal Server on a Sun Web Container

1. Log in as root.

NOTE If Mobile Access 6.2 support was installed on the Java Enterprise System 2003Q4 system, you must remove it prior to upgrading to Portal Server 6 2004Q2. To see if Mobile Access 6.2 support was installed, enter the following command:

```
> pkginfo -l | SUNWpswp
```

---

2. To remove Mobile Access 6.2, run the uninstaller that was generated when Mobile Access was installed. It is located in the installation directory chosen at install time. Also, run the following commands to clean the Mobile Access directories and files:

   a. Locate the `Tools` directory on the CD or download directory.

   b. Copy the `mobileaccess.tar.gz` file to a local hard drive.

   c. Unzip and untar the contents of the mobileaccess.tar.gz file to a local directory.

      Note that the `gnuzip` and `gnutar` commands should be used.

   d. Run the following command from the local directory:

```
> ./setup
```

3. Run the following commands to install the patches:

```
> patchadd 118377-01
> patchadd 116736-25
> patchadd 116749-25
```

Patches 118377-01, 116736-25 and 116749-25 are intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system).

NOTE If you want to upgrade to later versions of Portal Server, you will also need to install patch 118195-03 (Solaris SPARC) or 118196-03 (Solaris x86) to ensure compatibility with the Linux version.

---

4. Run the following commands to upgrade the Portal Server (with `/opt/SUNWps` as the default installation directory):

```
> cd /opt/SUNWps/lib
```

```
> ./upgradePS
```

```
> ./upgradeSRA
```

The `upgradeSRA` script is needed if Secure Remote Access is installed. These scripts will prompt you for passwords.

CAUTION Once the `upgradePS` or `upgradeSRA` scripts are run, any Portal Server patches that were applied cannot be backed out.

5. Follow these steps to upgrade Mobile Access support):

   a. Locate the `Tools` directory on the CD or download directory.

   b. Copy the `mobileaccess.tar.gz` file to a local hard drive.

   c. Unzip and untar the contents of the mobileaccess.tar.gz file to a local directory.

   Note that the `gnuzip` and `gnutar` commands should be used.

   d. Run the following command from the local directory:

```
> ./setup
```

CAUTION This step is required to upgrading Portal Server, regardless of whether Mobile Access 6.2 is actually be used.

Upgrading Portal Server on a Third-Party Web Container

You can install Sun Java System Portal server with a third-party web container (such as BEA WebLogic or IBM Websphere); however, Sun Java System Identity Server 2004Q2 supports third-party web containers differently. In this case, upgrading Portal Server for a third-party web container will require an additional machine and a reconfiguration of Portal Server and Identity Server software (or IS/PS separation).

Upgrading Using a WebLogic Web Container

The following lists describe the Portal Server configurations before and after the upgrade process. Use this information to determine the upgrade requirements for your Portal Server configuration.

Portal Server Configuration Before Upgrade (Machine 1)

- Sun ONE Portal Server 6.2
- Sun ONE Identity Server 6.1
- Sun ONE Directory Server 5.2
- WebLogic Application Server 6.1

Portal Server Configuration After Upgrade (Machine 1)

- Sun Java System Portal Server 6 2004Q2
- Sun Java System Identity Server 2004Q2 SDK
- Sun Java System Directory Server 5 2004Q2
- WebLogic Application Server 8.1

Portal Server Configuration After Upgrade (Machine 2)

- Sun Java System Identity Server 2004Q2
- Sun Java System Web Server or Sun Java System Application Server web container

To upgrade Sun ONE Portal Server 6.2 running with a WebLogic 6.1 SP4 container:

1. Stop the WebLogic 6.1 software running on Machine 1.
2. Verify that the Directory Server is running on Machine 1.
3. Run the Identity Server pre-upgrade script on Machine 1 (located on the Java Enterprise System CD):

   ```
   ./pre61to62upgrade
   ```

4. Restart the Directory Server on Machine 1.
5. Install the Identity Server 2004Q2 software and either the Sun Java System Web Server or Sun Java System Application Server software on Machine 2.

6. Run the Identity Server DIT upgrade script on Machine 2 (located at `/opt/SUNWam/migration/61to62/scripts`):

   ```
   ./Upgrade61DitTo62
   ```

7. Restart the Directory Server on Machine 1.

8. Start the Sun Java System Web Server or Sun Java System Application Server software on Machine 2.

9. Install the WebLogic 8.1 software on Machine 1.

NOTE Make sure that the port and instance name for the WebLogic 8.1 installation matches the port and instance name for the old WebLogic 6.1 installation.

10. Start the WebLogic 8.1 software on Machine 1.

11. Install the Identity Server SDK software on Machine 1.

    Specify Machine 1 as the Directory Server to be used and Machine 2 as the Identity Server to be used.

12. Run the following command to install the Identity Server administration console patch:

    ```
    > patchadd 117769-01
    ```

    Patch 117769-01 is intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system).

13. Run the following commands to install the Portal Server patches:

    ```
    > patchadd 118377-01
    ```

    ```
    > patchadd 116736-25
    ```

    ```
    > patchadd 116749-25
    ```

    Patches 118377-01, 116736-25 and 116749-25 are intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system).

NOTE If you want to upgrade to later versions of Portal Server, you will also need to install patch 118195-03 (Solaris SPARC) or 118196-03 (Solaris x86) to ensure compatibility with the Linux version.

14. Run the following command to upgrade the Portal Server on Machine 1 (with /opt/SUNWps as the default installation directory):

```
> cd /opt/SUNWps/lib
```

```
> ./upgradePS
```

The script will prompt you if you want to modify the following parameters in the /etc/opt/SUNWps/PSconfig.properties file (the user can specify values other than the default values that are listed):

Parameter	Default Value for WebLogic 6.1 SP4	Default Value for WebLogic 8.1 SP2
DEPLOY_ADMIN	system	weblogic
DEPLOY_DIR	/bea	/usr/local/bea
DEPLOY_DOCROOT	config/mydomain/applications/DefaultWebApp	Not Applicable
DEPLOY_JDK_DIR	/bea/jdk141_05	/usr/local/bea/jdk141_05

CAUTION Once the upgradePS script isrun, any Portal Server patches that were applied cannot be backed out.

15. Run the following command if Secure Remote Access is installed:

```
> ./upgradeSRA
```

CAUTION Once the upgradeSRA script is run, any Portal Server patches that were applied cannot be backed out.

16. Restart the Directory Server and WebLogic 8.1 software on Machine 1.

Upgrading Using a WebSphere Web Container

The following lists describe the Portal Server configurations before and after the upgrade process. Use this information to determine the upgrade requirements for your Portal Server configuration.

Portal Server Configuration Before Upgrade (Machine 1)

- Sun ONE Portal Server 6.2
- Sun ONE Identity Server 6.1
- Sun ONE Directory Server 5.2
- WebSphere Application Server 4.0.5

Portal Server Configuration After Upgrade (Machine 1)

- Sun Java System Portal Server 6 2004Q2
- Sun Java System Identity Server 2004Q2 SDK
- Sun Java System Directory Server 5 2004Q2
- WebSphere Application Server 5.1

Portal Server Configuration After Upgrade (Machine 2)

- Sun Java System Identity Server 2004Q2
- Sun Java System Web Server or Sun Java System Application Server web container

To upgrade Sun ONE Portal Server 6.2 running with a WebSphere 4.0.5 container:

1. Stop the WebSphere 4.0.5 software running on Machine 1.
2. Verify that the Directory Server is running on Machine 1.
3. Run the Identity Server pre-upgrade script on Machine 1 (located on the Java Enterprise System CD):

   ```
   ./pre61to62upgrade
   ```

4. Restart the Directory Server on Machine 1.
5. Install the Identity Server 2004Q2 software and either the Sun Java System Web Server or Sun Java System Application Server software on Machine 2.
6. Run the Identity Server DIT upgrade script on Machine 2 (located at `/opt/SUNWam/migration/61to62/scripts`):

   ```
   ./Upgrade61DitTo62
   ```

7. Restart the Directory Server on Machine 1.
8. Start the Sun Java System Web Server or Sun Java System Application Server software on Machine 2.
9. Install the WebSphere 5.1 software on Machine 1.

NOTE Make sure that the port and instance name for the WebSphere 5.1 installation matches the port and instance name for the old WebSphere 4.0.5 installation.

10. Start the WebSphere 5.1 software on Machine 1.

11. Install the Identity Server SDK software on Machine 1

    Specify Machine 1 as the Directory Server to be used and Machine 2 as the Identity Server to be used.

12. Run the following command to install the Identity Server administration console patch:

    ```
    > patchadd 117769-01
    ```

    Patch 117769-01 is intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system).

13. Run the following commands to install the Portal Server patches:

    ```
    > patchadd 118377-01
    ```

    ```
    > patchadd 116736-25
    ```

    ```
    > patchadd 116749-25
    ```

    Patches 118-377-01, 116736-25 and 116749-25 are intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system).

NOTE If you want to upgrade to later versions of Portal Server, you will also need to install patch 118195-03 (Solaris SPARC) or 118196-03 (Solaris x86) to ensure compatibility with the Linux version.

14. Run the following command to upgrade the Portal Server on Machine 1 (with `/opt/SUNWps` as the default installation directory):

    ```
    > cd /opt/SUNWps/lib
    ```

    ```
    > ./upgradePS
    ```

    The script will prompt you if you want to modify the following parameters in the `/etc/opt/SUNWps/PSconfig.properties` file (the user can specify values other than the default values that are listed):

Parameter	Default Value for WebSphere 4.0.5	Default Value for WebSphere 5.1
`DEPLOY_CELL`	Not Applicable	`DefaultNode`
`DEPLOY_DOCROOT`	`/opt/IBMHTTPD/htdoc/en_US`	`/opt/WebSphere/AppServer/installedApps/DefaultNode/DefaultApplication.ear/DefaultWebApplication`
`DEPLOY_INSTANCE`	`Server`	`server1`
`DEPLOY_JDK_DIR`	Not Applicable	`/opt/WebSphere/AppServer/Java`
`DEPLOY_NODE`	*server-name*	`DefaultNode`

CAUTION Once the `upgradePS` script is run, any Portal Server patches that were applied cannot be backed out.

15. Run the following command if Secure Remote Access is installed:

```
> ./upgradeSRA
```

CAUTION Once the `upgradeSRA` script is run, any Portal Server patches that were applied cannot be backed out.

16. Restart the Directory Server and WebSphere 5.1 software on Machine 1.

Upgrading the Gateway

You must upgrade Portal Server and Secure Remote Access before you can upgrade the gateway. See the following for more information:

- "Upgrading Portal Server on a Sun Web Container" on page 257
- "Upgrading Portal Server on a Third-Party Web Container" on page 258

If the Gateway is installed on a separate machine from the Portal Server machine, run the following command:

```
> patchadd 118377-01
```

Run the following command to upgrade the Gateway:

```
> patchadd 116738-23
```

Patch 117377-01 and 116738-23 are intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system).

Upgrading the Rewriter Proxy

You must upgrade Portal Server and Secure Remote Access before you can upgrade the Rewriter proxy. See the following for more information:

- "Upgrading Portal Server on a Sun Web Container" on page 257
- "Upgrading Portal Server on a Third-Party Web Container" on page 258

If the Rewriter Proxy is installed on a separate machine from the Portal Server machine, run the following command:

```
> patchadd 118377-01
```

Run the following command to upgrade the Rewriter Proxy:

```
> patchadd 116742-23
```

Patch 118377-01 and 116742-23 are intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system).

Upgrading the Netlet Proxy

You must upgrade Portal Server and Secure Remote Access before you can upgrade the Netlet proxy. See the following for more information:

- "Upgrading Portal Server on a Sun Web Container" on page 257
- "Upgrading Portal Server on a Third-Party Web Container" on page 258

If the Netlet Proxy is installed on a separate machine from the Portal Server machine, run the following command:

```
> patchadd 118377-01
```

Run the following command to upgrade the netlet proxy:

```
> patchadd 116740-23
```

Patch 118377-01 and 116740-23 are intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system).

Upgrading Localization

Run the following commands:

```
> patchadd 117012-05
```

```
> patchadd 117105-05
```

Patch 117012-05 is intended for a Solaris SPARC system (refer to "Sun Java System Portal Server 2004Q2 Patches" on page 251 for patch information for a Solaris x86 system). Patch 117105-05 is intended for both Solaris SPARC and x86 systems.

Appendix A

Tool Design

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

This appendix includes information, options or parameters, and usage information for the Sun ONE Portal Server 3.0 Data Migration Tool Suite and the Sun™ ONE Compass Server Migration Tools. This appendix contains the following sections:

- Sun ONE Portal Server 3.0 Data Migration Tool Suite
- Flow of Sun ONE Portal Server 3.0 Data Migration Tool Suite
- Sun ONE Compass Server Migration Tools

Sun ONE Portal Server 3.0 Data Migration Tool Suite

This section provides the tool design for the export, conversion, and import tools used to migrate from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2.

Export Tool Design

Synopsis

./exportps [-a] [-p *LDAP passphrase*] [*ExportDir*]

Description

The export tool copies the following Sun ONE Portal Server 3.0 components to a directory or a `.tar` file:

- Templates
- JSPs
- Resource bundles
- LDIF dump of the LDAP database
- XML exports of domains, roles and components

Options

Table A-1 describes the options available for the `exportps` command. This two-column table lists the options in the first column and their descriptions in the second column.

