Sun OpenSSO Enterprise 8.0 Integration Guide

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

Part No: 820–4729–12
December 2009

091223@23031

Contents

Preface

The *Sun™ OpenSSO Enterprise Integration Guide* provides information to help you determine which OpenSSO Enterprise to use in your deployment. This guide contains deployment architecture diagrams, process flow diagrams, and benefits/tradeoffs analysis for various OpenSSO Enterprise features.

The following topics are contained in this Preface:

Who Should Use This Guide

- IT professionals responsible for architecting enterprise solutions.
- Company executives responsible for evaluating enterprise solutions and for approving IT vendors and purchases.
- System integrator and administrators responsible for deploying OpenSSO Enterprise and related components.

Before You Read This Guide

Readers should be familiar with the following components and concepts:

- OpenSSO Enterprise technical concepts, as described in the *Sun OpenSSO Enterprise 8.0 Technical Overview*
- Deployment platform: Solaris™, Linux, or Windows operating system

- Web container that will run OpenSSO Enterprise, such as Sun Java System Application Server, Sun Java System Web Server, BEA WebLogic, or IBM WebSphere Application Server
- Technical concepts: Lightweight Directory Access Protocol (LDAP), Java™ technology, JavaServer Pages™ (JSP™) technology, HyperText Transfer Protocol (HTTP), HyperText Markup Language (HTML), and eXtensible Markup Language (XML)

Related Documentation

Related documentation is available as follows:

- "OpenSSO Enterprise Documentation Set" on page 10
- "Related Product Documentation" on page 11

OpenSSO Enterprise Documentation Set

The following table describes the OpenSSO Enterprise documentation set.

TABLE P–1 OpenSSO Enterprise Documentation Set

Title	Description
Sun OpenSSO Enterprise 8.0 Release Notes	Describes new features, installation notes, and known issues and limitations. The Release Notes are updated periodically after the initial release to describe any new features, patches, or problems.
Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide	Provides information about installing and configuring OpenSSO Enterprise including OpenSSO Enterprise server, Administration Console only, client SDK, scripts and utilities, Distributed Authentication UI server, and session failover.
Sun OpenSSO Enterprise 8.0 Technical Overview	Provides an overview of how components work together to consolidate access control functions, and to protect enterprise assets and web-based applications. It also explains basic concepts and terminology.
Sun OpenSSO Enterprise 8.0 Deployment Planning Guide	Provides planning and deployment solutions for OpenSSO Enterprise.
Deployment Example: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0	Provides step-by-step instructions for deploying OpenSSO Enterprise in a single sign-on environment using load balancers and redundant systems for high availability.
Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0	Provides step-by-step instructions for deploying OpenSSO Enterprise to achieve identity federation among an Identity Provider and multiple Service Providers.

TABLE P–1 OpenSSO Enterprise Documentation Set *(Continued)*

Title	Description
Sun OpenSSO Enterprise 8.0 Administration Guide	Describes how to use the OpenSSO Enterprise Administration Console as well as how to manage user and service data using the command-line interface (CLI).
Sun OpenSSO Enterprise 8.0 Administration Reference	Provides reference information for the OpenSSO Enterprise command-line interface (CLI), configuration attributes, log files, and error codes.
Sun OpenSSO Enterprise 8.0 Developer's Guide	Provides information about customizing OpenSSO Enterprise and integrating its functionality into an organization's current technical infrastructure. It also provides details about the programmatic aspects of the product and its API.
Sun OpenSSO Enterprise 8.0 C API Reference for Application and Web Policy Agent Developers	Provides summaries of data types, structures, and functions that make up the public OpenSSO Enterprise C APIs.
Sun OpenSSO Enterprise 8.0 Java API Reference	Provides information about the implementation of Java packages in OpenSSO Enterprise.
Sun OpenSSO Enterprise 8.0 Performance Tuning Guide	Provides information about how to tune OpenSSO Enterprise and its related components for optimal performance.
Sun OpenSSO Enterprise 8.0 Integration Guide (This book)	(This book) Provides information about how to integrate Sun Identity Manager, CA SiteMinder, or Oracle Access Manager with OpenSSO Enterprise.
Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for J2EE Agents	Provides an overview of version 3.0 policy agents.

Related Product Documentation

The following table provides links to documentation collections for related products.

TABLE P–2 Related Product Documentation

Product	Link
Sun Java System Directory Server 6.3	`http://docs.sun.com/coll/1224.4`
Sun Java System Web Server 7.0 Update 3	`http://docs.sun.com/coll/1653.3`
Sun Java System Application Server 9.1	`http://docs.sun.com/coll/1343.4`
Sun Java System Message Queue 4.1	`http://docs.sun.com/coll/1307.3`
Sun Java System Web Proxy Server 4.0.6	`http://docs.sun.com/coll/1311.6`

TABLE P–2 Related Product Documentation *(Continued)*

Product	Link
Sun Identity Manager 8.0	http://docs.sun.com/app/docs/coll/1514.5

Searching Sun Product Documentation

Besides searching Sun product documentation from the docs.sun.com℠ web site, you can use a search engine by typing the following syntax in the search field:

search-term `site:docs.sun.com`

For example, to search for "broker," type the following:

```
broker site:docs.sun.com
```

To include other Sun web sites in your search (for example, java.sun.com, www.sun.com, and developers.sun.com), use `sun.com` in place of `docs.sun.com` in the search field.

Related Third-Party Web Site References

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

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

Documentation, Support, and Training

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

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

Typographic Conventions

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

TABLE P–3 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

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

TABLE P–4 Shell Prompts

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

Default Paths and Directory Names

The OpenSSO Enterprise documentation uses the following terms to represent default paths and directory names:

TABLE P–5 Default Paths and Directory Names

Term	Description
zip-root	Represents the directory where the `opensso_enterprise_80.zip` file is unzipped.

OpenSSO-Deploy-base	Represents the deployment directory where the web container deploys the `opensso.war` file.
	
	This value varies depending on the web container. To determine the value of *OpenSSO-Deploy-base*, view the file name in the `.openssocfg` directory, which resides in the home directory of the user who deployed the `opensso.war` file. For example, consider this scenario with Application Server 9.1 as the web container:
	■ Application Server 9.1 is installed in the default directory: `/opt/SUNWappserver`.
	■ The `opensso.war` file is deployed by super user (`root`) on Application Server 9.1.
	
	The `.openssocfg` directory is in the `root` home directory (`/`), and the file name in `.openssocfg` is:
	
	`AMConfig_opt_SUNWappserver_domains_domain1_applications_j2ee-modules_opensso_`
	
	Then, the value for *OpenSSO-Deploy-base* is:
	
	`/opt/SUNWappserver/domains/domain1/applications/j2ee-modules/opensso`

ConfigurationDirectory	Represents the name of the configuration directory specified during the initial configuration of OpenSSO Enterprise server instance using the Configurator. The default is `opensso` in the home directory of the user running the Configurator. Thus, if the Configurator is run by `root`, *ConfigurationDirectory* is `/opensso`.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to `http://docs.sun.com` and click Send comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the guide or at the top of the document.

For example, the title of this guide is the *Sun OpenSSO Enterprise Integration Guide*, and the part number is 820-4729.

◆ ◆ ◆ CHAPTER 1

Integrating Sun Identity Manager

This chapter provides instructions for configuring Sun OpenSSO Enterprise to work with Sun Identity Manager. The examples in this chapter demonstrate how to configure OpenSSO Enterprise to protect Identity Manager, and to allow single sign-on login to the Identity Manager user and administrator interface. The examples also demonstrate how to configure Identity Manager to provision users and roles to OpenSSO Enterprise.

It is possible to configure the deployment for only single sign-on, or for only provisioning. If you do not require single sign-on between OpenSSO Enterprise and Identity Manager, then the OpenSSO Enterprise Policy Agent does not need to be installed or configured. You can disregard the steps that involve the OpenSSO Enterprise Policy Agent.

This chapter contains the following topics:

About the Deployment

In this deployment, OpenSSO Enterprise is installed in the Realm mode of operation. The OpenSSO Enterprise data store is configured to store configuration data.

A sub-realm named `idm` is created on OpenSSO Enterprise. The user data store for this sub-realm is a Sun Directory Server data store that has the OpenSSO Enterprise schema loaded into it. This sub-realm is used later when configuring the policy agent. The policy agent is deployed on Identity Manager to regulate access to the Identity Manager server. Identity Manager uses the MySQL database as its data store. Identity Manager can be configured to use MYSQL or Oracle databases as its configuration data store.

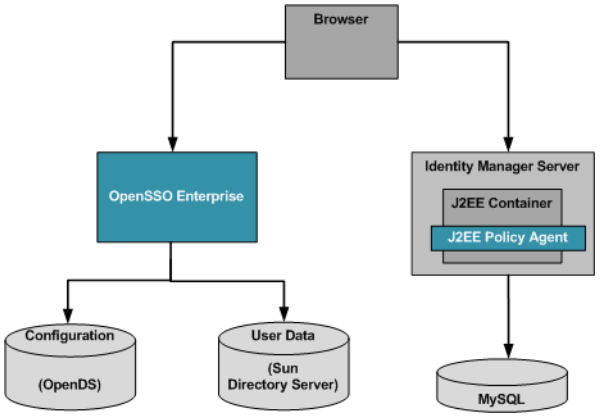


FIGURE 1–1 Deployment Architecture

About Sun Identity Manager

Sun Identity Manager enables you to securely and efficiently manage and user access to accounts and resources, and to distribute access management overhead. By mapping Identity Manager objects to the entities you manage such as users and resources, you significantly increase the efficiency of your operations.

The Identity Manager solution enables you to:

- Manage account access to a large variety of systems and resources.
- Securely manage dynamic account information for each user's array of accounts.
- Set up delegated rights to create and manage user account data.
- Handle large numbers of enterprise resources, as well as an increasingly large number of extranet customers and partners.

- Securely authorize user access to enterprise information systems. With Identity Manager, you have fully integrated functionality to grant, manage, and revoke access privileges across internal and external organizations.

See the Sun Identity Manager product documentation for more information. (`http://docs.sun.com/app/docs/coll/1514.5`)

About Sun Directory Server Enterprise Edition

Sun Directory Server Enterprise Edition provides secure, highly available, scalable directory services for storing and managing identity data. Directory Server Enterprise Edition is the foundation of an enterprise identity infrastructure. It enables mission-critical enterprise applications and large-scale extranet applications to access consistent and reliable identity data. Directory Server Enterprise Edition provides a central repository for storing and managing identity profiles, access privileges, application and network resource information. It also provides secure, on-demand synchronization of passwords, users, and groups with Microsoft Active Directory. See the Sun Java System Directory Server 6.3 product documentation (`http://docs.sun.com/coll/1224.4`).

About Sun MySQL

MySQL is the world's most popular open source database software. MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications. Powerful administration features enable users to fine-tune the server to optimize performance for the particular details of an embedded or bundled application. Plus, a pluggable storage engine architecture enables you to mix and match storage engines or just use what you need for an efficient optimized footprint.

Note – Identity Manager supports MySQL as a database resource in development or production deployments. MySQL is only supported as a repository database server in development deployment. See the MySQL documentation at`http://www.mysql.com/doc/` for more details.

Software Versions Used in the Deployment

TABLE 1–1 Software Versions Used in the Deployment

Product	Download Location
Sun Solaris Operating System 10	http://www.sun.com/software/solaris/get.jsp
JDK 5.0 Update 16	http://java.sun.com/javase/downloads/index_jdk5.jsp
OpenSSO Enterprise 8.0 (Build 5b)	https://opensso.dev.java.net/public/use/index.html#stableopensso
Sun Identity Manager 8.0	
Sun Java Web Server 7.0	Sun Java System Web Server 7.0 Update 2
Sun Java Application Server 9.1	Sun Java Application Server 9.1
Sun Java Directory Server 6.3	http://www.sun.com/software/products/directory_srvr_ee/get1.jsp Choose Directory Server Enterprise Edition 6.x.
MySQL 5.0 Identity Manager supports MySQL as a database resource in development or production deployments. MySQL is only supported as a repository database server in development deployment. See http://www.mysql.com/ for more details.	http://dev.mysql.com/downloads/mysql/5.0.html#solaris
MySQL Connector/J 5.0	http://dev.mysql.com/downloads/
Sun Java AM Policy Agent 3.0 (for Sun Application Server 9.1)	https://opensso.dev.java.net/public/use/index.html#stableagent
NetBeans IDE 6.0	http://download.netbeans.org/netbeans/6.0/final/
Sun Identity Manager IDE Plug-in 8.0 Beta 1	https://identitymanageride.dev.java.net/

Installing and Configuring MySQL

To install and configure MySQL, follow these steps:

1. Install MySQL
2. Configure MySQL

▼ To Install MySQL

1 Follow the installation instructions provided at the MySQL website.

See http://dev.mysql.com/doc/refman/5.0/en/installing-binary.html.

The following is output from an installation session:

```
# groupadd mysql

# useradd -g mysql mysql

# pwd
/opt/MySQL

# ls -al
total 106
drwxr-xr-x  14 root     root         512 Jan  2 12:48 .
drwxr-xr-x  30 root     sys         1024 Jan  2 12:40 ..
drwxr-xr-x   2 root     root        2048 Jan  2 12:48 bin
-rwxr-xr-x   1 root     root         801 Jan  2 12:48 configure
-rw-r--r--   1 root     root       19071 Jan  2 12:48 COPYING
drwxr-x---   4 root     root         512 Jan  2 12:48 data
drwxr-xr-x   2 root     root         512 Jan  2 12:48 docs
-rw-r--r--   1 root     root        5139 Jan  2 12:48 EXCEPTIONS-CLIENT
drwxr-xr-x   3 root     root        1536 Jan  2 12:48 include
-rw-r--r--   1 root     root        8528 Jan  2 12:48 INSTALL-BINARY
drwxr-xr-x   2 root     root         512 Jan  2 12:48 lib
drwxr-xr-x   4 root     root         512 Jan  2 12:48 man
drwxr-xr-x   9 root     root         512 Jan  2 12:48 mysql-test
-rw-r--r--   1 root     root        1410 Jan  2 12:48 README
drwxr-xr-x   2 root     root         512 Jan  2 12:48 scripts
drwxr-xr-x   3 root     root         512 Jan  2 12:48 share
drwxr-xr-x   5 root     root        1024 Jan  2 12:48 sql-bench
drwxr-xr-x   2 root     root         512 Jan  2 12:48 support-files
drwxr-xr-x   2 root     root         512 Jan  2 12:48 tests

# chown -R mysql .

# chgrp -R mysql .

# ls -al
total 106
drwxr-xr-x  14 mysql    mysql        512 Jan  2 12:48 .
drwxr-xr-x  30 root     sys         1024 Jan  2 12:40 ..
drwxr-xr-x   2 mysql    mysql       2048 Jan  2 12:48 bin
-rwxr-xr-x   1 mysql    mysql        801 Jan  2 12:48 configure
-rw-r--r--   1 mysql    mysql      19071 Jan  2 12:48 COPYING
drwxr-x---   4 mysql    mysql        512 Jan  2 12:48 data
```

```
drwxr-xr-x   2 mysql    mysql        512 Jan  2 12:48 docs
-rw-r--r--   1 mysql    mysql       5139 Jan  2 12:48 EXCEPTIONS-CLIENT
drwxr-xr-x   3 mysql    mysql       1536 Jan  2 12:48 include
-rw-r--r--   1 mysql    mysql       8528 Jan  2 12:48 INSTALL-BINARY
drwxr-xr-x   2 mysql    mysql        512 Jan  2 12:48 lib
drwxr-xr-x   4 mysql    mysql        512 Jan  2 12:48 man
drwxr-xr-x   9 mysql    mysql        512 Jan  2 12:48 mysql-test
-rw-r--r--   1 mysql    mysql       1410 Jan  2 12:48 README
drwxr-xr-x   2 mysql    mysql        512 Jan  2 12:48 scripts
drwxr-xr-x   3 mysql    mysql        512 Jan  2 12:48 share
drwxr-xr-x   5 mysql    mysql       1024 Jan  2 12:48 sql-bench
drwxr-xr-x   2 mysql    mysql        512 Jan  2 12:48 support-files
drwxr-xr-x   2 mysql    mysql        512 Jan  2 12:48 tests

# scripts/mysql_install_db --user=mysql
Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:
./bin/mysqladmin -u root password 'new-password'
./bin/mysqladmin -u root -h HostName-01 password 'new-password'
See the manual for more instructions.
You can start the MySQL daemon with:
cd . ; ./bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd mysql-test ; perl mysql-test-run.pl

Please report any problems with the ./bin/mysqlbug script!

The latest information about MySQL is available on the web at
http://www.mysql.com
Support MySQL by buying support/licenses at http://shop.mysql.com
#

# chown -R root .

# chown -R mysql data

# bin/mysqld_safe --user=mysql &
5994
Starting mysqld daemon with databases from /opt/MySQL/data
```