Table A-1 Options for the `exportps` command

Option	Description
`-a`	Runs all modules without a menu. To see the export menu, do not use the `-a` option.
`-p` *LDAP passphrase*	Sets the LDAP passphrase in order to avoid the LDAP passphrase prompt. If you do not use the `-p` *LDAP passphrase* option, the tool will prompt you for the LDAP passphrase.
ExportDir	Specifies the directory where the information is to be saved. If you do not use the *ExportDir* option, the tool prompts you for the export directory.

Usage

If you run the `exportps` command without the *ExportDir* option, the script exports the data to `/tmp/psExport`. In addition to the options listed in the Options section, you can also get help and version information by issuing the following commands:

- **`./exportps --help`**

  This command gives the following output:

  ```
  exportps is used to save an iPlanet Portal Server 3.0
  (Service Pack 3a or 4) system to disk


    USAGE
      exportps [-a] [-p <passphrase>] [<directory>]


    OPTIONS
      -a Export all modules (no menu)
      -p <passphrase> Set LDAP root DN passphrase
      <directory> where <directory> is place to save exported data
  ```

- **`./exportps --version`**

  This command gives the following output:

  ```
  Name: Portal Server export utility
  Version: 6.2
  Product: Portal Server, Hana release
  Legal-notices:
  Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
  PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license
  terms.
  ```

Conversion Tool Design

Synopsis

`./convertps [-a] [-f] [-i` *ExportDir*`] [-o` *ImportDir*`]`

Description

The results of the `convertps` command are converted to files to be used by the `importps` command for loading Sun ONE Portal Server 3.0 customizations into Sun ONE Portal Server 6.2.

Options

Table A-2 describes the options available for the `convertps` command. This two-column table lists the options in the first column and their descriptions in the second column.

Table A-2 Options for the `convertps` command

Option	Description
`-a`	Runs all modules without a menu. To see the conversion menu, do not use the `-a` option.
`-f`	Converts Sun ONE Portal Server 3.0 roles toSun ONE Identity Server roles and places users under the organization. This option retains the multiple role to user assignment similar to Sun ONE Portal Server 3.0, but does not retain the hierarchical role functionality. All roles are created under the organization and are not prioritized. The display profile documents are prioritized and merged accordingly. To see if the roles have been migrated to roles, view the roles under the organization in the admin console. If you do not use the `-f` option, the conversion tool converts Sun ONE Portal Server 3.0 roles toSun ONE Identity Server suborganizations and places users under the suborganization. Without the `-f` option, the conversion tool retains the hierarchical functionality and customizations from Sun ONE Portal Server 3.0. The disadvantage is that it is difficult to move users from one suborganization to another. To see if the roles have been migrated to suborganizations, view the hierarchy in the admin console.
`-i` *ExportDir*	Specifies the input directory for the conversion. The input directory is the directory which contains the exported data created by the export tool. The conversion tool searches for export data in `/tmp/psExport` unless you specify a different input directory using the `-i` *ExportDir* option. If you specify a directory which does not contain data exported with the export tool, the conversion tool notifies you that the directory does not have export data and prompts you to enter a valid export directory.

Table A-2 Options for the `convertps` command *(Continued)*

Option	Description
`-o` *ImportDir*	Specifies the output directory for the conversion tool. The output directory is the directory used by the import tool. If you choose an import directory which already contains converted data, the conversion tool notifies you that modifying an existing migration may render import data unusable and prompts you to delete the import directory.

Usage

If you run the `convertps` command without the `-i` *ExportDir* and `-o` *ImportDir* options, the conversion tool will check for the `/tmp/ipsExport` directory. If the tool finds valid exported data, it will store the converted data in `/tmp/psImport`. In addition to the options listed in the Options section, you can also get help and version information by issuing the following commands:

- **`./convertps --help`**

  This command gives the following output:

  ```
  convertps is used to migrate iPlanet Portal Server 3.0

  (Service Pack 3a or 4) data to files that can be uploaded into

    a Portal Server 6.2 system

    OPTIONS

      -a Convert all modules (no menu)

      -f Converts roles to roles, instead of roles to
         sub-organizations

      -i <directory> where <directory> is iPS export directory

      -o <directory> where <directory> is place to save migrated files

         for future upload using importps
  ```

- **`./convertps --version`**

  This command gives the following output:

  ```
  Name: Portal Server conversion utility

  Version: 6.2

  Product: Portal Server, Hana release
  ```

```
Legal-notices:

Copyright 2003 Sun Microsystems, Inc.  All rights reserved.

PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license
terms.
```

Import Tool Design

Synopsis

`./importps [-a] [-k] [-m] [`*ImportDir*`]`

Description

The import tool imports into the Sun ONE Portal Server 6.2 system the data converted with the conversion tool.

Options

Table A-3 describes the options available for the `importps` command. This two-column table lists the options in the first column and their descriptions in the second column.

Table A-3 Options for the `importps` command

Option	Description
`-a`	Runs all modules without a menu. To see the import menu, do not use the `-a` option.
`-k`	Specifies that the tool does not overwrite existing users. When you run `importps` without the `-k` option, if it encounters a conflict with an existing user, it will delete the existing user and import the new user entry defined in *ImportDir*`/ldap/user.ldif.` If you run `importps -k`, if it encounters a conflict with an existing user, it leaves the existing user and places the rejected user entry in *ImportDir*`/ldap/rejected_users.ldif`. If the user has a display profile document in the *ImportDir*`/dp/user/` directory, the display profile file may change. The `-k` option does not affect importing user display profile documents. The `-k` option only affects whether an existing LDAP user’s LDIF record is updated. The updates to the LDIF record include everything except display profile updates.

Table A-3 Options for the `importps` command *(Continued)*

Option	Description
`-m`	Merges display profile documents. When you run `importps` without the `-m` option, it overwrites any existing display profile documents it finds. This means that the root suffix, organization, suborganization, and user level display profile documents, if they exist, are overwritten. If you run `importps -m`, when the tool encounters a conflict with an existing display profile document it will merge it.
ImportDir	Specifies the directory where the converted data is located. If the directory does not exist or if the directory does not contain valid converted data, the import tool prompts you to enter a directory containing the converted data to be imported. If you do not specify an import directory, the import tool will search `/tmp/psImport`.

Usage

If you run the `importps` command without the *ImportDir* option, the script looks in `/tmp/psImport` for converted data. In addition to the options listed in the Options section, you can also get help and version information by issuing the following commands:

- **`./importps --help`**

  This command gives the following output:

  ```
  importps is used to upload converted iPlanet Portal Server

  3.0 (Service Pack 3a or 4) customizations into Portal Server 6.2


    USAGE

      importps [-a] [-k] [-m] [<directory>]


    OPTIONS

      -a Import all modules (no menu)

      -k Do not overwrite existing users

      -m Merge display profile documents

      <directory> where <directory> is location of converted data
  ```

- **./importps --version**

  ```
  Name: Portal Server import utility

  Version: 6.2

  Product: Portal Server, Hana release

  Legal-notices:

  Copyright 2003 Sun Microsystems, Inc.  All rights reserved.

  PROPRIETARY/CONFIDENTIAL.  Use of this product is subject to license
  terms.
  ```

Flow of Sun ONE Portal Server 3.0 Data Migration Tool Suite

Export Tool

The export tool adds to *ExportDir*/`templates.tar`:

- `/etc/opt/SUNWips/desktop/`
- changed files in `/etc/opt/SUNWips/auth/`
- *BaseDir*/`SUNWips/locale/`
- *BaseDir*/`SUNWips/public_html` except for `admin`, `launch.html`, `netlet`, `banner.html`, `index.html`, `netfile`, `netmail`, and `third_party`
- files with fully-qualified path names listed in *BaseDir*/`SUNWps/migration/modules/desktop/migrate_files.txt`

Conversion Tool

The conversion tool:

- extracts *ExportDir*/`templates.tar` into `/tmp/.temp_desktop/exportData/`
- does the conversion of templates, JSPs, and resource bundles in `/tmp/.temp_desktop/exportData/`

- saves converted data in `/tmp/.temp_desktop/importData/`
- adds to *ImportDir*`/desktop.tar` the contents of `/tmp/.temp_desktop/importData/desktop/`
- adds to *ImportDir*`/locale.tar` the contents of `/tmp/.temp_desktop/importData/locale/`
- adds to *ImportDir*`/auth.tar` the contents of `/tmp/.temp_desktop/importData/auth/`
- adds to *ImportDir*`/public_html.tar` the contents of `/tmp/.temp_desktop/importData/public_html/`
- removes the directory `/tmp/.temp_desktop/`
- automatically extracts the files and directories added by `migrate_files.txt` as long as they are fully qualified pathnames.

Import Tool

The import tool:

- extracts *ImportDir*`/desktop.tar` into `/etc/opt/SUNWps/desktop/`
- extracts *ImportDir*`/auth.tar` into *BaseDir*`/SUNWam/web-apps/services/WEB-INF/config/auth/default/`
- extracts *ImportDir*`/locale.tar` into *BaseDir*`/SUNWps/web-src/WEB-INF/classes`
- extracts *ImportDir*`/public_html.tar` into *BaseDir*`/SUNWam/public_html`

Sun ONE Compass Server Migration Tools

This section provides the tool design for the export and import tools used to migrate Sun™ ONE Compass Server 3.01C, Service Pack 1 or later to Sun ONE Portal Server 6.2.

Sun ONE Compass Server Export Tool Design

Synopsis

`perl./export.pl [-o` *ExportDir*`] [-i` *Target_InstanceDir*`]`

Description

The Sun ONE Compass Server export tool exports data from Sun ONE Compass Server 3.01C, Service Pack 1 and later to the Sun ONE Portal Server 6.2 system.

Parameters

Table A-4 shows the parameters and their descriptions for the `export.pl` script. This two-column table lists the parameters in the first column and their descriptions in the second column.

Table A-4 Parameters and descriptions for the `export.pl` script

Parameter	Description
`-o` *Exportdir*	Specifies the directory where the exported data is to be saved.
`-i` *Target_InstanceDir*	Specifies the directory for the target instance to be exported.

Usage

If you do not specify any parameters, the Sun ONE Compass Server export tool will export all instances to the current directory.

Sun ONE Compass Server Import Tool Design

Synopsis

`perl./import.pl [-o` *Target_InstanceDir*`] [-i` *Source_InstanceDir*`] [-p` *Portal_InstallDir*`]`

Description

The Sun ONE Compass Server import tool imports the Sun ONE Compass Server 3.01C, Service Pack 1 or later data exported with the `export.pl` script into Sun ONE Portal Server 6.2.

Parameters

Table A-5 shows the parameters and their descriptions for the `export.pl` script. This two-column table lists the parameters in the first column and their descriptions in the second column.

Table A-5 Parameters and descriptions for the `import.pl` script

Parameter	Description
`-o` *Target_InstanceDir*	Specifies the directory of the Sun ONE Portal Server 6.2 Search root.
`-i` *Source_InstanceDir*	Specifies the directory of the Sun ONE Compass Server 3.01C, Service Pack 1 or later instance.
`-p` *Portal_InstallDir*	Specifies the directory of the Sun ONE Portal Server 6.2 installation.

Appendix B

Sun ONE Portal Server 3.0 Data Migration Module Author's Guide

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

The data migration tools are designed to be expandable by using modules. The module can define a menu option, order items within the menu, and specify when it should run relative to other modules. The migration tools themselves are designed as a series of modules which makes them excellent examples of how to write modules.

This appendix contains the following sections:

- What Makes a Module?
- Module Example

What Makes a Module?

The syntax for module file names is:

`[0-9][0-9][`*descriptive_text*`][export,convert, or import]`

Migration modules must meet certain criteria. In particular, they must:

- Exist under the installed migration directory `/modules`.

  The migration tools are located in *BaseDir*`/SUNWps/migration`. The core migration tools search for modules in *BaseDir*`/SUNWps/migration/modules` in order to build the module list used by the migration executables. Each of the core module's directories (`cert`, `desktop`, `ldap`, `rules`) contain all the files needed to support the module as well as the module executables. For better organization, you should create your own directory under the modules directory that contains all files needed to run your module, even though files located directly in the modules directory will be added.

- Have the digits 0-9 as the first two characters of the file name.

  The first two digits of the module's file name tell the core tools where in the menu this module belongs as well as the execution order of the module. A higher number will run later in the process. A lower number will run earlier. LDAP is priority 40, Desktop is 50, and certificate is 60. So the menu appears as:

  ```
  1) LDAP Database

  2) Desktop

  3) Certificate Databases

  4) All of the above

  5) Exit
  ```

  If, for example, you place a new module file `30customexport` in *BaseDir*`/SUNWps/migration/modules/custom`, the export menu would appear as:

  `1)` *Custom menu text*

  ```
  2) LDAP Database

  3) Desktop

  4) Certificate Databases

  5) All of the above

  6) Exit
  ```

  If the user selected `All of the above`, the custom module would be run first, LDAP second, Desktop third, and certificate last.

- Have a file name ending with `export`, `convert`, or `import`.