2 Stop the MySQL server.

```
# cd /opt/MySQL

# ./bin/mysqladmin -u root -p shutdown
Enter password: <"password">
STOPPING server from pid file /opt/MySQL/data/HostName-01.pid
080104 09:39:21  mysqld ended

[1]+  Done                    ./bin/mysqld_safe
#
```

▼ To Configure MySQL

1 Set the password for the root user in MySQL.

```
# ./bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.0.45-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('password');
Query OK, 0 rows affected (0.00 sec)

mysql>  SET PASSWORD FOR 'root'@'HostName-01' = PASSWORD('password');
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
#
```

2 Set environment parameters for the MySQL script.

Change the file `/opt/MySQL/support-files/mysql.server`:

```
basedir=/opt/MySQL

datadir=/opt/MySQL/data

...

basedir=/opt/MySQL

bindir=/opt/MySQL/bin
# cp /opt/MySQL/support-files/mysql.server /etc/sfw/mysql
```

3 **Start the MySQL server.**

```
# cd /opt/MySQL

# ./bin/mysqld_safe --user=mysql --log&
[1] 7764
# Starting mysqld daemon with databases from /opt/MySQL/data

#
```

Installing Identity Manager on Application Server

To install Identity Manager on Application Server, follow these steps:

1. Install the Application Server
2. Install Identity Manager on Application Server
3. Create Identity Manager Tables in MySQL
4. Configure the Application Server Data Source to Work with Identity Manager
5. Configure Identity Server to Work with Application Server
6. Configure Application Server to Work with Identity Manager
7. Create an OpenSSO Enterprise Realm Administrator
8. Create an OpenSSO Enterprise Realm Resource Object

▼ To Install the Application Server

1 **Follow the installations instructions in the Application Server product documentation.**

See the Application Server product documentation`http://docs.sun.com/coll/1343.4`.

2 **Start the Application Server.**

```
# /opt/SUNWappserver91/bin/asadmin start-domain domain1
```

▼ To Install Identity Manager on the Application Server

The `idm.war` file is used because you will make manual changes to the deployed WAR in a subsequent procedure.

1 **Follow the installation instructions (with one exception) in the Identity Manager Installation Guide for deploying the`idm.war` file on the Application Server. This is the exception:**

Do not recreate the file suggested in the Identity Manager Installation Guide. Use the `idm.war` file that is available in the downloaded zip distribution.

See the Sun Identity Manager 8.0 Installation Guide at `http://docs.sun.com/app/docs/coll/1514.5`.

2 **Remove the following file:**

```
/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm/WEB-INF/lib/j2ee.ja
```

This file causes conflicts with the `j2ee.jar` file that ships with Application Server.

3 **Set the Application Server classpath.**

a. **Log in to the Application Server console.**

b. **In the left frame, click Application Server.**

c. **In the right frame, navigate to the "JVM Settings | Path Settings" tab.**

d. **Add the following entries to the Server Classpath in this exact order:**

```
/opt/SUNWappserver91/lib/appserv-admin.jar
/opt/SUNWappserver91/lib/appserv-rt.jar
/opt/SUNWappserver91/imq/lib/imq.jar
/opt/SUNWappserver91/lib/j2ee.jar
/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm/
  WEB-INF/lib/mysql-connector-java-5.0.8-bin.jar
```

The `mysql-connector-java-5.0.8-bin.jar` will not be available at this file location at this time. The JAR will be added to that directory later. See "To Configure the Application Server Data Source to Work with Identity Manager" on page 26.

e. **Click Save.**

4 **Set the Application Server JVM options.**

In the right frame of the Application Server console, navigate to the "JVM Settings | JVM Options" tab.

To add or modify the following JVM options, click the Add JVM Option button.

a. **Increase the JVM heap size to -Xmx1024M.**

b. **Set the Identity Manager home location to:**

```
-Dwaveset.home=/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm
```

c. **Add the following option to ensure you can create resources in Identity Manager.**

```
-Dcom.sun.enterprise.server.ss.ASQuickStartup=false
```

d. **Click Save.**

5 **Stop the Application Server.**

```
# /opt/SUNWappserver91/bin/asadmin stop-domain domain1
```

▼ To Create Identity Manager Tables in MySQL

1. **Run the following commands:**

```
# cd /opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm/sample
# /opt/mysql/bin/mysql -uroot -ppassword < create_waveset_tables.mysql
```

2. **Verify that the Waveset database was successfully created.**

```
-$ /opt/mysql/bin/mysqlshow  -uroot -ppassword
+--------------------+
|     Databases      |
+--------------------+
| information_schema |
| mysql              |
| test               |
| waveset            |
+--------------------+
-$
```

You should see the `waveset` database name in the output above.

▼ To Configure the Application Server Data Source to Work with Identity Manager

1. **Download the MySQL Connector/J 5.0.**

2. **Extract the archive** `mysql-connector-java-5.0.8.tar.gz`**.**

3. **Copy** `mysql-connector-java-5.0.8-bin.jar` **from the above download to** `/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm/WEB_INF/lib/`

4. **Set the password for the Waveset user in MySQL.**

```
# cd /opt/mysql

# ./bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.0.45-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SET PASSWORD FOR 'waveset'@'localhost' = PASSWORD('password');
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> exit
Bye
#
```

5 Start the Application Server.

```
# /opt/SUNWappserver91/bin/asadmin start-domain domain1
```

6 Connect to the data source.

```
# cd /opt/SUNWappserver91/domains/domain1/applications/
  j2ee-modules/idm/bin

# chmod +x lh

# export WSHOME=/opt/SUNWappserver91/domains/domain1/applications/
  j2ee-modules/idm

# export CLASSPATH=/opt/SUNWappserver91/lib/appserv-rt.jar:
  /opt/SUNWappserver91/lib/javaee.jar:$CLASSPATH

# ./lh setRepo -v -tMySQL   -ujdbc:mysql://localhost/waveset -Uwaveset -Ppassword
Defaulting administrator to 'configurator'.
Defaulting credentials to 'configurator'.
DB Server @ jdbc:hsqldb:hsql://127.0.0.1:53878/idm
Defaulting jdbcDriver to 'org.gjt.mm.mysql.Driver'.
Checking 'MysqlDataStore:jdbc:mysql://localhost/waveset'...
Switching to 'MysqlDataStore:jdbc:mysql://localhost/waveset'...
Getting current location....
Current Location is 'MysqlDataStore:jdbc:mysql://localhost/waveset'
userid is 'waveset'
password is '(set)'
jdbcDriver is 'org.gjt.mm.mysql.Driver'
#
```

▼ To Configure Identity Manager to Work with Application Server

1 Set the environment variables that will be required for the setup program:

```
# export WSHOME=/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm

# export JAVA_HOME=/usr/java

# export PATH=/usr/java/bin:$PATH
```

2. **Start an X server on your local machine, and set the DISPLAY variable on the Application Server host computer.**

3. **Run the following commands:**

```
# cd /opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm/bin

# ./lh setup
```

4. **Select MySQL (JDBC Driver) as the Repository Type.**

5. **Enter the same password for the `waveset` user that you set earlier in MySQL.**

6. **Click the Next button.**

7. **Accept the default setting to setup a demo environment.**

8. **Enter information about the demo user.**

   In this case, enter following credentials:

   User Name: demoapprover

   Password: password

9. **In the next screen, select the option for a Notification File for the Mail Settings.**

   You may accept the default file or customize it.

10. **In the next screen, click Execute.**

    The `lh` program logs the details of the execution steps in the screen. See the Example in the "Sample Output" on page 123at the end of this chapter.

    Click Done.

11. **Change permissions so that Identity Manager can perform certain actions.**

    Add the following lines to `/opt/SUNWappserver91/domains/domain1/config/server.policy`:

```
grant {
        permission java.lang.RuntimePermission "accessClassInPackage.sun.io";
        permission java.lang.RuntimePermission "getClassLoader";
        permission java.lang.RuntimePermission "createClassLoader";
        permission java.lang.RuntimePermission "accessDeclaredMembers";
        permission com.waveset.repository.test.testConcurrentLocking "read";
        permission java.net.SocketPermission "*", "connect,resolve";
        permission java.io.FilePermission "*", "read";
        permission java.util.PropertyPermission "*", "read,write";
};
```

```
grant codeBase "file:${waveset.home}/-" {
        permission java.util.PropertyPermission "waveset.home", "read,write";
        permission java.util.PropertyPermission "security.provider", "read,write";
        permission java.io.FilePermission "${waveset.home}${/} *",
            "read,write,execute";
        permission java.io.FilePermission "${waveset.home}/help/index/-",
            "read,write,execute,delete";
        permission java.io.FilePermission "$(java.io.tmpdir)$(/)*",
            "read,write,delete";
        permission java.util.PropertyPermission "*", "read,write";
        permission java.lang.RuntimePermission "accessClassInPackage.sun.io";
        permission java.net.SocketPermission "*", "connect,resolve";
};
```

12 To enable Identity Manager to connect to OpenSSO Enterprise with the `SunAccessManagerRealmResourceAdapter`, add the two following policies:

```
grant {
        permission java.lang.RuntimePermission "shutdownHooks";
        permission java.io.FilePermission "${waveset.home}/WEB-INF/spe/config/spe.tld", "read";
};
```

13 Restart the Application Server.

```
# /opt/SUNWappserver91/bin/asadmin stop-domain domain1

# /opt/SUNWappserver91/bin/asadmin start-domain domain1
```

Watch for any errors in the Application Server `server.log` file.

14 Verify that you can successfully log in to Identity Manager.

Go to the Identity Manager console at `http://`*ApplicationServerHost:Port*`/idm/login.jsp`

a. Log in using the following credentials:

Username: configurator

Password: configurator

To minimize security risk, it is a good practice to change the default password for this administrator.

b. Log out.

c. Log in using the following credentials:

Username: administrator

Password: administrator

d. Log out.

e. Log in using the following credentials:

Username: demoapprover

Password: password

f. Log out.

▼ To Configure Application Server to Work with Identity Manager

Before You Begin In the following steps, you configure the `AMConfig.properties` you generate in the first step. Use the credentials of the `amadmin` user to connect with the OpenSSO Enterprise server. You could use a user other than `amadmin` as long as the user has privileges to read the OpenSSO Enterprise configuration data. This should not be a security concern because the `AMConfig.properties` file is required only to perform the initial configuration and to test the Access Manager Realm Resource adapter instance. The `AMConfig.properties` file is not needed after the Policy Agent has been installed on the Identity Manager server, and the file can be deleted afterward.

1 Generate the OpenSSO Enterprise client configuration file.

Go to the directory, where you extracted the OpenSSO Enterprise zip distribution, and unzip the *opensso*`/samples/opensso-client.zip archive` in a temporary directory. Then run the following commands:

```
# cd opensso/samples/tmp/sdk

# chmod +x scripts/compile-samples.sh

# scripts/compile-samples.sh

# chmod +x scripts/setup.sh

# scripts/setup.sh
Debug directory (make sure this directory exists):
  /opt/SUNWappserver91/domains/idm/logs/opensso_debug
Application user (e.g. URLAccessAgent)passord: password
Protocol of the server: http
Host name of the server: host1.example.com
Port of the server: 8280
Server's deployment URI: /opensso
Naming URL (hit enter to accept default value,
  http://host1.example.com:8280//opensso/namingservice):
```

```
  http://host1.example.com8280/opensso/namingservice
#
```

You should now see a `AMConfig.properties` file created in the `sdk/resources` directory.

2 Install the OpenSSO Enterprise command-line tools.

They are present in the OpenSSO Enterprise zip distribution, in the *opensso*`/tools/ssoAdminTools.zip` archive.

```
# mkdir /opt/opensso-tools

# cd /opt/opensso-tools

# unzip /export/software/
  FAM_80_B3_QA_Test/opensso_zip/opensso/tools/ssoAdminTools.zip

# export JAVA-HOME=/usr/java
# ./setup
Path to config files of OpenSSO server (example: /openSSO):
/opt/fam80-qatest-server1
Debug Directory: /opt/opensso-tools/debug
Log Directory: /opt/opensso-tools/log
The scripts are properly setup under directory:
/opt/opensso-tools/opensso
Debug directory is /opt/opensso-tools/debug.
Log directory is /opt/opensso-tools/logs.
The version of this tools.zip is: Express build 5b(2008-September-22 07:55)
The version of your server instance is: Express build 5b(2008-September-22 07:55)
#
```

You will now see an `opensso` directory (or a directory with the name of the context-root of your OpenSSO Enterprise deployment), in the `/opt/opensso-tools` directory.

3 Encrypt the password for the `amadmin` user using the `ampassword` utility.

First, you need to create a text file containing the password of the `amadmin` user in plain text. In the following example, the password file `/export/software/amadmin_pwd` is created:

```
# cd /opt/opensso-tools/opensso/bin

# ./ampassword --encrypt /export/software/amadmin_pwd
AQICSw+UrU2DJyY1KBeoC0iuzv3gQTGkbI39
#
```

4 Customize the `AMConfig.properties` file that was created in step 1.

a. In the OpenSSO Enterprise console, navigate to Configuration > Servers and Sites > *server-entry* > Security.

b. Copy the value from the property Password Encryption Key, and use the value to modify the following property:

```
am.encryption.pwd=AQICrPmBjI5aThg1H6kKcJr0/Lu4D9LdTlqe
```

c. Modify the following property as shown:

```
com.sun.identity.agents.app.username=amadmin
```

d. For security purposes, either comment out the following line, or leave the value empty:

```
#com.iplanet.am.service.password=
```

e. Modify the following property using the value from the encrypted password generated in step 3 above:

```
com.iplanet.am.service.secret=AQICSw+UrU2DJyY1KBeoC0iuzv3gQTGkbI39
```

5 Copy the OpenSSO Enterprise Client files to the Identity Manager application directory. You will need the following files:

- The `openssoclientsdk.jar` library that is present in the `/sdk/lib` directory from the `fam-client.zip` archive in the OpenSSO Enterprise zip distribution.

```
# cp /export/software/
  FAM_80_IDM_80_Integration/fam_zip/opensso/samples/
tmp/sdk/lib/openssoclientsdk.jar  /opt/SUNWappserver91/domains/domain1/
  applications/j2ee-modules/idm/WEB-INF/lib/
```

- The `AMConfig.properties` generated above.

```
# mkdir /opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/
idm/WEB-INF/classes
# cp /export/software/FAM_80_IDM_80_Integration/fam_zip/opensso/samples/
tmp/sdk/resources/AMConfig.properties /opt/SUNWappserver91/domains/domain1/
applications/j2ee-modules/
idm/WEB-INF/classes
```

6 Update the Application Server classpath.

a. Login to the Application Server Console.

b. Navigate to Application Server | JVM Settings | Path Settings

c. Update the Classpath Suffix to contain the following entries:

```
/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm/
   WEB-INF/lib/openssoclientsdk.jar

/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm/WEB-INF/classes
```

d. Click Save to save your changes.

e. Log out from the Application Server Console.

7 Restart the Application Server.

```
# /opt/SUNWappserver91/bin/asadmin stop-domain domain1

# /opt/SUNWappserver91/bin/asadmin start-domain domain1
```

Watch for any errors in the Application Server `server.log` `log` file.

Creating an OpenSSO Enterprise Realm Administrator

If you plan to use Identity Manager to manage objects in the OpenSSO Enterprise top-level realm, then create a user in the OpenSSO Enterprise root realm. Give this user the same privileges as the Top-Level Admin Role. The privileges should allow this user "Read and write access to all realm and policy properties." This user will be used to configure the Identity Manager Resource adapter.

If you plan to use Identity Manager to manage objects in the OpenSSO Enterprise sub-realm, then create a user in the OpenSSO Enterprise sub-realm. Give this user privileges to "Read and write access to all realm and policy properties." This user will have the privileges of a sub-realm administrator, and will be used to configure the Identity Manager Resource adapter. In this example, a realm administrator `sradmin` with the password `password` was created in the sub-realm `(Top-Level Realm) > idm`.

▼ To Create an OpenSSO Enterprise Realm Resource Object

1 Access the Identity Manager console.

In this example, go to `http://`*ApplicationServerHost*:*Port*`/idm/login.jsp`. The Identity Manager login page is displayed.