  The last text of the filename must be one of these, otherwise the core tools will not recognize them. There can be other text between the priority number (described in the next bullet) and the suffix to give more meaning to the name. For example, `40ldapexport` is the name of the LDAP export module.

- Be written in `ksh`.

  The core tools, written in `ksh`, call the module files using the "`.`" function. This requires that the module be written in `ksh`. Other executables, however, can be run within the module file. The module has access to all variables defined within the core tools, including `JAVA_HOME` and `PERL` which allows Java and perl code to be run within the module. There are many examples of this within the LDAP and Desktop modules.

- Be executable.

  The module file must be executable in order to be recognized as a module. Note that module files that return a non-zero exit code will cause the core tool to stop and print an error message. Module files that return zero will assume to have completed successfully.

- Return a single line of text or nothing when invoked with the `--menu` option.

  The text returned represents the option that will be placed in the menu. If no text is returned, the module is considered "always-run" and will be executed regardless of the option the user selects. The menu option text must be localized by the module file prior to returning the value to the core tool.

Module Example

This section provides examples of the Desktop module located in *BaseDir*`/SUNWps/migration/modules/desktop`.

Desktop Module Files

Table B-1 shows the files comprising the Desktop module and their definitions. This two-column table shows the module name in the first column and its description in the second column.

Table B-1 Desktop Module File Names and Their Descriptions

File Name	Description
`20dtxpconvert`	This is an “always-run” module that extracts the Desktop templates to a temporary directory. The LDAP module uses some of these templates as well, so it runs for every conversion.
`50dtconvert`	Contains the functionality for converting Desktop templates.
`50dtexport`	Contains the functionality for exporting Desktop templates from Sun ONE Portal Server 3.0.
`50dtimport`	Contains the functionality for importing Desktop templates to Sun ONE Portal Server 6.2.
`auth.cksum`	File used to compare authentication templates.
`convertJsp.pl`	Converts a single JSP file.
`convertTemplate.pl`	Converts a single template file.
`migrate_files.txt`	Contains additional files to be migrated.

Module Code Examples

This section provides code extracts from two Desktop Modules.

- First four lines from the Desktop conversion module:

```
if [ “$1” == “--menu” ]; then

print “`$GETTEXT ‘Desktop’`”

exit 0

fi
```

  In the second line in the code for the Desktop conversion module, `$GETTEXT` is used to place the item `Desktop` in the conversion menu.

- First four lines from the “always-run” Desktop module:

```
if [ “$1” == “--menu” ]; then

print ““

exit 0

fi
```

Note that the Desktop menu option localized the menu option prior to printing it.

Important Variables

Table B-2 shows variables that are considered important for use by modules. This two-column table shows the variable in the first column and its description in the second column.

Table B-2 Important Variables Used by Modules

Variable	Definition
$exportDir	Location of export data.
$importDir	Location of import data.
$tempDesktop/exportData	Location of extracted templates during conversion.
$isFlatRole	Describes role migration type used only during conversion. If *$isFlatRole*=`0` (default), then the conversion tool performs role to suborganization migration. If *$isFlatRole*=`1`, then the conversion tool peforms role to role migration.
$isMerge	Describes the display profile merge type used only during import. If *$isMerge*=`0` (default), then the import tool overwrites all existing display profile documents when they conflict with imported display profile documents. If *$isMerge*=`1`, then the import tool merges all existing display profile documents when they conflict with imported display profile documents.

Table B-2 Important Variables Used by Modules *(Continued)*

Variable	Definition
$keepUsers	Describes user LDIF record add and delete type used only during import. If *$keepUsers*=0 (default), then overwrite all existing users by deleting any existing users that conflict with imported users and add new users. If *$keepUsers*=1, then keep existing users that conflict with imported users.
$reportFile	FQP to report file. You should only append to this file.
$errorFile	FQP to error file. You should only append to this file.

All variables and functions in each of the core tools are available to the module files.

The following is an example of writing to *$reportFile* and *$errorFile*:

```
if [ ! -d $tempDesktop/exportData ]; then

  print "`$GETTEXT 'Error - Cannot find exported template data in'`
$tempDesktop/exportData" | tee -a $errorFile $reportFile
```

Appendix C

Public API Changes Between Sun ONE Portal Server 3.0 and Sun ONE Portal Server 6.2

NOTE	All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products. Also, all instances of the Sun™ ONE Identity Server 5.1 product refer to what was formerly known as the iPlanet™ Directory Server Access Management Edition 5.1 product.

This appendix provides information on public API changes between Sun ONE Portal Server 3.0 and Sun™ ONE Portal Server 6.2 that you need to know for migration purposes. The tables in this appendix are not meant to be a replacement for the javadocs.

Table C-1 shows the API changes you need to know to migrate Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken. This three-column table shows the class and method signature in Sun ONE Portal Server 3.0 in the first column. The second column shows the application migration class method signature in Sun ONE Portal Server 6.2. The third column shows the provider migration class method signature in Sun ONE Portal Server 6.2.

Table C-1 Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken

Class/Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`com.iplanet.portalserverauth.server.Login`	`com.sun.identity.authentication.spi.AMLoginModule`	`com.sun.portal.providers.context.ProviderContext`
`Properties getAllTokenNames()`	`No equivalent available in JAAS based authentication framework.`	
`String [] getAllTokens()`	`No equivalent available in JAAS based authentication framework.`	
`String [] getAllTokensForState (int stateNumber)`	`No equivalent available in JAAS based authentication framework.`	

`Profile getApplicationProfile (String name)`	`AMStoreConnection.getSchema`
	and
	`AMSchema.getAttributeDefaults`	

`int getAuthLevel()`	`int getAuthLevel()`	
`int getCurrentState()`	`int getCurrentState()`	
`Profile getDomainProfile (String name)`	`AMStoreConnection.getOrganization()`	
`HttpServletRequest getHttpServletRequest()`	`HttpServletRequest getHttpServletRequest()`	
`HttpServletResponse getHttpServletResponse()`	`HttpServletResponse getHttpServletResponse()`	
`String getLocale()`	`String getLocale()`	`Locale getLocale()`
`int getNumberofStates()`	`int getNumberOfStates()`	
`int getNumberofTokens()`	`No equivalent available in JAAS based authentication framework.`	
`int getNumberofTokensForState()`	`No equivalent available in JAAS based authentication framework.`	
`boolean getPersistentCookieOn()`	`boolean getPersistentCookieOn()`	
`String getRequestDomain`	`String getRequestOrg()`	
`Session getSession()`	`SSOToken getSSOSession()`	
`String getSessionId()`	`String getSessionId()`	`String getSessionID()`
`String getToken()`	`No equivalent available in JAAS based authentication framework.`	

Table C-1 Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken *(Continued)*

Class/Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`String getToken (String name)`	`No equivalent available in JAAS based authentication framework.`	
`Profile getUserProfile (String name)`	`AMUser AMStoreConnecion getUser(String userDN) throws Authentication.Exception`	
`String getUserSessionProperty (String name)`	`String getUserSessionProperty (String name)throws javax.security.auth.login.Login. Expception`	`Object getSessionProperty (String name)`
`abstract String getUserTokenID()`	`javax.security.Principal getPrincipal()`	
`abstract void init()`	`abstract void init(javax.security.auth.Subject subject,java.util.Map sharedState, java.util.Map options)`	
`void logout()`	`void logout()`	
`int setAuthLevel (int auth_level)`	`boolean setAuthLevel (int auth_level`	
`void setDefaultURL (String url)`	`void setLoginSuccessURL(String url)`	
	`void setLoginFailureURL(String url)`	
`void setDomain (String domain)`	`Not supported.`	
`void setNextScreen(int num)`	`No equivalent available in JAAS based authentication framework.`	
`void setPersistantCookieOn()`	`boolesn setPersistantCookieOn()`	

Table C-1 Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken *(Continued)*

Class/Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2

`void setReplaceText (int screenNumber, int token, String text)`	`void replaceCallback(int state, int index, javax.security.callback. Callback)`
	To replace header text, the REPLACE text can be specified in the module properties XML file. Then the module can all
	`void replaceHeader(int state, String header) which throws javax.security.auth.login.LoginException`	

`void setReplaceText (int token, String text)`	`void replaceCallback(int state, int index, javax.security.callback. Callback)`
	To replace header text, the REPLACE text can be specified in the module properties XML file. Then the module can all
	`void replaceHeader (int state, String header) which throws javax.security.auth.login. LoginException`	

`void setReplaceText(int token, String[] text)`	`void replaceCallback(int state, int index, javax.security.callback. Callback)`
	To replace header text, the REPLACE text can be specified in the module properties XML file. Then the module can all
	`void replaceHeader(int state, String header) which throws javax.security.auth.login. LoginException`	

Table C-1 Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken *(Continued)*

Class/Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`void setUserSessionProperty (String name, String value)`	`void setUserSessionProperty(String name, String value) throws javax.security.auth.login.LoginException`	`void setSessionProperty(String name, Object val)`
`abstract void validate()`	`int process(javax.security.auth.callback.Callback[] callback, int state)`	
`com.iplanet.portalserver.auth.server.LoginException`	`javax.security.auth.login.LoginExceptio`	
`com.iplanet.portalserver.session.SessionListener`	`com.iplanet.sso.SSOTokenListener`	
`void sessionChanged (SessionEvent evt)`	`void ssoTokenChanged (SSOTokenEvent evt)`	
`com.iplanet.portalserver.session.Session`	`com.iplanet.sso.SSOToken`	
`void addSessionListener (SessionListener listener)`	`void addSSOTokenListener(SSOTokenListener listener)`	
`void addSessionListenerOnAllSession(SessionListener listener)`	`void addSSOTokenListener(SSOTokenListener listener)`	
`void addSessionListenerOnAllSession(String server, SessionListener listener)`	`void addSSOTokenListener(SSOTokenListener listener)`	
`String encodeURL(HttpServletResponse res, String url)`	`String encodeURL(String url)`	`String encodeURL(String url)`
`String encodeURL(String url)`	`String encodeURL(String url)`	
`Profile getAppProfile (String name)`	use the `SSOToken` to create an `AMStoreConnection` and then call `AMStoreConnection.getSchema`	

Table C-1 Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken *(Continued)*

Class/Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`String getClientDomain ()`	use the `SSOToken` to create an `AMStoreConnection` and then call `AMStoreConnection.getOrganization`	
`String getClientID()`	use SSOToken.getPrincipal().getName()	ProviderContext.getUserID method
`boolean getCookieSupport()`	`boolean getPersistentCookieOn()`	
`int getEventType()`	Any programs that use this API in Sun ONE Portal Server 3.0 have a race condition because there is no way to ensure that events are not missed. Replace use of this method with an `SSOTokenListener` object.	
`SessionID getID()`	`java.lang.String getSessionId()`	
`long getIdleTime()`	`long getIdleTime()`	
`long getMaxCachingTime()`	`long getMaxCachingTime()`	
`long getMaxIdleTime()`	`long getMaxIdleTime()`	
`long getMaxSessionTime()`	`long getMaxSessionTime()`	
`String getProperty(String name)`	`String getProperty(String name)`	
`static Session getSession(SessionID sid)`	`SSOToken getSSOSession()`	
`int getState(boolean reset)`	`int getCurrentState()`	
`long getTimeLeft()`	`long getTimeLeft()`	
`int getType()`	All publicly accessible sessions in Sun ONE Portal Server 3.0 have a session type of `Session.USER_SESSION`. So any application code that uses this method can assume that the method only returns `Session.USER_SESSION`.	
`Profile getUserProfile()`	`AMUser AMStoreConnection.getUser(String userDN)` throws `AuthenticationException`	

Table C-1 Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken *(Continued)*

Class/Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`Vector getValidSessions()`	Sun ONE Identity Server does not provide functionality for session (or SSO token) management. This is a planned feature for a future release.	
`Vector getValidSessions(String server)`	Sun ONE Identity Server does not provide functionality for session (or SSO token) management. This is a planned feature for a future release.	
`void logout()`	`void logout()`	
`void setDebug(Debug debug, String profileName)`	This method does the following: `Profile p = getAppProfile(profileName)`; `debug.setDebug(p.getAttributeString(profileName + "-debug"));` See the other entries in the table for the equivalents of `getAppProfile`, `getAttributeString`, and `setDebug`.	
`void setProperty(String name, String value)`	`void setProperty(String name, String value)`	
`com.iplanet.portalserver.session.SessionEvent`	`com.iplanet.sso.SSOTokenManager`	
`void destroySession(Session session)`	`void destroyToken(SSOToken token)`	
`void refresh(boolean reset)`	`void refreshSession(SSOToken token)`	
`com.iplanet.portalserver.session.SessionEvent`	`com.iplanet.sso.SSOTokenEvent`	
`Session getSession()`	`SSOToken getToken()`	
`long getTime()`	`long getTime()`	
`int getType()`	`int getType()`	

Table C-1 Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken *(Continued)*

Class/Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`com.iplanet.portalserver.session.SessionID`	`com.iplanet.sso.SSOTokenID`	
`boolean equals(Object object)`	`boolean equals(Object object)`	
`boolean getCookieMode()`	use `com.iplanet.authentication.spi` and then `boolean getPersistentCookieOn()`	
`String getRandomString()`	The structure of the SSO token cookie value has changed in iSun ONE Identity Server, hiding the internals of its structure from the application. If this value is needed, the `SSOToken.setProperty` method can be used to save this information and retrieve it later.	
`String getSessionDomain()`	The structure of the SSO token cookie value has changed in Sun ONE Identity Server, hiding the internals of its structure from the application. If this value is needed, the `SSOToken.setProperty` method can be used to save this information and retrieve it later.	
`String getSessionServer()`	The structure of the SSO token cookie value has changed in Sun ONE Identity Server, hiding the internals of its structure from the application. If this value is needed, the `SSOToken.setProperty` method can be used to save this information and retrieve it later.	
`String getSessionServerPort()`	The structure of the SSO token cookie value has changed in Sun ONE Identity Server, hiding the internals of its structure from the application. If this value is needed, the `SSOToken.setProperty` method can be used to save this information and retrieve it later.	