2 Log in using the following credentials:

User Name: `configurator`

Password: `configurator`

3 Add the OpenSSO Enterprise realm adapter to the resource classpath.

a. Navigate to Resources | Configure Types.

b. At the bottom of the page, click "Add Custom Resource."

c. Add the following to the Resource Classpath:

```
com.waveset.adapter.SunAccessManagerRealmResourceAdapter
```

In earlier versions of OpenSSO Enterprise, it was possible to install Access Manager in the legacy mode of operation. In legacy mode, a different Identity Manager resource adapter `com.waveset.adapter.SunAccessManagerResourceAdapter`, should be configured on Identity Manager. Both types of adapters have the same functionality. But `com.waveset.adapter.SunAccessManagerResourceAdapter` uses the legacy Access Manager `AMSDK` API, while the `com.waveset.adapter.SunAccessManagerRealmResourceAdapter`uses the OpenSSO Enterprise `idRepo` API.

d. Click Save.

4 Configure the OpenSSO Enterprise Realm adapter.

a. Navigate to Resources | List Resources

b. Choose --Resource Type Actions-- | New Resource

c. Choose Sun Access Manager Realm from the list of resources. Click New.

d. In the Create Sun Access Manager Realm Resource Wizard screen, click Next.

e. In the Resource Parameters screen, provide the following information:

Host:	Fully-qualified hostname of the OpenSSO Enterprise server. Example: `host1.example.com`
TCP Port:	Port number of the OpenSSO Enterprise server. In this example, 48080.
User:	`sradmin` You must use an OpenSSO Enterprise realm administrator, and not a non-administrator user, because it requires special permissions. If you use a non-administator user, this test will fail. Use the realm administrator configured in the previous section.
Password:	`password` This is the plain-text password of the user realm administrator.
Protocol:	Protocol of the OpenSSO Enterprise server realm or Identity Manager. In this example, enter `http`.

Realm:	This is the realm name of the OpenSSO Enterprise server. In this example, enter `/idm`. If the user entered above were in the top-level realm, you would enter just a slash (`/`).
Encryption Key:	This is the value of the `am.encryption.pwd` property in the `AMConfig.properties` file. You can obtain the value of `am.encryption.pwd` from the OpenSSO console. Navigate to Configuration > Servers and Sites > *server-entry* > Security .
JCE Encryptor Class :	This is the value of the `com.iplanet.security.encryptor` property in the `AMConfig.properties` file. In this example, enter: `com.iplanet.services.util.JCEEncryption`.
Naming Service URL:	This is the value of the `com.iplanet.am.naming.url` property in the `AMConfig.properties` file. In this example, enter `:http://host1.example.com:48080/opensso/namingservice`.
Error Log Level:	`message`
Error Log Directory:	Directory into which the Identity Manager Access Manager Resource will write debug logs. This directory must already exist. In this example, enter:`/opt/SUNWappserver91/domains/domain1/logs/opensso_debug`.

5 Click Test Configuration.

The following message will be displayed: "Test connection succeeded for resource(s): SunAccessManagerRealm." If you don't see this message, then you must troubleshoot by looking at the following logs:

- Application Server `server.log`

  `/opt/SUNWappserver91/domains/domain1/logs/server.log`
- Access Manager client logs at `/opt/SUNWappserver91/domains/domain1/logs/opensso_debug` (specified in the form above)

Click Next.

6 In the Account Attributes page, set the following mapping:

Identity System Attribute: fullname

Resource User Attribute: `cn`

Attribute Type: string

Required: yes

Click Next.

7 **In the Identity Template page, make sure you have this entry:**

```
$accountId$
```

Click Next.

8 **In the Identity System Parameters page, select `uid` for the Display Name Attribute parameter.**

Click Save to save the value.

The Resource List page is displayed. You should see a resource of the type Sun Access Manager Realm. To expand this branch, click the arrow next to it.

a. **Expand the Sun Access Manager Realm type by clicking the arrow next to it.**

You should see an entry `SunAccessManagerRealm`.

b. **Expand the `SunAccessManagerRealm` branch by clicking the arrow next to it.**

You should see a listing of all OpenSSO Enterprise roles and groups under this branch that exist in the OpenSSO Enterprise sub-realm that the Identity Manager Resource was configured with in step 4e above.

Provisioning Identities from Identity Manager to OpenSSO Enterprise

Provisioning users from one data store to a second allows the individual applications (for which the user stores are being provisioned) to use their proprietary repositories to read and write attributes that are specific to them. Before you can provision a user, role, or group into OpenSSO Enterprise from Identity Manager, the `SunAccessManagerRealm` adapter must be configured with the information required to successfully log in to OpenSSO Enterprise and the target OpenSSO Enterprise realm.

When you provision a user, role, or group from Identity Manager into OpenSSO Enterprise, you must select the Resource that you want to provision into. In this example, you will select the `SunAccessManagerRealm` adapter as the Resource to provision into. This adapter uses OpenSSO Enterprise APIs (OpenSSO package `com.sun.identity.idm.*`), to communicate with OpenSSO Enterprise. OpenSSO Enterprise receives the request to create or retrieve a user, role, or group. OpenSSO Enterprise then performs the task on its configured data stores in the

relevant OpenSSO Enterprise realm. Similarly, for role or group retrieval from the OpenSSO Enterprise data store, the `SunAccessManagerRealm` adapter uses OpenSSO Enterprise APIs to communicate with OpenSSO Enterprise and to retrieve these objects.

In Identity Manager, Resource Objects correspond to OpenSSO Enterprise roles and groups. Resource Accounts correspond to OpenSSO Enterprise users that have been provisioned from Identity Manager. Since Resource Objects and Resource Accounts are managed differently in Identity Manager, both types of resources are viewable on separate tabs in the Identity Manager console.

The following figure illustrates how objects are provisioned and retrieved in Identity Manager.

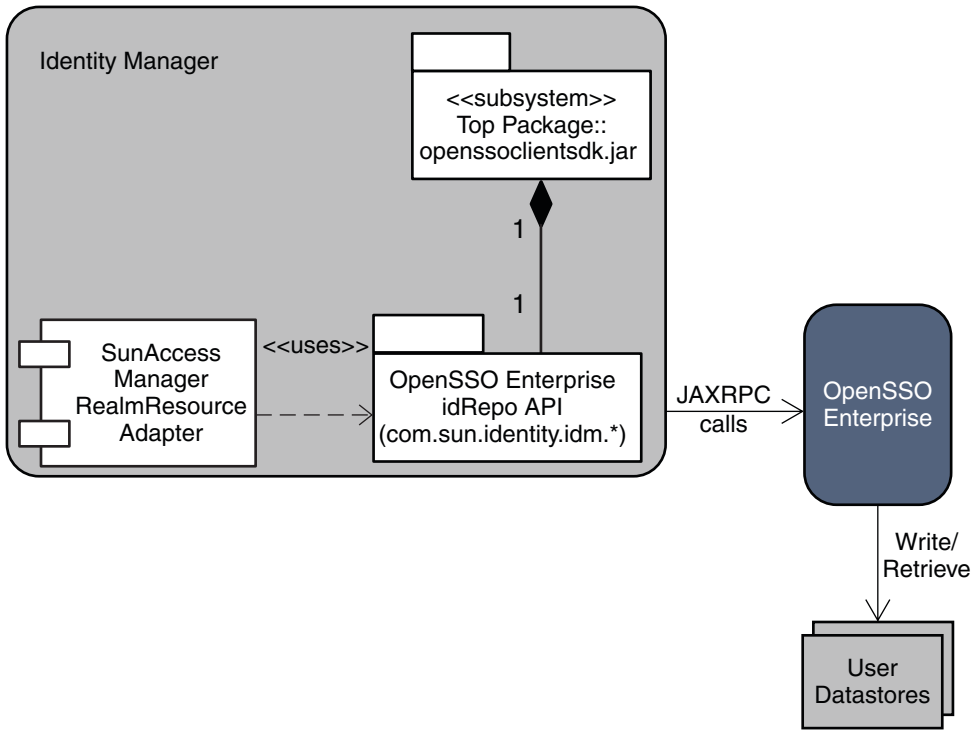


FIGURE 1–2 Overview of Provisioning and Retrieving Objects in Identity Manager

To provision identities from Identity Manager to OpenSSO Enterprise, follow these steps:

1. View OpenSSO Enterprise Roles and Groups in Identity Manager
2. View OpenSSO Enterprise User Accounts in Identity Manager
3. Provision a Test User From Identity Manager Into OpenSSO Enterprise

4. Verify that Identities Were Successfully Provisioned
5. Provision a Test Role From Identity Manager Into OpenSSO Enterprise
6. Verify the Test Role Was Successfully Provisioned from Identity Manager Into OpenSSO Enterprise
7. Provision an Admin-User From Identity Manager Into OpenSSO Enterprise
8. Verify the Admin-User Was Successfully Provisioned from Identity Manager into OpenSSO Enterprise
9. Provision an Admin-Role From Identity Manager Into OpenSSO Enterprise

▼ To View OpenSSO Enterprise Roles and Groups in Identity Manager

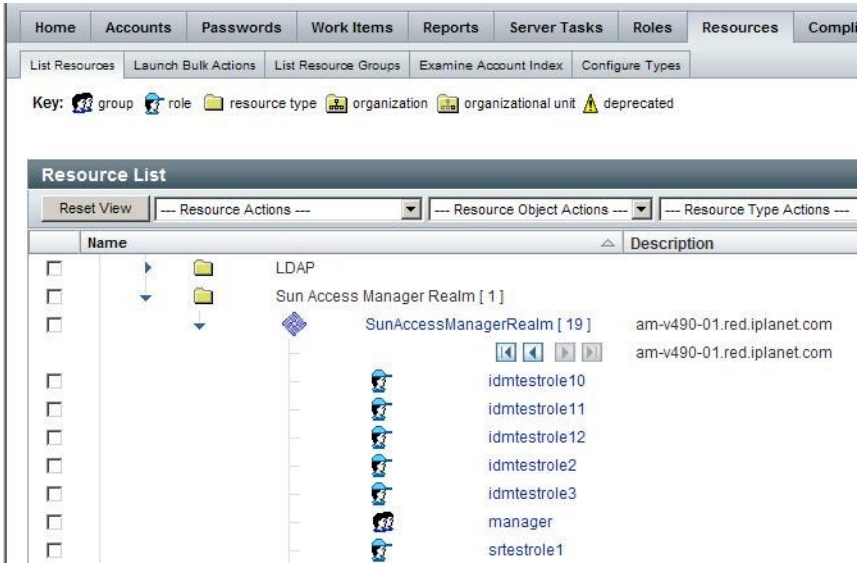
1 **1) Login to the Identity Manager console using the following credentials:**

User Name: configurator

Password: configurator

2 **Navigate to the tab Resources | List Resources.**

3 **Expand the branch for the `SunAccessManagerRealm` adapter instance.**



▼ To View OpenSSO Enterprise User Accounts in Identity Manager

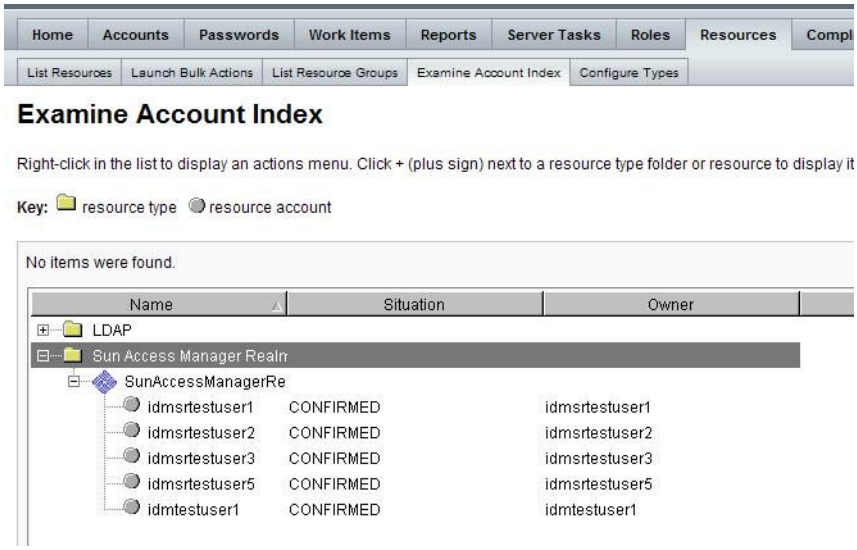
1 **Log in to the Identity Manager console using the following credentials:**

User Name: configurator

Password: configurator

2 **Navigate to the tab Resources | Examine Account Index.**

3 **Expand the branch for the `SunAccessManagerRealm` adapter instance.**



You can also view the provisioned OpenSSO Enterprise user accounts in the Identity Manager console by navigating to the tab Accounts | List Accounts. However, that page will show you all Identity Manager accounts in the Identity Manager server, including those provisioned into OpenSSO Enterprise and any other resource or system, that has been configured in Identity Manager. For example, if Identity Manager were configured for SAP and OpenSSO Enterprise Resources, you would see a listing of users that have been provisioned into both systems. You can also use this page to create or provision users as described in sections below.

To view the accounts created per Resource Type, navigate to the tab Resources | Examine Account Index. This page is for viewing only, and you cannot use this page to create or provision a user.

▼ To Provision a Test User From Identity Manager Into OpenSSO Enterprise

Follow these steps to test the Access Manager Realm Resource that was configured in Identity Manager.

1. **Log in to the Identity Manager console using the following credentials:**

   User Name: configurator

   Password: configurator

2. **Navigate to the tab Accounts | List Accounts.**

3. **Select the option New Actions | New User.**

4. **In the Create User page, enter these values:**

AccountID:	`idmuser`
First Name	Identity Manager
Last Name:	User
Password:	password
Confirm Password password:	password

5. **In the Create User page, click the Resources tab.**

   Select the `SunAccessManagerRealmResourceAdapter` resource as the Current Resource.

6. **Click Save at the bottom of the screen.**

7. **In the next screen, you should see a success message "Account idmuser created." Click OK.**

   The User List page is displayed and contains a list with the newly-created user `idmuser`.

8. **Log out from the Identity Manager console.**

▼ To Verify that Identities Were Successfully Provisioned

1. **In the Identity Manager console, return to the Accounts tab | List Accounts tab. .**

   You should see the `idmuser` user entry in the listing

2. **Log in to the OpenSSO Enterprise console and verify that the user account is visible from the Subjects tab of your realm.**

   If the end-user entry is visible in both Identity Manager and OpenSSO Enterprise, then the end-user has successfully been provisioned.

3. **Log in to Identity Manager and verify that you are logged into the Identity Manager User Page.**

   In this example, go to the following URL:

   `http://`*ApplicationServerHost:Port*`/idm/user`

   Log in using the following credentials:

   User Name: `idmuser`

   Password: password

4. **Log in to OpenSSO Enterprise.**

   In this example, go to the following URL:

   `http://host1.example.com:48080/amserver/UI/Login?realm=idm`

   Log in using the following credentials:

   User Name: `idmuser`

   Password: password

5. **Verify that you are logged into OpenSSO Enterprise and that you can see the user profile page.**

6. **Log out from the Identity Manager administrator interface and OpenSSO Enterprise console.**

Next Steps If you are not able to log in as the user, do the following:

- Verify that you can see the user entry in the sub-realm in the OpenSSO Enterprise console.
- Troubleshoot the issue using the OpenSSO Enterprise debug logs and the debug logs written by the Identity Manager's Access Manager Resource

▼ To Provision a Test Role From Identity Manager Into OpenSSO Enterprise

Before You Begin In the OpenSSO Enterprise console, in the data store configuration page for the realm or sub-realm into which you will be provisioning the role, for the property LDAP Roles Attributes, add `cn` to the list of values.

The `cn` attribute is not defined as an attribute for the `IdType.ROLE` in the Data Store configuration by default. This attribute is set, when the role is provisioned to OpenSSO

Enterprise. If the cn attribute is not already defined, Identity Manager shows the following error on the Identity Manager administrator interface:

```
com.waveset.util.WavesetException:
Error creating object 'idmsrtestrole5'.
com.waveset.util.WavesetException:
Error setting attributes for 'idmsrtestrole5'
com.sun.identity.idm.IdRepoException:
Illegal arguments: One or more required arguments is null or empty
```

1. **Log in to the Identity Manager administrator interface using the following credentials:**

   User Name: configurator

   Password: password

2. **Navigate to the tab Resources | List Resources.**

3. **Expand the branch for the Sun Access Manager Realm entry.**

4. **Mark the checkbox in front of the `SunAccessManagerRealm` entry.**

5. **Choose the option Resource Actions | Create Resource Object.**

6. **In the New Resource Object page, select Role from the dropdown box, and click New.**

   In the next page:

   a. **Enter the name of the role as `idm_users`.**

   b. **Assign the user `idmuser` to this role.**

7. **Click Save.**

8. **In the Create Role Results page, click OK.**

   The Resource List page is displayed, and contains a list with the role `idm_users` created when you expand the SunAccessManagerRealm branch.