Table C-1 Migrating Sun ONE Portal Server 3.0 Session to Sun ONE Identity Server SSOToken *(Continued)*

Class/Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`String getSessionServerProtocol()`	The structure of the SSO token cookie value has changed in Sun ONE Identity Server, hiding the internals of its structure from the application. If this value is needed, the `SSOToken.setProperty` method can be used to save this information and retrieve it later.	
`int hashCode()`	`int hashCode()`	
`String toString()`	`String toString()`	
`com.iplanet.portalserver.session.SessionNotificationHandler`	`No similiar class`	
`com.iplanet.portalserver.session.SessionException`	`com.iplanet.sso.SSOException`	

You can find samples in *BaseDir*/`SUNWam/samples/authentication` and *BaseDir*/`SUNWam/samples/sso`.

Table C-2 shows the API changes you need to know to migrate the Sun ONE Portal Server 3.0 profile to Sun ONE Identity Server user management classes and service management classes. This three-column table shows the method signature in Sun ONE Portal Server 3.0 in the first column. The second column shows the application migration class method signature in Sun ONE Portal Server 6.2. The third column shows the provider migration class method signature in Sun ONE Portal Server 6.2.

Table C-2 Migrating the Sun ONE Portal Server 3.0 Profile to Sun ONE Identity Server User Management Classes and Service Management Classes

Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`com.iplanet.portalserver.profile.ProfileListener`	`com.iplanet.am.sdk.AMEventListener`	

Table C-2 Migrating the Sun ONE Portal Server 3.0 Profile to Sun ONE Identity Server User Management Classes and Service Management Classes *(Continued)*

Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
void profileChanged(ProfileEvent notif)	void objectChanged(AMEvent event)	
com.iplanet.portalserver.profile.Profile	com.iplanet.am.sdk.AMObject	com.sun.portal.providers.context.ProviderContext
java.lang.String getAttributeString(java.lang.String name) throws ProfileException	java.lang.String getStringAttribute(java.lang.String attributeName) throws AMException, SSOException	String getStringAttribute(String name)
java.util.Enumeration getAttribute(java.lang.String name) throws ProfileException	java.util.Set getAttribute(java.lang.String attributeName) throws AMException, SSOException	Map getCollectionProperty (String channel, String name) throws ProviderContextException or Object getProperty(String channel, String name) throws ProviderContextException
java.util.Hashtable getAttributes() throws ProfileException	java.util.Map getAttributes() throws AMException, SSOException	
java.util.Hashtable getAttributes(java.lang.String wildchar) throws ProfileException	java.util.Map getAttributes(java.util.Set attributeNames) throws AMException, SSOException	
void setAttributeString(java.lang.String name, java.lang.String value, int operation) throws ProfileException	void setStringAttribute(java.lang.String attributeName, java.lang.String value) throws AMException, SSOException	setProperty(String channel, String key, Object val) throws ProviderContextException or void setCollectionProperty(String channel, String key, Map val) throws ProviderContextException
void setAttribute(java.lang.String name, java.util.Enumeration value, int operation) throws ProfileException	void setAttributes(java.util.Map attributes) throws AMException, SSOException	

Table C-2 Migrating the Sun ONE Portal Server 3.0 Profile to Sun ONE Identity Server User Management Classes and Service Management Classes *(Continued)*

Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`void delAttribute(java.lang.String name)` throws `ProfileException`	`void removeAttributes(java.util.Set attributes)` throws `AMException, SSOException`	
`void delAttributes(java.lang.String wildchar)` throws `ProfileException`	`void removeAttributes(java.util.Set attributes)` throws `AMException, SSOException`	
`void addProfileListener (ProfileListener l)`	`void addEventListener(AMEventListener listener)` throws `SSOException`	
`void removeProfileListener(ProfileListener l)`	`public void removeEventListener(AMEventListener listener)`	
`void store(boolean force)` throws `ProfileException`	`void store()`throws `AMException, SSOException`	
`boolean isAllowed(java.lang.String privilege)` throws `ProfileException`	Call `getAttribute` on a boolean attribute.	
`boolean isAllowed(java.lang.String privilege, java.lang.String value, int match)` throws `ProfileException`	Call `getAttribute` on one or more attributes and perform the appropriate matching against the attribute value. The policy interface being made public is planned for a future release.	
`boolean isAllowedIgnoreCase(java.lang.String privilege, java.lang.String value, int match)` throws `ProfileException`	Call `getAttribute` on one or more attributes and perform the appropriate matching against the attribute value. The policy interface being made public is planned for a future release.	
`void loadPrivileges(java.util.Set privilegeNames)` throws `ProfileException`	`AMTemplate createPolicyTemplate (java.lang.String serviceName,java.util.Map attributes)` throws `java.lang.UnsupportedOperationException, AMException, SSOException`	

Table C-2 Migrating the Sun ONE Portal Server 3.0 Profile to Sun ONE Identity Server User Management Classes and Service Management Classes *(Continued)*

Method Signature in Sun ONE Portal Server 3.0	Application Migration Class Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`java.util.Enumeration getPrivilegeNames()` throws `ProfileException`	`java.util.Map getPolicy(java.lang.String serviceName)` throws `java.lang.UnsupportedOperationException, AMException, SSOException`	
`java.util.Enumeration getPrivilegeNames(java.lang.String wildchar)` throws `ProfileException`	`java.util.Map getPolicy(java.lang.String serviceName)` throws `java.lang.UnsupportedOperationException, AMException, SSOException`	
`java.lang.String getProfileName()`	`java.lang.String getDN()`	
`int getProfileType()`	Use `instanceof` to determine whether the object being operated on is `AMUser`, `AMRole`, `AMOrganization`, or `AMSchema`.	
`com.iplanet.portalserver.profile.ProfileEvent`	`com.iplanet.am.sdk.AMEvent`	
`Profile getProfile()`	`java.lang.String getSourceDN()`	
`int getType()`	`int getEventType()`	
`long getTime()`	No equivalent. If it is necessary for the application to communicate the times for profile modification events, this can be done using an out-of-band mechanism.	
`com.iplanet.portalserver.profile.ProfileException`	`com.iplanet.am.sdk.AMException`	`com.sun.portal.providers.context.ProviderContextException`
`int getType()`	`java.lang.String getErrorCode()`	

You can find samples in *BaseDir*`/SUNWam/samples/sdk`.

Table C-3 shows the API changes you need to know to migrate Sun ONE Portal Server 3.0 logging to Sun ONE Identity Server logging APIs. This three-column table shows the method signature in Sun ONE Portal Server 3.0 in the first column. The second column shows the method signature in Sun ONE Portal Server 6.2. The third column shows the provider migration class method signature in Sun ONE Portal Server 6.2.

Table C-3 Migrating Sun ONE Portal Server 3.0 Logging to Sun ONE Identity Server Logging APIs

Method Signature in Sun ONE Portal Server 3.0	Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`com.iplanet.portalserver.logging.LogManager`	`com.iplanet.log.LogManager`	
`LogManager(Session sess)`	`LogManager(SSOToken token)`	
`LogManager(SessionID sid)`	Use the `SSOToken` version of the constructor.	
`LogManager(SessionID sid, java.lang.String server)`	Use the `SSOToken` version of the constructor.	
`LogManager(Session sess, java.lang.String server)`	Use the `SSOToken` version of the constructor.	
`void create(java.lang.String logName)`	`void create(java.lang.String logName)`	
`void delete(java.lang.String logName)`	Deleting a log file via the API is no longer supported.	
`java.util.Hashtable getLogAttrs(java.lang.String logName)`	The log APIs for reading logs and setting or getting log attributes do not exist in Sun ONE Identity Server. In the future, the plan is for Sun ONE products to migrate towards the use of the standard Java 1.4 logging APIs. Use of these proprietary log APIs is discouraged.	
`java.util.Vector  list()`	The log APIs for reading logs and setting or getting log attributes do not exist in Sun ONE Identity Server. In the future, the plan is for Sun ONE products to migrate towards the use of the standard Java 1.4 logging APIs. Use of these proprietary log APIs is discouraged.	

Table C-3 Migrating Sun ONE Portal Server 3.0 Logging to Sun ONE Identity Server Logging APIs

Method Signature in Sun ONE Portal Server 3.0	Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`java.util.Vector list(java.lang.String logName)`	The log APIs for reading logs and setting or getting log attributes do not exist in Sun ONE Identity Server. In the future, the plan is for Sun ONE products to migrate towards the use of the standard Java 1.4 logging APIs. Use of these proprietary log APIs is discouraged.	
`java.util.Vector read(java.lang.String logName)`	The log APIs for reading logs and setting or getting log attributes do not exist in Sun ONE Identity Server. In the future, the plan is for Sun ONE products to migrate towards the use of the standard Java 1.4 logging APIs. Use of these proprietary log APIs is discouraged.	
`java.util.Vector read(java.lang.String logName, java.lang.String query)`	The log APIs for reading logs and setting or getting log attributes do not exist in Sun ONE Identity Server 5.1. In the future, the plan is for Sun ONE products to migrate towards the use of the standard Java 1.4 logging APIs. Use of these proprietary log APIs is discouraged.	
`java.util.List readLogRecord(java.lang. String logName)`	The log APIs for reading logs and setting or getting log attributes do not exist in Sun ONE Identity Server. In the future, the plan is for Sun ONE products to migrate towards the use of the standard Java 1.4 logging APIs. Use of these proprietary log APIs is discouraged.	
`java.util.List readLogRecord(java.lang. String logName, java.lang.String query)`	The log APIs for reading logs and setting or getting log attributes do not exist in Sun ONE Identity Server. In the future, the plan is for Sun ONE products to migrate towards the use of the standard Java 1.4 logging APIs. Use of these proprietary log APIs is discouraged.	

Table C-3 Migrating Sun ONE Portal Server 3.0 Logging to Sun ONE Identity Server Logging APIs

Method Signature in Sun ONE Portal Server 3.0	Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`void setLogAttrs(java.lang.String logName, java.util.Hashtable logattrs)`	The log APIs for reading logs and setting or getting log attributes do not exist in Sun ONE Identity Server1. In the future, the plan is for Sun ONE products to migrate towards the use of the standard Java 1.4 logging APIs. Use of these proprietary log APIs is discouraged.	
`void write(LogRecord lr, java.lang.String logName)`	`void write(LogRecord lr, java.lang.String logName)`	
`com.iplanet.portalserver.logging.LogRecord`	`com.iplanet.log.LogRecord`	
`LogRecord(java.lang.String type, java.lang.String msg)`	`LogRecord(java.lang.String type, java.lang.String msg)`	
`java.lang.String getRecMsg()`	`java.lang.String getRecMsg()`	
`java.lang.String getRecType()`	`java.lang.String getRecType()`	
`com.iplanet.portalserver.logging.LogException`	`com.iplanet.log.LogException`	
`LogException()`	`LogException()`	
`LogException(java.lang.String s)`	`LogException(java.lang.String msg)`	
`LogException(java.lang.String s, int type)`	`LogException(java.lang.String msg, int type)`	
`int getType()`	`int getType()`	

Table C-4 shows the API changes you need to know to migrate Sun ONE Portal Server 3.0 debug to Sun ONE Identity Server debug APIs. This three-column table shows the method signature in Sun ONE Portal Server 3.0 in the first column. The second column shows the method signature in Sun ONE Portal Server 6.2. The third column shows the provider migration class method signature in Sun ONE Portal Server 6.2.

Table C-4 Migrating Sun ONE Portal Server 3.0 Debug toSun ONE Identity Server Debug APIs

Method Signature in Sun ONE Portal Server 3.0	Method Signature in Sun ONE Portal Server 6.2	Provider Migration Class Method Signature in Sun ONE Portal Server 6.2
`com.iplanet.portalserver.util.Debug`	`com.iplanet.am.util.Debug`	
`Debug(java.lang.String debugName)`	Use `Debug.getInstance(String)`	
`boolean debugEnabled()`	Use `Debug.messageEnabled()`	
`void destroy()`	`void destroy()`	
`void error(java.lang.String msg)`	`void error(java.lang.String msg)`	
`void error(java.lang.String msg, java.lang.Throwable t)`	`void error(java.lang.String msg, java.lang.Throwable t)`	
`static Debug getInstance(java.lang. String debugName)`	`static Debug getInstance(java.lang. String debugName)`	
`int getState()`	`int getState()`	
`void message(java.lang.String msg)`	`void message(java.lang.String msg)`	
`void message(java.lang.String msg, java.lang.Throwable t)`	`void message(java.lang.String msg, java.lang.Throwable t)`	
`boolean messageEnabled()`	`boolean messageEnabled()`	
`void setDebug()`	This is done by default by the `Debug.getInstance` call if `setDebug(int)` is not called. To return the Debug object to the debug level set in the system configuration file, use `Debug.setDebug(SystemProperties.get(“com.iplanet.services.debug.level”))`	
`void setDebug(int debugType)`	`void setDebug(int debugType)`	
`void setDebug(java.lang.String debugType)`	`void setDebug(java.lang.String debugType)`	
`void warning(java.lang.String msg)`	`void warning(java.lang.String msg)`	
`void warning(java.lang.String msg, java.lang.Throwable t)`	`void warning(java.lang.String msg, java.lang.Throwable t)`	
`boolean warningEnabled()`	`boolean warningEnabled()`	

Table C-5 shows the API changes you need to know to migrate the Desktop SDK from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2. This three-column table shows the Sun ONE Portal Server 3.0 class and method in the first column. The second column shows the Sun ONE Portal Server 6.2 class and method. The third column shows comments, if appropriate.

Table C-5 Migrating the Desktop SDK from Sun ONE Portal Server 3.0 to Sun ONE Portal Server 6.2

Class/Method signature in Sun ONE Portal Server 3.0	Class/Method signature in Sun ONE Portal Server 6.2	Comments
`Provider, ProviderAdapter, ProfileProviderAdapter`		
`init(Session)`	`init(HttpServletRequest)`	remove dependency on specific service APIs, separation of service layer
`getBackgroundColor()`		type 2 property
`getColumn()`		type 2 property
`getEdit(Map)`	`getEdit(HttpServletRequest, HttpServletResponse)`	Map argument version is deprecated. Use the `HttpServletRequest` version.
`getContent(Map)`	`getContent(HttpServletRequest, HttpServletResponse)`	Map argument version is deprecated. Use `HttpServletRequest` version.
`getRow()`		type 2 property
`hasBorder()`		type 2 property
`hasFrame()`		type 2 property
`hasHelp()`	`getHelp()`	if `getHelp()` returns non-null, then the channel has help.
`isDetachable()`		type 2 property
`isDetached()`		type 2 property
`isMinimizable()`		type 2 property
`isMinimized()`		type 2 property
`isMovable()`		type 2 property
`isRemovable()`		type 2 property
`load()`		Separation of service layer. Use `ProviderContext` (accessible through `ProviderAdapter` implementation) to access properties. `Property load` is implicit in `ProviderContext`.

Table C-5 Migrating the Desktop SDK from Sun ONE Portal Server 3.0 to Sun ONE Portal Server

Class/Method signature in Sun ONE Portal Server 3.0	Class/Method signature in Sun ONE Portal Server 6.2	Comments
`processEdit(map)`	`processEdit(HttpServletRequest, HttpServletResponse)`	Map argument version is deprecated. Use the `HttpServletrequest` version.
`setColumn()`		type 2 property
`setDetached()`		type 2 property
`setMinimized()`		type 2 property
`setRow()`		type 2 property
`store()`		Separation of service layer. Use `ProviderContext` (accessible through `ProviderAdapter` implementation) to access properties. `Property load` is implicit in `ProviderContext`.
`EDIT_COMPLETE`	`ProviderEditTypes.EDIT_COMPLETE`	
`EDIT_SUBSET`	`ProviderEditType.EDIT_SUBSET`	
`WIDTH_THIN`	`ProviderWidths.WIDTH_THIN`	
`WIDTH_THICK`	`ProviderWidths.WIDTH_THICK`	
`WIDTH_FULL_TOP`	`ProviderWidths.WIDTH_FULL_TOP`	
`WIDTH_FULL_BOTTOM`	`ProviderWidths.WIDTH_FULL_BOTTOM`	
`ProviderAdapter`		
`getSession()`	`ProviderContext` “session” services	Separation of services. Explicit access to a “session” object not required, functionality is provided by `ProviderContext` (accessible through `ProviderAdapter` implementation.
`ProfileProviderAdapter`		
`getResourceBundle()`	`ProviderAdapter.getResourceBundle()`	Implementation moved up in the class hierarchy.
`getListProperty(List, Vector)`	`getListProperty(List, List)`	Default value is supplied as `List`.
`isAllowed(String)`	`isAllowed(String)`	Semantic difference. In Sun ONE Portal Server 6.2, there are technically no privileges. Instead, this fetches a boolean property of the given name.

Table C-5 Migrating the Desktop SDK from Sun ONE Portal Server 3.0 to Sun ONE Portal Server

Class/Method signature in Sun ONE Portal Server 3.0	Class/Method signature in Sun ONE Portal Server 6.2	Comments
`isAllowed(String, String)`	`ProviderContext.getBooleanProperty(String, String)`	The Sun ONE Portal Server 3.0 method fetched a privilege from another component (for example, a channel). Using `ProviderContext` directly in Sun ONE Portal Server 6.2 allows for this functionality.
`isAllowed(String, String, int)`		Sun ONE Portal Server 3.0 allowed pattern matching to check privileges. This feature is not available in Sun ONE Portal Server 6.2.
`isAllowed(String, String, String, int)`		Sun ONE Portal Server 3.0 allowed pattern matching to check privileges. This feature is not available in Sun ONE Portal Server 6.2.
`setListProperty(String, String, Vector)`	`ProviderContext.setCollectionProperty(String, String, List)`	`List` instead of `Vector`, setting of value for other component (channel) available through `ProviderContext`.
`setLocalListProperty()`	`ProviderContext.setSessionProperty()`, `ProviderContext.getSessionProperty()`	In Sun ONE Portal Server 3.0, a local property was set and stored for the channel but not persisted. Use `ProviderContext`'s session properties for the same functionality.
`setLocalStringProperty()`	`ProviderContext.setSessionProperty()`, `ProviderContext.getSessionProperty()`	In Sun ONE Portal Server 3.0, a local property was set and stored for the channel but not persisted. Use `ProviderContext`'s session properties for the same functionality.
`setStringProperty(String, String, String)`	`ProviderContext.setStringProperty(String, String)`	Setting of value for other component (channel) available through `ProviderContext`.

Table C-5 Migrating the Desktop SDK from Sun ONE Portal Server 3.0 to Sun ONE Portal Server

Class/Method signature in Sun ONE Portal Server 3.0	Class/Method signature in Sun ONE Portal Server 6.2	Comments
DesktopURL	ProviderContext.getDesktopURL(), DesktopServlet	In Sun ONE Portal Server 3.0, the DesktopURL class served two purposes. It was a placeholder for the javadocs explaining the parameters understood by the DesktopServlet. This information has been moved into the DesktopServlet javadocs. It also assisted in generating URLs for access to the Desktop application. This functionality is now available from the ProviderContext method.
LoadPropertiesException		Loading of properties is explicit in Sun ONE Portal Server 6.2.
ProfileProviderException		This exception was not used in Sun ONE Portal Server 3.0 and therefore needs no replacement.
ProviderWrapperException	ProviderException	Exception wrapping functionality is implemented in ProviderException.

Appendix D

Tag and String Changes Made in Templates and JSPs by the Migration Tools

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

This appendix contains two tables illustrating the changes the Sun ONE Portal Server 3.0 Data Migration Tool Suite performs for tags and strings within the template and JSP™ files.

Table D-1 shows template tag and string changes by the migration tools. This table contains two columns. The first column lists the template tag or string in Sun ONE Portal Server 3.0. The second column lists the template tag or string in Sun ONE Portal Server 6.2.

Table D-1 Template Tag and String Changes Made by the Migration Tools

Sun ONE Portal Server 3.0 Template Tag or String	Sun ONE Portal Server 6.2 Template Tag or String
User Tags:	User Tags:
`borderSize`	`borderWidth`
`iwtDesktop-fontFace1`	`fontFace1`
`iwtUserInfoProvider-firstName`	`givenname`
`iwtUserInfoProvider-lastName`	`sn`
`iwtUser-SMTPServerName`	`iplanet-ps-netmail-smtp-server-name`

Table D-1 Template Tag and String Changes Made by the Migration Tools *(Continued)*

Sun ONE Portal Server 3.0 Template Tag or String	Sun ONE Portal Server 6.2 Template Tag or String
`iwtUser-IMAPServerName`	`iplanet-ps-netmail-imap-server-name`
`iwtUser-IMAPUserId`	`iplanet-ps-netmail-imap-userid`
`iwtUser-IMAPPassword`	`iplanet-ps-netmail-imap-password`
User Strings:	
Providers	

If the source has:
`*input.*hidden.*value=”iwtOptionsProvider”`
	the migration tools will substitute `iwtOptionsProvider` with:
	`[tag:frontContainerName]`
	and add:
	`<INPUT TYPE=HIDDEN NAME=”[tag:frontContainerName].containerEdit” SIZE=-1 VALUE=”option”>`

If the source has:
`*input.*hidden.*value=”iwtContentProvider”`
	the migration tools will substitute `iwtContentProvider` with:
	`[tag:frontContainerName]`
	and add:
	`<INPUT TYPE=HIDDEN NAME=”[tag:frontContainerName].containerEdit” SIZE=-1 VALUE=”content”>`

If the source has:
`*input.*hidden.*value=”iwtLayoutProvider”`
	the migration tools will substitute `iwtLayoutProvider`
	with:
	`[tag:frontContainerName]`
	and add:
	`<INPUT TYPE=HIDDEN NAME=”[tag:frontContainerName].containerEdit” SIZE=-1 VALUE=”layout”>`

If the source has: `*input.*hidden.*value=”iwtEditProvider”`	the migration tools will substitute `iwtEditProvider` with: `[tag:providerName]`

Table D-1 Template Tag and String Changes Made by the Migration Tools *(Continued)*

Sun ONE Portal Server 3.0 Template Tag or String	Sun ONE Portal Server 6.2 Template Tag or String
If the Source has:	the migration tools will substitute
/DesktopServlet?action=content&provider=iwtFrontProvider	/DesktopServlet?action=content&provider=iwtFrontProvider with dt
Cancel button:	
*input.*button.*onclick=”openFrontPage();	/OnClick=”location=’[tag:desktop_url]’”
Form Action:	
*form.action=”DesktopServlet” sub: action=”DesktopServlet”	ACTION=”dt”
.*fullurl.split.*”/DesktopServlet” sub: “/DesktopServlet”	“dt”
action=attach.*provider=[tag:providername] sub: action=attach.*provider=[tag:providername]	action=process&amp;provider=[tag:frontContainerName]&amp; [tag:frontContainerName].channelAction=attach&amp;[tag:frontContainerName].targetProvider=[tag:providerName]
%hanaString sub: $_	$hanaString{$_}
Login Channel:	
file: /Login/display.html	file: /iwtLoginProvider/display.html/
a href.*module=Membership.*	
sub: module=Membership	with: module=Membership&arg=newsession/
else	
sub: .*input.*hidden.*userName.*req_uid.*	<INPUT TYPE=HIDDEN NAME=”arg” SIZE=-1 VALUE=”newsession”>
Content:	

Table D-1 Template Tag and String Changes Made by the Migration Tools *(Continued)*

Sun ONE Portal Server 3.0 Template Tag or String	Sun ONE Portal Server 6.2 Template Tag or String
userTemplate.html	userTemplate.html
[tag:content]	“<!-- BEGIN TOP FULL PROVIDERS -->”; “<TABLE WIDTH=100% BORDER=0 CELLPADDING=2 CELLSPACING=0>”; “<TR><TD VALIGN=TOP><CENTER>”; “[tag:fullTopContent]”; “</CENTER></TD></TR></TABLE>”; “<!-- END TOP FULL PROVIDERS -->”; “<TABLE WIDTH=100% BORDER=0 CELLPADDING=0 CELLSPACING=4><TR>”; “<TD WIDTH=”[tag:leftWidth]%” VALIGN=TOP>”; “<!-- BEGIN LEFT PROVIDERS -->”; “[tag:leftContent]”; “<!-- END LEFT PROVIDERS -->”; “</TD>; “<TD WIDTH=”[tag:centerWidth]%” VALIGN=TOP>”; “<!-- BEGIN CENTER PROVIDERS -->”; “[tag:centerContent]”; “<!-- END CENTER PROVIDERS -->”; “</TD>”; “<TD WIDTH=”[tag:rightWidth]%” VALIGN=TOP>”; “<!-- BEGIN RIGHT PROVIDERS -->”; “[tag:rightContent]”; “<!-- END RIGHT PROVIDERS -->”; “</TD></TR></TABLE>”; “<!-- BEGIN BOTTOM FULL PROVIDERS -->”; “<TABLE WIDTH=100% BORDER=0 CELLPADDING=2 CELLSPACING=0>”; “<TR><TD VALIGN=TOP><CENTER>”; “[tag:fullBottomContent]”; “</CENTER></TD></TR></TABLE>”; “<!-- END BOTTOM FULL PROVIDERS -->”;

Table D-1 Template Tag and String Changes Made by the Migration Tools *(Continued)*

Sun ONE Portal Server 3.0 Template Tag or String	Sun ONE Portal Server 6.2 Template Tag or String
`userTemplate.template - change bgcolor`	FFFFFF
`333366`	
`tabs.html`	`remove 100%`
`remove 100% width`	`$offset = index($line, $search);`
`width`	`$length = 12;`
	`$line = substr($line, 0, $offset-1) . substr ($line, $offset+$length)`
	`Place </center> tag at beginning of file.`
	`Place <center> tag at end of file.`
`tab.html/selectedTab.html`	`remove 100% width.`

Table D-2 shows the JSP changes made by the `ConvertJSP.pl` script. This table contains two columns. The first column lists the JSP methods or elements in Sun ONE Portal Server 3.0. The second column lists the JSP methods or elements in Sun ONE Portal Server 6.1.