9. **Log out from the Identity Manager administrator interface.**

▼ To Verify the Test User Role Was Successfully Provisioned from Identity Manager Into OpenSSO Enterprise

1 **Log in to OpenSSO Enterprise.**

In this example, go to the following URL:

`http://host1.example.com:48080/opensso`

Log in using the following credentials:

User Name: amadmin

Password: password

2 **Navigate to the sub-realm `idm` and tab Subjects | User.**

The user Identity Manager User should be listed as one of the users.

3 **Navigate to the tab Subjects | Role.**

The role `idm_users` should be listed as one of the roles.

4 **In the role profile page, click the `idm_admins_role` role entry.**

5 **Click on the User tab.**

The user Identity Manager User should be selected into the role.

6 **Log out of the OpenSSO Enterprise console.**

Next Steps If you are not able to see the role entry or the user assigned to the role troubleshoot the issue using the OpenSSO Enterprise debug logs and the debug logs written by the Identity Manager's Access Manager Resource.

▼ To Provision an Admin-User From Identity Manager Into OpenSSO Enterprise

At this point, the Identity Manager is not yet protected by the policy agent. Follow these steps to create a user that will have administrative privileges on Identity Manager.

1 **Login to the Identity Manager administrator interface using the following credentials:**

User Name: configurator

Password: configurator

2 **Navigate to the tab Accounts | List Accounts.**

3 **Choose the option New Actions | New User.**

4 **In the Create User page, enter these values:**

AccountID:	idmadmin
First Name	Identity Manager
Last Name:	Admin
Password:	password
Confirm Password :	password

5 **In the Create User page, click the Resources tab.**

For the Individual Resource Assignment, choose the `SunAccessManagerRealm` as the Current Resource.

6 **In the Create User page, click the Security tab.**

a. **For the Capabilities property, select all capabilities as Assigned Capabilities.**

b. **For the Controlled Organizations property, choose top as the Selected Organizations.**

7 **Click Save at the bottom of the screen.**

8 **In the next screen, you should see a success message, "Account idmadmin created."**

Click OK.

The User List page is displayed and contains a list with the new user `idadmin`.

9 **Log out from the Identity Manager administrator interface.**

▼ To Verify the Admin-User Was Successfully Provisioned from Identity Manager into OpenSSO Enterprise

1 **In the Identity Manager administrator interface, return to the Accounts tab | List Accounts tab**

You should see the admin-user entry in the listing.

2 **Log in to the OpenSSO Enterprise console.**

Verify that the admin-user account is visible from the Subjects tab of your realm. If the admin-user entry is visible in both Identity Manager and OpenSSO Enterprise, then the admin-user has successfully been provisioned.

3 **Log in to Identity Manager.**

In this example, go to the following URL:

```
http://host1.example.com:2080/idm
```

Log in using the following credentials:

User Name: idmadmin

Password: password

Verify that you are logged into the Identity Manager administrator interface.

4 **Log in to OpenSSO Enterprise.**

In this example, go to the following URL:

```
http://host1.example.com:48080/opensso/UI/Login?realm=idm
```

Log in using the following credentials:

User Name: idmadmin

Password: password

Verify that you are logged into OpenSSO Enterprise and can see the user profile page.

▼ To Provision an Admin-Role From Identity Manager Into OpenSSO Enterprise

At this point, the Identity Manager is not yet protected by the policy agent. The role that will be created here will not have any special privileges assigned to it. It will only be used to group the administrative users, and this role will be used later in a policy in OpenSSO Enterprise.

1 **Log in to the Identity Manager administrator interface as using the following credentials:**

User Name: configurator

Password: configurator

2 **Navigate to tab Resources | List Resources.**

3 **Expand the branch for the Sun Access Manager Realm entry.**

4. **Mark the checkbox in front of the `SunAccessManagerRealm` entry.**
5. **Choose the option Resource Actions | Create Resource Object.**
6. **In the New Resource Object page, select Role from the dropdown box, and click New.**
7. **In the next page, enter the name of the role as `idm_admins`, and assign the user `idmadmin` to this role.**

   Click Save.
8. **In the Create Role Results screen, click OK.**

   When you expand the `SunAccessManagerRealm` branch, the Resource List page is displayed and contains a list with the new role `idm_admins`.
9. **Log out of the Identity Manager administrator interface.**

▼ To Verify the Test Admin Role Was Successfully Provisioned from Identity Manager Into OpenSSO Enterprise

1. **Log in to OpenSSO Enterprise.**

   In this example, go to the following URL:

   `http://host1.example.com:48080/opensso`

   Log in using the following credentials:

   User Name: idm_admins

   Password: password
2. **Navigate to the sub-realm `idm` and tab Subjects | User.**

   The user Identity Manager Admin should be listed as one of the users.
3. **Navigate to the tab Subjects | Role.**

   The role `idm_admins` should be listed as one of the roles.
4. **In the role profile page, click the `idm_admins` role entry.**
5. **Click on the User tab.**

   The user Identity Manager Admin should be selected into the role.

6 Log out of the OpenSSO Enterprise console.

Next Steps If you are not able to see the role entry or the user assigned to the role, troubleshoot the issue using the OpenSSO Enterprise debug logs and the debug logs written by the Identity Manager Access Manager Resource.

Installing And Configuring the OpenSSO Enterprise Policy Agent on Identity Manager

Although this document describes an example where Identity Manager and OpenSSO Enterprise are configured for both single sign-on and provisioning, it is possible to configure a deployment for single sign-on without provisioning, or for provisioning without single sign-on. If single sign-on between OpenSSO Enterprise and Identity Manager is not required, then the OpenSSO Enterprise Policy Agent does not need to be installed or configured. In that case, you can ignore the steps that involve the OpenSSO Enterprise Policy Agent.

To install and configure the OpenSSO Enterprise policy agent on Identity Manager, follow these steps:

1. Create the OpenSSO Enterprise Agent Profile On The OpenSSO Enterprise Server.
2. Install the OpenSSO Enterprise Policy Agent on the Identity Manager Server.
3. Configure the OpenSSO Enterprise Policy Agent on OpenSSO Enterprise .
4. Create Policies on OpenSSO Enterprise.
5. Disable Protection of Identity Manager Server by the OpenSSO Enterprise Policy Agent .
6. Configure The OpenSSO Enterprise Policy Agent On Identity Manager Server.

▼ To Create the OpenSSO Enterprise Agent Profile On The OpenSSO Enterprise Server

1 Download Policy Agent 3.0 for Sun Application Server 9.1.

2 Log in to the OpenSSO Enterprise console.

3 Navigate to Access Control | / (Top-Level Realm) | Agents | J2EE.

4 In the Agent section, New and create a new agent profile with these values:

Name: idmagent

Password:	password
Re-Enter Password:	password
Server URL:	`http://host1.example.com:48080/opensso`
Agent URL:	`http://host1.example.com:2080/agentapp`

Click Create.

The console displays the J2EE Policy Agent page again with a hyperlink for the agent profile `idmagent`.

5 **Click on the `idmagent` hyperlink.**

The "Edit idmagent" page is displayed. The agent profile is now created.

6 **If OpenSSO Enterprise is deployed on a web server, in the Agent profile page, navigate to the tab SSO.**

Select the property SSO Decode (`com.sun.identity.agents.config.sso.decode`).

It is necessary to select this property only when OpenSSO Enterprise is deployed on a web server. If you leave this property unselected, then you will find that, after you login to OpenSSO Enterprise, the browser appears to be stuck and hanging on the OpenSSO Enterprise login screen.

Click Save.