Table D-2 JSP Changes Made by the `ConvertJSP.pl` script

Sun ONE Portal Server 3.0 JSP Methods or Elements	Sun ONE Portal Server 6.2 JSP Methods or Elements
`loadProperties()`	`n/a`
`com.iplanet.portalserver.providers.jsp.JSPProvider`	`com.sun.portal.providers.jsp.JSPProvider`
`getStringProperty.*iwtUserInfoProvider.*firstName.`	`getProviderContext().getStringAttribute(“givenname”)`
`getStringProperty.*iwtUserInfoProvider.*lastName.*`	`getProviderContext().getStringAttribute(”sn”)`
`response.sendRedirect(.*new DesktopURL(.*)`	`response.sendRedirect(jspProviderReference.getProviderContext().getDesktopURL(request))`
`iwtUser-IMAPServerName`	`iplanet-ps-netmail-imap-server-name`
`iwtUser-IMAPUserId`	`iplanet-ps-netmail-imap-userid`

Table D-2 JSP Changes Made by the `ConvertJSP.pl` script *(Continued)*

Sun ONE Portal Server 3.0 JSP Methods or Elements	Sun ONE Portal Server 6.2 JSP Methods or Elements
`iwtUser-IMAPPassword`	`iplanet-ps-netmail-imap-password`

`*)DesktopURL(.*)=.*new DesktopURL.*`
`example: DesktopURL durl = new DesktopURL(p.getSession()).toString())`
	`“String “.$2.” = (String)pageContext.getAttribute(”url”)`

`/images`	`[surl:/ps3.0Images/$1]`
`/href=”/docs/(.*?)”`	`[surl:/docs/$1]`

Appendix E

Extended Sun ONE Portal Server 3.0 Component Migration

NOTE	All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

This appendix describes the steps you need to take when migrating an extended Sun ONE Portal Server 3.0 component to a Sun™ ONE Portal Server 6.2 service.

This appendix contains the following sections:

- Options For Making Manual Changes After Migration
- Example of Adding Custom Sun ONE Identity Server Attributes for Sun ONE Portal Server 6.2

Options For Making Manual Changes After Migration

The Sun ONE Portal Server 3.0 Migration Tool Suite retains the mapping between `iwtUser` and the `amUser` service. If, for example, you have extended `iwtUser` and attributes have been added, the migration tools do not migrate those attributes. After migration, you need to make manual changes using one of the two following options:

Option 1

1. Corresponding to the `iwtUser` extensions that you have made, add entries to the *BaseDir*`/SUNWam/config/xml/<amUser.xml` file. These changes should be in accordance with the *BaseDir*`/SUNWam/dtd/sms.dtd` file. For more details, see the *Sun ONE Identity Server Programmer's Guide*.

2. Using the `amadmin` utility, in *BaseDir*`/SUNWam/bin`, delete the existing user Service. For example:

   ./amadmin -u "uid=amadmin,ou=People,o=*defaultorg***,o=***rootsuffix* | *dc=defaultorg***,**
   dc=*rootsuffix***" -w** *password* **--deleteService iPlanetAMUserService**

3. Load the new, modified user service. For example:

   ./amadmin -u "uid=amadmin,ou=People,o=*defaultorg***,o=***rootsuffix* | *dc=defaultorg***,**
   dc=*rootsuffix***" -w** *password* **--schema amUser.xml**

4. Write a script that converts user attributes corresponding to the attributes added in the previous steps to the equivalent attributes in the Sun ONE Portal Server 6.2 service. The script should generate an LDIF file.

5. Use `ldapmodify` in *BaseDir*`/SUNWam/bin` to load the generated LDIF file to the directory server. For example:

   ./ldapmodify -D "cn=Directory Manager" -w *password* **-f** *convertedfile*

This will ensure that all the extended attributes are now available in Sun ONE Portal Server 6.2.

Option 2

1. Corresponding to the `iwtUser` extensions that you have made, create a new `amUser1.xml` file. These changes should be in accordance with the *BaseDir*`/SUNWam/dtd/sms.dtd` file. For more details, see the*Sun ONE Identity Server Programmer's Guide*.

2. Load the new, modified user service. For example:

   ./amadmin -u "uid=amadmin,ou=People,o=*defaultorg***,o=***rootsuffix* | *dc=defaultorg***,**
   dc=*rootsuffix***" -w** *password* **--schema amUser1.xml**

3. Write a script that converts user attributes corresponding to the attributes added in the previous steps to the equivalent attributes in the Sun ONE Portal Server 6.2 service.

4. Use `ldapmodify` in *BaseDir*/`SUNWam/bin` to load the generated file to the directory server. For example:

   `./ldapmodify  -D “cn=Directory Manager” -w` *password* **`-f`** *convertedfile*

Code Example E-1 provides a sample script.

Code Example E-1 Sample Script to Convert Attributes

```
#!/usr/bin/perl

open(USERFILE,”/tmp/user.ldif”) || die “$!: Cannot open it”;

@attrsToAdd=(“test-attr-one-at”,”test-attr-to-add-at”);
#while(<USERFILE>){
 while (@block=getBlock(\*USERFILE)){
    $_=shift(@block);
    #print “\n”;
        foreach $attr(@attrsToAdd){
    $j=0;
    foreach $block_line(@block){
        $position = index($block_line,$attr);
        if($position >= 0){
            @splitAttr=split(/:/,$block_line);
            $reqAttr[$j]=$splitAttr[1];
            $j++;
        }
    }
    if($j > 0){
        print “\n”;
        print $_;
        print “\nchangetype:modify”;
        print “\nadd:”;
        print $attr;$k=0;
                while($k < $j){
            print “\n$attr:$reqAttr[$k]”;
            $k++;
        }
        print “\n”;
    }
    }
 }
#}

close USERFILE;

sub getBlock {
    my ($fhref) = @_;
    local ($curLine, $inBlock);
    $inBlock=0;
    while (!eof($fhref)){
    $_=getLine($fhref);
# Not sure if I want to ignore this one yet
#    if ($curLine =~ /0---iwt-Profile-Reversed---0/) {next;}
```

Code Example E-1 Sample Script to Convert Attributes *(Continued)*

```
# Check for new block
    if (/^dn: cn=(.+?),/) {
        $user=$1;
# Reset user attribute array
        @blockInfo=($_);
        $inBlock=1;
        next;
    }
# Check for end of Block
    if (/^$/){
        if ($inBlock ==1){
            return @blockInfo;
        } else {
            next;
        }
    }
# Regular data, push it on
    push @blockInfo,$_;
    }
    if ($inBlock==1){
    # Add last line
    if ($_=getLine($fhref)){
        push @blockInfo,$_;
    }
    return @blockInfo;
    } else {
    return ();
    }
}

sub getLine {
    my ($fhref) = @_;

    while(<$fhref>) {
    chop;
    if (/^ (.+)/){
        $lastLine.=$1;
#    print “Line is now:\n$lastLine\n”;
    } else {
        $thisLine=$lastLine;
        $lastLine=$_;
        return $thisLine;
    }
    }
# Take care of that last line
    if ($lastLine) {
    $thisLine=$lastLine;
    $lastLine=””;
    return $thisLine;
    }
    return $lastLine;
}
```

Example of Adding Custom Sun ONE Identity Server Attributes for Sun ONE Portal Server 6.2

This example shows the steps necessary for adding timezone and department attributes to Sun ONE Identity Server for use with Sun ONE Portal Server 6.2. You must first use the admin console to add `iplanetamuserservice` as a required service for all users. This permits custom attributes to be populated properly from the Sun ONE Identity Server SDK.

1. In the admin console, from the top level (`o=isp`), select service management and for service name Sun ONE Identity Server Configuration, select the icon next to Administration.
2. Scroll down the list of settings to Required Services and add `iplanetamuserservice`.
3. Click Save.
4. Restart Sun ONE Identity Server. For example:

   BaseDir`/SUNWam/bin/amserver start`

5. Add custom LDAP object class and attributes by adding the following to the `/usr/ldap/slapd-host/config/schema/99user.ldif` file using `ldapmodify`:

   ```
   objectclasses: (examplePerson-oid NAME ‘examplePerson’ DESC ‘example
   Person Extensions’ SUP top AUXILIARY MAY (example-am-user-timezone $
   example-am-user-deptno) )
   ```

   ```
   attributetypes: ( example-am-user-timezone-oid NAME
   ‘example-am-user-timezone’ DESC ‘example User Timezone and Region’
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN (‘user defined’) )
   ```

   ```
   attributetypes: ( example-am-user-deptno-oid NAME
   ‘example-am-user-deptno’ DESC ‘example User Department Number’ SYNTAX
   1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN (‘user defined’) )
   ```

   Note that the OID `1.3.6.1.4.1.1466.115.121.1.15` represents an LDAP string type.

6. Make of copy of the *BaseDir*`/SUNWam/config/xml/amUser.xml` file. If you want to add entries to the `amUser.xml` file, backup the original first.
7. Corresponding to the `iwtUser` extensions that you have made, add entries to the copy of the *BaseDir*`/SUNWam/config/xml/amUser.xml` file. These changes should be in accordance with the *BaseDir*`/SUNWam/dtd/sms.dtd` file. For more details, see the *Sun ONE Identity Server Programmer’s Guide*.

8. Add the following to the schema global description to refer to custom LDAP objectclass:

```
<Schema
  serviceHierarchy=”/DSAMEConfig/iPlanetAMUserService”
  i18nFileName=”amUser”
  i18nKey=”iplanet-am-user-service-description”>
  <Global>
    <AttributeSchema name=”serviceObjectClasses”
      type=”list”
      syntax=”string”
      i18nKey=””>
      <DefaultValues>
        <Value>iplanet-am-user-service</Value>
        <Value>examplePerson</Value>
      </DefaultValues>
    </AttributeSchema>
  </Global>
```

9. Add the following new custom attributes:

   ```
   <AttributeSchema name=”example-am-user-timezone”
    type=”single”
    syntax=”string”
    any=”display|optional|filter”
    i18nKey=”u121”>
   </AttributeSchema>
   <AttributeSchema name=”example-am-user-deptno”
    type=”single”
    syntax=”string”
    any=”display|optional|filter”
    i18nKey=”u122”>
   </AttributeSchema>
   ```

10. Using the `amadmin` utility, in *BaseDir*/`SUNWam/bin`, delete the existing user Service. For example:

    `./amadmin -u “uid=amadmin,ou=People,o=`*defaultorg***`,o=`***rootsuffix | dc=defaultorg***`,`** **`dc=`***rootsuffix***`” -w`** *password* **`--deleteService iPlanetAMUserService`**

11. Load the new, modified user service. For example:

    `./amadmin -u “uid=amadmin,ou=People,o=`*defaultorg***`,o=`***rootsuffix | dc=defaultorg***`,`** **`dc=`***rootsuffix***`” -w`** *password* **`--schema amUser.xml`**

12. Edit `u121` and `u122` strings in *BaseDir*/`SUNWam/locale/amUser.properties`.

13. Restart Sun ONE Portal Server,Sun ONE Identity Server, and Sun ONE Directory Server.

14. Shut down the LDAP server manually and start it again by issuing the following commands:

    *BaseDir***`/SUNWam/bin/amserver stop`**

    `/usr/ldap/slapd-`*host***`/stop-slapd`**

    *BaseDir***`/SUNWam/bin/amserver start`**

NOTE You must shutdown the LDAP server manually for the new 99user.dif schema extensions to take effect.

Appendix F

Authentication Framework Changes Between Sun ONE Portal Server 6.0 and Sun ONE Portal Server 6.2

NOTE	All instances of the Sun™ ONE Identity Server 5.1 product refer to what was formerly known as the iPlanet™ Directory Server Access Management Edition 5.1 product.

This appendix provides information on the changes to the authentication framework between Sun™ ONE Portal Server 6.0 and Sun™ ONE Portal Server 6.2 service that you need to know for migration purposes. The information in this appendix is not meant to be a replacement for the Javadocs.

This appendix contains the following sections:

- Authentication Framework
- Authentication Configuration and Property Files
- Rewriting the Authentication Modules

Authentication Framework

The Sun ONE Portal Server product uses the Sun™ ONE Identity Server authentication framework. The authentication framework has changed considerably between the Sun ONE Identity Server 5.1 release (used by the Sun ONE Portal Server 6.0 product) and Sun ONE Identity Server 6.1 release (used by the Sun ONE Portal Server 6.2 product).

The Sun ONE Identity Server 6.1 authentication framework provides the Authentication Service Provider Interface (SPI) to write custom authentication modules. The Authentication SPI implements the Java™ Authentication and Authorization Service (JAAS) LoginModule API, and provides methods to access the Authentication Service and module configuration properties. The Authentication SPI is defined in the `com.sun.identity.authentication.spi` package. The Authentication SPI contains the abstract class, `AMLoginModule,` used for writing pluggable authentication modules. To write an authentication module, you extend the `AMLoginModule` class. Writers of custom authentication modules need to understand and be familiar with JAAS, especially the Callback mechanism.

Authentication Configuration and Property Files

Authentication modules require several files to enable them to be recognized by the core authentication service and define their service attributes, localization properties, and module properties.

amAuth.xml File

The `amAuth.xml` file defines the “parent” core authentication service. This file is located in the *BaseDir*`/SUNWam/config/xml` directory. Whenever a new authentication service is created, attributes in this file must be modified in order to recognize an authentication module. Refer to “Modifying the amAuth.xml File” on page 68 for information on modifying this file.

XML Service File

Each authentication service must have its own service XML file. This file specifies the user and administrator configurable attributes and allows them to be managed using the Sun ONE Identity Server administration console. The name of the XML service file typically follows the format `amAuth`*ModuleName*`.xml` (for example, `amAuthSafeWord.xml` or `amAuthLDAP.xml`) and is located in the *BaseDir*`/SUNWam/config/xml` directory. This is migrated automatically by the Sun ONE Identity Server 6.1 migration script.

Localization Properties File

The localization properties file defines the language-specific screen text for the attribute names for the module. The name of the localization properties file typically follows the format amAuth*ModuleName*.properties (for example, amAuthSafeWord.properties or amAuthLDAP.properties) and is located in the *BaseDir*/SUNWam/locale directory under sub-directory appropriate to the locale (for example, en_US, or FR).

Authentication Module Configuration File

The authentication module configuration file specifies the authentication module credentials required by an identity (either user, service or application) in order to authenticate to the module. The name of the authentication module configuration file typically follows the format *ModuleName*.xml (for example, SafeWord.xml or LDAP.xml) and is located in the *BaseDir*/SUNWam/web-apps/services/config/auth/default directory.

The settings in this file define the user authentication screens properties. In Sun ONE Identity Server 5.1 implementation, this file was a .properties text file (for example, LDAP.properties) that contained certain keywords which were interpreted by the Sun ONE Identity Server 5.1 authentication service to render the screen. In Sun ONE Identity Server 6.1 implementation, this file is now a .xml file (for example, LDAP.xml) that describes the screen properties, data, and credential requirements to send to the authentication UI as JAAS Callbacks.This file needs to be manually translated to the new model, that is, into an XML file. The Auth_Module_Properties.dtd is the DTD file which defines all the different module screen states.

Table F-1shows the changes to the authentication module property types. This three-column table lists the property type in the first column. The second column shows the Sun ONE Identity Server 5.1 property name. The third column shows the Sun ONE Identity Server 6.1 property name.

Table F-1 Authentication Module Properties

Property Type	Sun ONE Identity Server 5.1 Property Name	Sun ONE Identity Server 6.1 Property Name
Screen Number	SCREEN	order=*order_number*
Page Timeout Value	TIMEOUT	timeout=*value*
Template Name	HTML	template=*template_name*
Text Value	TEXT	header=*text*

Table F-1 Authentication Module Properties

Property Type	Sun ONE Identity Server 5.1 Property Name	Sun ONE Identity Server 6.1 Property Name
User Name	TOKEN	NameCallback
Password	PASSWORD	PasswordCallback
Attribute Name	ATTRIBUTE	Callback (depends on the type of attribute)
Image Name	IMAGE	None (image can be specified in the customized JSP)

For example, the following is a sample `LDAP.properties` file from Sun ONE Identity Server 5.1:

```
SCREEN
TIMEOUT 120
TEXT LDAP Authentication
TOKEN Enter UserId
PASSWORD Enter Password

SCREEN
TIMEOUT 120
TEXT Password Expiring Please Change
PASSWORD <REPLACE><BR> Enter Current Password
PASSWORD Enter New Password
PASSWORD Confirm New Password
```

After conversion to the Sun ONE Identity Server 6.1 format, the file contents would be as follows:

```
<!DOCTYPE ModuleProperties PUBLIC "=//iPlanet//Authentication Module Properties XML
Interface 1.0 DTD//EN"

"jar://com/sun/identity/authentication/Auth_Module_Properties.dtd">


<ModuleProperties moduleName="LDAP" version="1.0" >

    <Callbacks length="2" order="1" timeout="120"

          header="This server uses LDAP Authentication" >

        <NameCallback>

            <Prompt> User Name: </Prompt>

        </NameCallback>

        <PasswordCallback echoPassword="false" >

            <Prompt> Password: </Prompt>

        </PasswordCallback>

  </Callbacks>


 Callbacks length="4" order="2" timeout="120" header="Change
Password&lt;BR/&gt;#REPLACE#&lt;BR/&gt;" >
```

```
  <PasswordCallback echoPassword="false" >
             <Prompt>Old Password </Prompt>
         </PasswordCallback>
         <PasswordCallback echoPassword="false" >
             <Prompt> New Password </Prompt>
         </PasswordCallback>
         <PasswordCallback echoPassword="false" >
             <Prompt> Confirm Password </Prompt>
         </PasswordCallback>

  <Prompt> New Password </Prompt>
         </PasswordCallback>
         <PasswordCallback echoPassword="false" >
             <Prompt> Confirm Password </Prompt>
         </PasswordCallback>

 <ConfirmationCallback>
             <OptionValues>
                 <OptionValue>
                     <Value> Submit </Value>
                 </OptionValue>
                 <OptionValue>
                     <Value> Cancel </Value>
                 </OptionValue>
             </OptionValues>
         </ConfirmationCallback>
     </Callbacks>
</ModuleProperties>
```

Rewriting the Authentication Modules

In Sun ONE Identity Server 6.1, authentication modules extend the `AMLoginModule` SPI which extends the JAAS LoginModule Interface and provides other convenient methods.

Table F-2 lists the methods in Sun ONE Identity Server 5.1 AuthenticationModule SPI and the equivalent method in Sun ONE Identity Server 6.1 AMLoginModule SPI. This three-column table lists the method in the first column. The second column shows the Sun ONE Identity Server 5.1 method. The third column shows the Sun ONE Identity Server 6.1 method. Refer to Sun ONE Identity Server Javadocs for exact details on these methods.

.

NOTE Since the authentication framework has changed dramatically between Sun ONE Identity Server 5.1 and Sun ONE Identity Server 6.0 some Sun ONE Identity Server 5.1 methods do not have an equivalent in the Sun ONE Identity Server 6.0 JAAS-based framework.

Table F-2 Authentication Module Properties

Method	**Sun ONE Identity Server 5.1 Authentication Module SPI**	**Sun ONE Identity Server 6.1 AMLoginModule SPI**
init method Must be implemented by the LoginModule.	`void init()`	`void init(javax.security.auth.Subject.subject, java.util.Map sharedState, java.util.Map options)`
LoginModule process method Must be implemented by the LoginModule.	`void validate()`	`int process(javax.security.auth.callback.Callback[] callbacks, int state)`
get user id or principal Must be implemented by the LoginModule.	`String getUserTokenID`	`javax.security.Principal getPrincipal()`
get all token names for the current authentication state	`Properties getAllTokenNames`	
get all user entered tokens in the current authentication state	`String getAllTokensForState()`	

Table F-2 Authentication Module Properties

Method	Sun ONE Identity Server 5.1 Authentication Module SPI	Sun ONE Identity Server 6.1 AMLoginModule SPI
get all user entered tokens in the specified state	`String getAllTokensForState()`	
get auth level	`int getAuthLevel()`	`int getAuthLevel()`
get current state	`int getCurrentState()`	`int getCurrentStatel()`
get the HttpServletRequest object	`HttpServletRequest get HttpServletRequest()`	`HttpServletRequest get HttpServletRequest()`
get the HttpServletResponse object	`HttpServletResponse get HttpServletResponse()`	`HttpServletResponse get HttpServletResponse()`
get the locale	`String getLocale()`	`String getLocale()`
get the number of screens	`int getNumberOfScreens()`	
get the number of authentication states for the module	`int getNumberOfStates()`	`int getNumberOfStates()`
get the number of tokens in the current authentication state	`int getNumberOfTokens()`	
get the number of tokens for the specified authentication state	`int getNumberOfTokensForState(int stateNumber)`	
get the organization attributes for the organization specified	`java.util.Map getOrgProfile(String orgDN)`	`java.util.Map getOrgProfile(String orgDN) throws javax.security.auth.login.LoginException`
get the service template attribute defined for the organization	`java.util.Map getOrgServiceTemplate(String orgDN, String serviceName)`	`java.util.Map getOrgServiceTemplate(String orgDN, String serviceName) throws javax.security.auth.login.LoginException`
check if persistent cookie is on or off	`boolean getPersistentCookieOn()`	`boolean getPersistentCookieOn()`
return orgDN for this session	`String getRequestOrg()`	`String getRequestOrg()`

Table F-2 Authentication Module Properties

Method	Sun ONE Identity Server 5.1 Authentication Module SPI	Sun ONE Identity Server 6.1 AMLoginModule SPI
get service configuration attributes	`java.util.Map getServiceConfig(String name)`	`java.util.Map getServiceConfig(String name) throws javax.security.auth.login.LoginException`
get the session ID which is the unique key for this authentication session	`String getSessionId()`	`getSessionId()`
get authentication SSO session	`SSOToken getSSOSessionId()`	`SSOToken getSSOSessionId()`
get the user entered value for the specified token in the current authentication state	`String getToken()`	
get the user entered value for the specified authentication state	`String getToken(int index)`	`java.security.auth.callback.Callback getCallback(int index) throws javax.security.auth.login.LoginException`
get the user entered value for the specified token in the current authentication state	`String getToken(String name)`	
get the user profile for the user specified	`AMUser getUserProfile(String userDN)`	`AMUser getUserProfile(String userDN) throws javax.security.auth.login.LoginException`
get the property specified from the user session	`Sting getUserSessionProperty(String name)`	`Sting getUserSessionProperty(String name) throws javax.security.auth.login.LoginException`
log out	`logout()`	`AMLoginModule implements the JAAS logout() method`
set the authentication level for the session	`void setAuthLevel(String authLevel)`	`boolean setAuthLevel(int auth_level)`

Table F-2 Authentication Module Properties

Method	Sun ONE Identity Server 5.1 Authentication Module SPI	Sun ONE Identity Server 6.1 AMLoginModule SPI

set the default URL for the user to redirect to on authentication	`void setDefaultURL(String url)`	`void setLoginSuccessURL(String url) throws javax.security.auth.login.LoginException`
		`void setLoginFailureURL(String url) throws javax.security.auth.login.LoginException`

set the number of next login screen that will be sent to the user during authentication	`void setNextScreen(int num)`	
set the user’s organization	`void setOrg(String orgDN)`	
turn the persistent cookie on for this auth session	`int setPersistentCookieOn()`	`boolean setPersistentCookieOn()`

replace header test on the screen specified	`setReplaceText(int screenNumber, int token, String text)`
	`setReplaceText(int token, String text)`
		`void replaceCallback(int state, int index, callback) throws javax.security.auth.login.LoginException`
		`void replaceHeader(int state, String header) throws javax.security.auth.login.LoginException`

set a property in the user session	`void setUserSessionProperty(String name, String value)`	`void setUserSessionProperty(String name, String value) throws javax.security.auth.login.LoginException`
get attribute name for the specified callback in the state specified		`String getAttribute(int state, int index)`
check if a callback is required to have an input		`boolean isRequired(int state, int index)`
set the failure UserID. This ID is used to log failed authentication is the logs		`void setFailureID(String userID)`

Table F-2 Authentication Module Properties

Method	Sun ONE Identity Server 5.1 Authentication Module SPI	Sun ONE Identity Server 6.1 AMLoginModule SPI
set attribute values which will be used when authentication service is configured dynamically by user		`void setUserAttributes(java.util.Map attribute ValuePairs)`
reset callback to original callback for specified state and callback index.		`void resetCallback(int state, int index)`

Appendix G

Example of Sun ONE Portal Server 3.0 Provider Migrated to Sun ONE Portal Server 6.x

NOTE All instances of the Sun™ ONE Portal Server 3.0 product refer to what were formerly known as the iPlanet™ Portal Server 3.0, Service Pack 3a, iPlanet™ Portal Server 3.0, Service Pack 4 products, and iPlanet™ Portal Server 3.0, Service Pack 5 products.

This appendix provides an example of a Sun ONE Portal Server 3.0 provider migrated to run on a Sun™ ONE Portal Server 6.x implementation.

Code Example G-1 shows a code sample of a quotations provider implemented for the Sun ONE Portal Server 3.0 product.

Code Example G-1 Code Sample of a Quotations Provider for Sun ONE Portal Server 3.0.

```
<!--
Attributes, specific
-->


<iwt:Att name="iwtQuotationProvider-possibleCategories"
      desc="Possible Quotation Categories"
      type="stringlist"
      idx="possible_quotation_categories"
      userConfigurable="TRUE">
      <Val>Computers</Val>
      <Val>Freedom</Val>
      <Val>Science</Val>
      <Val>Work</Val>
      <Rperm>ADMIN</Rperm><Rperm>OWNER</Rperm>
      <Wperm>ADMIN</Wperm>
      </iwt:Att>
```

Code Example G-1 Code Sample of a Quotations Provider for Sun ONE Portal Server 3.0.

```
<iwt:Att name="iwtQuotationProvider-selectedCategories"
      desc="Selected Quotation Categories"
      type="stringlist"
      idx="selected_categories"
      userConfigurable="TRUE">
      <Val>Computers</Val>
      <Val>Freedom</Val>
      <Val>Science</Val>
      <Val>Work</Val>
      <Rperm>ADMIN</Rperm><Rperm>OWNER</Rperm>
      <Wperm>ADMIN</Wperm><Wperm>OWNER</Wperm>
      </iwt:Att>
<iwt:Att name="iwtQuotationProvider-displayCategories"
      desc="Display category along with quotation?"
      type="boolean"
      idx="display_categories"
      userConfigurable="TRUE">
      <Val>true</Val>
      <Rperm>ADMIN</Rperm><Rperm>OWNER</Rperm>
      <Wperm>ADMIN</Wperm><Wperm>OWNER</Wperm>
      </iwt:Att>

      <iwt:Att name="iwtQuotationProvider-fileLocation"
      desc="Location of the quotations file"
      type="string"
      idx="file_location"
      userConfigurable="TRUE">
      <Val>/var/tmp/quotations</Val>
      <Rperm>ADMIN</Rperm><Rperm>OWNER</Rperm>
      <Wperm>ADMIN</Wperm>
      </iwt:Att>
```

Code Example G-1 shows a code sample of a quotations provider implemented for the Sun ONE Portal Server 6.x product.

Code Example G-2 Code Sample of a Quotations provider for Sun ONE Portal Server 6.x.

```
// QuotationProvider.java
 //   This is a sample desktop provider application used to display
 //   famous/humorous quotations on the desktop.
 //
 //   It shows how to use the BaseProvider API to:
 //     - display html content on the desktop
 //     - allow the user to edit attributes
 //
// Where the QuotationProvider.class file can be found.
 package custom;

 // Use some basic Java classes
 import java.lang.*;
 import java.io.*;
 import java.util.*;
 import java.net.*;
// Use some basic Java Servlet classes
 import javax.servlet.*;
 import javax.servlet.http.*;
// Use some iPS classes
 import com.sun.portal.providers.*;
// The QuotationProvider Class
 public class QuotationProvider extends ProfileProviderAdapter {
     // The constructor-don’t need to do any special initialization
     // so just invoke the ProfileProviderAdapter’s constructor
     public QuotationProvider() {
         super();
     }
```

Code Example G-2 Code Sample of a Quotations provider for Sun ONE Portal Server 6.x.

```
// Generate the HTML that will be displayed in this content
// provider’s area on the desktop
     public StringBuffer getContent(HttpServletRequest req,
     HttpServletResponse res) throws ProviderException {

         StringBuffer content = new StringBuffer();
         // Contains the HTML to be displayed
         Hashtable quotesHash = new Hashtable();
         // A Hash of quote vectors,
                // one vector per category
         BufferedReader quotesReader;
         // Used to read the quotes
         String quote;    // A single quotation
         Random randomGenerator = new Random();
               // To randomly pick quotes

     // Read in the quotations into a Hashtable of Vectors,
     // each containg the quotations of a specific category.
     // If we can’t read the quotations file it’s okay - we
     // just won’t display any quotes later.
     // The category and quotation are separated by a "|"
     // in the quotations file.

    try {
       String quotationFileName = getStringProperty("fileLocation");
       quotesReader = new BufferedReader(new FileReader(new
File(quotationFileName)));
       while ((quote = quotesReader.readLine()) != null) {
       String type = quote.substring(0,quote.indexOf(’|’));
           if (!quotesHash.containsKey(type)) {
           quotesHash.put(type,new Vector());
           }

((Vector)(quotesHash.get(type))).addElement(quote.substring(quote.indexOf(’|’
)+1));
          }
     } catch (Exception e) {
   }

// Determine is user’s preference is to display the category
// along with a quotation.
boolean displayCategories = getBooleanProperty("displayCategories");
 // Get the possible quotation categories.
       List categoriesSel =
          getListProperty("selectedCategories");
```

Code Example G-2 Code Sample of a Quotations provider for Sun ONE Portal Server 6.x.

```
// Print a quotation for each category selected by the user.
       for (int i = 0; i < categoriesSel.size(); i++) {
         String category = (String)categoriesSel.get(i);

         // If the user wanted to display the category along with
         // the quotation then do so.
         if (displayCategories) {
           content.append("<B><I>");
           content.append(category);
           content.append("</B></I>");
         }

// If there are any quotations for this category, pick a random
 // one and display it, otherwise note that we didn’t find one.
if (quotesHash.containsKey(category)) {
  Vector quoteVector = (Vector)quotesHash.get(category);
  int whichQuoteIndex =
Math.abs(randomGenerator.nextInt())%(quoteVector).size();
    content.append("<UL>");
    content.append("<LI>");
    content.append((String)(quoteVector.elementAt(whichQuoteIndex)));
    content.append("</LI>");
    content.append("</UL>");
  } else {
    content.append("<UL>");
    content.append("<LI>");
    content.append("(no quotes of this type found)");
    content.append("</LI>");
    content.append("</UL>");
    }
 }

// Now return the HTML we have constructed.
 return content;
}
// Generate the HTML that will be displayed to let the user set
 // his/her preferneces.
 public StringBuffer getEdit(HttpServletRequest req,
     HttpServletResponse res) throws ProviderException {
 // Contains the HTML to be displayed
   StringBuffer content = new StringBuffer();
```

Code Example G-2 Code Sample of a Quotations provider for Sun ONE Portal Server 6.x.

```
// Display things in a single column table - so things line up.
    content.append("<P><TABLE>");

 // Let the user choose which categories s/he wants.
    content.append("<TR>");
    content.append("<TD VALIGN='top' HALIGN='left'>");
    content.append("Types of Quotes to Display");
    content.append("</TD>");
// Get the user’s selected categories
List selectedCategories = getListProperty("selectedCategories");
// List the possible categories, marking those that have already
// been selected by the user.
content.append("<TD VALIGN='top' HALIGN='left'>");
List possibleCategories = getListProperty("possibleCategories")
for (int i = 0; i < possibleCategories.size(); i++) {
      String category = (String)possibleCategories.get(i);
       content.append("<INPUT TYPE='checkbox' NAME='category-");
       content.append(category);
       content.append("' VALUE='");
       content.append(category);
       content.append("'");
       if (selectedCategories.contains(category)) {
          content.append(" CHECKED");
       }
          content.append(">");
          content.append(category);
          content.append("<BR>");
       }
         content.append("</TD>");
         content.append("</TR>");
```

Code Example G-2 Code Sample of a Quotations provider for Sun ONE Portal Server 6.x.

```
// Let the user decide if they want to display a category with
 // each quotation.
    content.append("<TR>");
    content.append("<TD VALIGN='top' HALIGN='left'>");
    content.append("<INPUT TYPE='checkboX' NAME='display_categories'
VALUE='yes'");

if (getBooleanProperty("displayCategories")) {
    content.append(" CHECKED");
    }
    content.append(">Display the Category Along With the Quotation");
    content.append("</TD>");
    content.append("</TR>");
    content.append("</TABLE>");
   return content;
  }
// Handle the submittal of the edit form.
public URL processEdit(HttpServletRequest req,
  HttpServletResponse res) throws ProviderException {

List categorySelections = new ArrayList(); // Categories user picked
boolean displayCategories = false;        // Display category too ?
for (Enumeration e = req.getParameterNames(); e.hasMoreElements(); ) {
// Check all the passed parameters and add the selected categories
// as well as check to see if user opted to display categories.
String parameter = (String)e.nextElement();
if (parameter.equals("display_categories")) {
 displayCategories = true;
 } else if (parameter.startsWith("category-")) {
 String []category = (String [])req.getParameterValues(parameter);
 categorySelections.add(category[0]);
      }
 }
// Set the list of selected categories for later storing in the
// profile database.
setListProperty("selectedCategories", categorySelections);

// Set whether the user wants to display categories or not for later
// storing in the profile database.
   setBooleanProperty("displayCategories", displayCategories);

   return null;

   }
}
```

Code Example G-2 Code Sample of a Quotations provider for Sun ONE Portal Server 6.x.

```
<Channel name="QuotationChannel" provider="QuotationProvider">
    <Properties>
         <String name="title" value=" Quotations Provider"/>
         <String name="description" value="This channel displays the
quotations for selected categories at random."/>
         <String name="refreshTime" value="0" advanced="true"/>
         <Collection name="possibleCategories">
             <String value="Computers"/>
             <String value="Freedom"/>
             <String value="Science"/>
             <String value="Work"/>
         </Collection>
         <Collection name="selectedCategories">
             <String value="Computers"/>
             <String value="Freedom"/>
             <String value="Science"/>
             <String value="Work"/>
         </Collection>
         <Boolean name="displayCategories" value="true"/>
         <String name="fileLocation" value="/var/tmp/quotations"/>
     </Properties>
 </Channel>
```

Code Example G-2 Code Sample of a Quotations provider for Sun ONE Portal Server 6.x.

```
  <Provider name="QuotationProvider" class="custom.QuotationProvider">
     <Properties>
         <String name="title" value=" Quotations Provider"/>
         <String name="description" value="This channel displays the
quotations for selected categories at random."/>
         <Boolean name="isEditable" value="true" advanced="true"/>
         <String name="width" value="thick"/>
         <String name="editType" value="edit_subset"/>
         <String name="refreshTime" value="0" advanced="true"/>
         <Collection name="possibleCategories">
             <String value="Computers"/>
             <String value="Freedom"/>
             <String value="Science"/>
             <String value="Work"/>
         </Collection>
         <Collection name="selectedCategories">
             <String value="Computers"/>
             <String value="Freedom"/>
             <String value="Science"/>
             <String value="Work"/>
         </Collection>
         <String name="fontFace1" value="Sans-serif"/>
         <String name="productName" value="Sun ONE Portal Server"/>
         <Boolean name="displayCategories" value="true"/>
         <String name="fileLocation" value="/var/tmp/quotations"/>
     </Properties>
 </Provider>
```

Code Example G-2 Code Sample of a Quotations provider for Sun ONE Portal Server 6.x.

```
Computers|"If you can’t beat your computer at chess, try kickboxing." -Anon.
Computers|"A computer lets you make more mistakes faster than any invention in
human history - with the possible exceptions of handguns and
tequila." -Mitch Ratliffe
Computers|"If the automobile had followed the same development cycle as the
computer, a Rolls-Royce would today cost $100, get one million
miles to the gallon, and explode once a year, killing everyone inside."
-Robert X Cringely
Computers|"What goes up must come down. Ask any system administrator." -Anon.
Computers|"Perl - The only language that looks the same before and after RSA
encryption." -Keith Bostic
Computers|"C makes it easy to shoot yourself in the foot. C++ makes it harder,
but when you do, it blows away your whole leg." -Bjarne
Stroustrup
Computers|"There are two major products that come out of Berkeley: LSD and
UNIX. We don’t believe this to be a coincidence." -Jeremy S.
Anderson
```

Code Example G-2 Code Sample of a Quotations provider for Sun ONE Portal Server 6.x.

```
Freedom|"Live free or die." -New Hampshire State Motto
Freedom|"Man is free at the moment he wishes to be." -Voltaire
Freedom|"People demand freedom of speech as a compensation for the freedom of
thought which they seldom use." -Kierkegaard
Freedom|"The right to be heard does not autmatically include the right to be
taken seriously." -Hubert Humphrey
Freedom|"Extremism in the defense of liberty is no vice. And moderation in the
pursuit of justice is no virtue." -Barry Goldwater
Freedom|"I disapprove of what you say, but I will defend to the death your
right to say it." -Voltaire
Science|"Give me a lever long enough and a fulcrum on which to place it, and I
shall move the world." -Archimedes, Pappus of Alexandria
Science|"A scientific truth does not triumph by convincing its opponents and
making them see the light, but rather because its opponents
eventually die and a new generation grows up that is familiar with it."
-Maxwell Planck
Science|"Every great advance in science has issued from a new audacity of
imagination." -John Dewey, The Quest for Certainty
Stupid|"Caution: Cape does not enable user to fly." -Batman Costume warning
label
Stupid|"Your food stamps will be stopped effective March 1992 because we
received notice that you passed away. May God bless you. You may
reapply if there is a change in your circumstances." -Department of Social
Services, Greenville, South Carolina
Stupid|"We’re going to turn this team around 360 degrees." -Jason Kidd
Stupid|"Traditionally, most of Australia’s imports come from overseas."
-Keppel Enderbery
Stupid|"If you’re killed, you’ve lost a very important part of your life."
-Brooke Shields
Stupid|"We don’t necessarily discriminate. We simply exclude certain types of
people." -Colonel Gerald Wellman, ROTC Instructor
Stupid|"Men, I want you just thinking of one word all season. One word and one
word only: Super Bowl." -Bill Peterson, football coach
Stupid|"I have opinions of my own -- strong opinions --but I don’t always
agree with them." -George Bush, US President
```

Index