7 **Log out of the OpenSSO Enterprise console.**

8 **Verify that you can login to the OpenSSO Enterprise console as this user.**

9 **Create an policy agent password file named `/export/software/agent_pwd`.**

This file should contain only the password for the Agent profile, in plain text

▼ To Install the OpenSSO Enterprise Policy Agent on the Identity Manager Server

The Policy Agent provides these capabilities:

- Retrieve and map an OpenSSO Enterprise user session attribute (`UserToken`), to an Identity Manager attribute (`sois_user`), so that Identity Manager can perform the single sign-on from OpenSSO Enterprise.
- Access protection for the Identity Manager pages in addition to the protection offered by the specific capabilities that can be explicitly assigned to a user from the Identity Manager administrator interface.

The `sois_user` is the authentication property in Identity Manager that is used during single sign-on between OpenSSO Enterprise and Identity Manager. The name `sois_user` given to the property was an abbreviation for Sun ONE Identity Server User. The Sun ONE Identity Server product was a predecessor to OpenSSO Enterprise.

1. **Follow instructions in the policy agent documentation for installing the Policy Agent on Application Server.**

2. **Deploy the `agentapp.war` on the Sun Application Server.**

3. **When the policy agent installation is complete, verify that the agent is installed and functioning properly.**

   Install the sample application `agentsample` that is ships with the agent and test the application. Instructions to install and test the sample application are available on the OpenSSO website.

Caution – Before you deploy and test the `agentsample` application, you must remove the following entries in the GlassFish JVM path: :

```
/opt/SUNWappserver91/domains/idm/applications/j2ee-modules/idm/WEB-INF/lib/openssoclie
/opt/SUNWappserver91/domains/idm/applications/j2ee-modules/idm/WEB-INF/classes
```

These entries were added in the procedure "To Configure Application Server to Work with Identity Manager" on page 30.

If you do not remove these entries before deploying the `agentsample` application, you will get a 500 error on the browser when you try to access the `agentsample` application.

▼ To Configure the OpenSSO Enterprise Policy Agent on OpenSSO Enterprise

1. **Configure the OpenSSO Enterprise Agent Profile**

   a. **Log in to the OpenSSO Enterprise console as `amadmin`.**

   b. **Navigate to Access Control | /(Top-Level Real) | Agents | J2EE.**

   c. **Click the policy agent profile that was created earlier and was associated with the agent installation.**

   d. **Navigate to the tab OpenSSO Services.**

e. **For the property OpenSSO Enterprise Login URL (`com.sun.identity.agents.config.login.url`), remove the existing entry, and add this entry:**

```
[0]=http://host1.example.com:48080/opensso/UI/Login?realm=idm
```

The value must be the login URL that the AM users should use to login to AM

Click Save.

2 **Navigate to the tab Application.**

a. **For the property Session Attribute Fetch Mode (`com.sun.identity.agents.config.session.attribute.fetch.mode`), choose the option `HTTP_HEADER`.**

b. **For the property Session Attribute Mapping (`com.sun.identity.agents.config.session.attribute.mapping`), remove the existing entry, and add this entry:**

```
[UserToken]=sois_user
```

c. **For the property Not Enforced URIs (`com.sun.identity.agents.config.notenforced.uri`), add these entries:**

```
/idm/styles/*

/idm/includes/*

/idm/images/*
```

Click Save.

3 **Log out from the OpenSSO Enterprise console.**

▼ To Create Policies on OpenSSO Enterprise

For detailed information on creating policies on OpenSSO Enterprise, see "Creating Policies and Referrals" in *Sun OpenSSO Enterprise 8.0 Administration Guide*.

Create the following roles in the realm where the users will be provisioned. If the policy is to be created in a sub-realm, then you must first create a Referral Policy in the top-level realm for the same URLs.

1. **Identity Manager User Policy**

   This policy restricts access to the Identity Manager user pages, only to the users in the `idm_users` role. So regular Identity Manager users will not be allowed to access the Identity Manager administrator interface URIs.

   a. **URL Policy**

      For http://server:port/idm/user, allow GET and POST actions .

   b. **URL Policy**

      For http://server:port/idm/user/*, allow GET and POST actions .

   c. **URL Policy**

      For http://server:port/idm/user/*?*, allow GET and POST actions.

   Subject Type: OpensSSO Identity Subject | Role | idm_users

2. **Identity Manager Admin Policy**

   This policy restricts access to the Identity Manager pages, to only the users in the `idm_admins` role. The users in this role will be able to access all Identity Manager pages, both administrator and user pages.

   a. **URL Policy**

      For `http://server:port/idm`, allow GET and POST actions

   b. **URL Policy**

      For `http://server:port/idm/*`, allow GET and POST actions

   c. **URL Policy**

      For http://server:port/idm/*?*, allow GET and POST actions.

   Subject Type: OpenSSO Identity Subject | Role | idm_admins

▼ To Disable Protection of Identity Manager Server by the OpenSSO Enterprise Policy Agent

This task enables you to perform the tasks described in the sections below without the policy agent getting in the way. At this point, the policies haven't been set up on OpenSSO Enterprise. You would be denied access to all Identity Manager URLs until policies are set up. The protection by the policy agent will be re-enabled in a subsequent procedure. See "To Re-Enable Identity Manager Protection by the OpenSSO Enterprise Policy Agent " on page 60.

1. **Log in to the OpenSSO Enterprise console using the following credentials:**

   User Name: amadmin

   Password: password

2. **Navigate to Access Control | /(Top-Level Realm) | Agents | J2EE | idmagent | Application.**

3. **For the property Not Enforced URI (`com.sun.identity.agents.config.notenforced.uri`), add this entry:**

   ```
   /idm/*
   /idm/*?*
   ```

4. **Click Save.**

5. **Log out of the OpenSSO Enterprise console.**

▼ To Configure the OpenSSO Enterprise Policy Agent On Identity Manager Server

1. **Modify the Identity Manager application descriptor.**

   a. **Go to the directory where the application descriptor is present.**

      ```
      # cd /opt/SUNWappserver91/domains/
      domain1/applications/j2ee-modules/idm/WEB-INF
      ```

   b. **Back up the file `web.xml`.**

   c. **Edit `web.xml`.**

      - Change DOCTYPE as follows:

        ```
        <web-app version="2.4"
        xmlns="http://java.sun.com/xml/ns/j2ee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
          http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
        ```

      - Delete the single instance of `<web-app>` in the next line.
      - Add the following just before the first `<filter>` definition:

        ```
        <filter>
          <filter-name>Agent</filter-name>
          <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
        </filter>
        ```

```
<filter-mapping>
  <filter-name>Agent</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>INCLUDE</dispatcher>
  <dispatcher>FORWARD</dispatcher>
  <dispatcher>ERROR</dispatcher>
</filter-mapping>
```

2 Log in to the Application Server console.

3 Navigate to Application Server > JVM Settings > Path Settings.

4 Update the classpath suffix.

Remove the following entries that you had added earlier:

```
/opt/SUNWappserver91/domains/domain1/applications/
j2ee-modules/idm/WEB-INF/lib/openssoclientsdk.jar

/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/
idm/WEB-INF/classes
```

At this time, you can also physically delete the `openssoclientsdk.jar` file and the `classes` directory. They are no longer needed.

5 Click Save.

In the following steps, the recommended approach is to update the `web.xml` file (above), recreate the `idm.war`, and then redeploy the new `idm.war` file on the Application Server.

6 Stop the Application Server.

```
# /opt/SUNWappserver91/bin/asadmin stop-domain domain1
```

7 Delete the generated Identity Manager application files.

They will be re-generated when you access the Identity Manager application. If you don't do this step, the changes that you made in the `web.xml` file may not go into effect.

```
# cd /opt/SUNWappserver91/domains/domain1/generated/xml/j2ee-modules

# rm -rf idm
```

8 Start the Application Server.

```
# /opt/SUNWappserver91/bin/asadmin start-domain domain1
```

Watch for any errors in the Application Server `server.log` file.

Configuring Identity Manager for Single Sign-On

When you configure Identity Manager for single sign-on, the user can log into both Identity Manager and OpenSSO Enterprise at one time, and without having to re-authenticate to OpenSSO Enterprise.

To Configure Identity Manager for single sign-on with OpenSSO Enterprise, complete the following steps:

1. Configure Identity Manager Login Module Groups.
2. Configure the Identity Manager user login interface.
3. Configure the Identity Manager administrator login interface.
4. Test single sign-on from OpenSSO Enterprise to Identity Manager .

The following figure illustrates the process flow of single sign-on from OpenSSO Enterprise to Identity Manager.

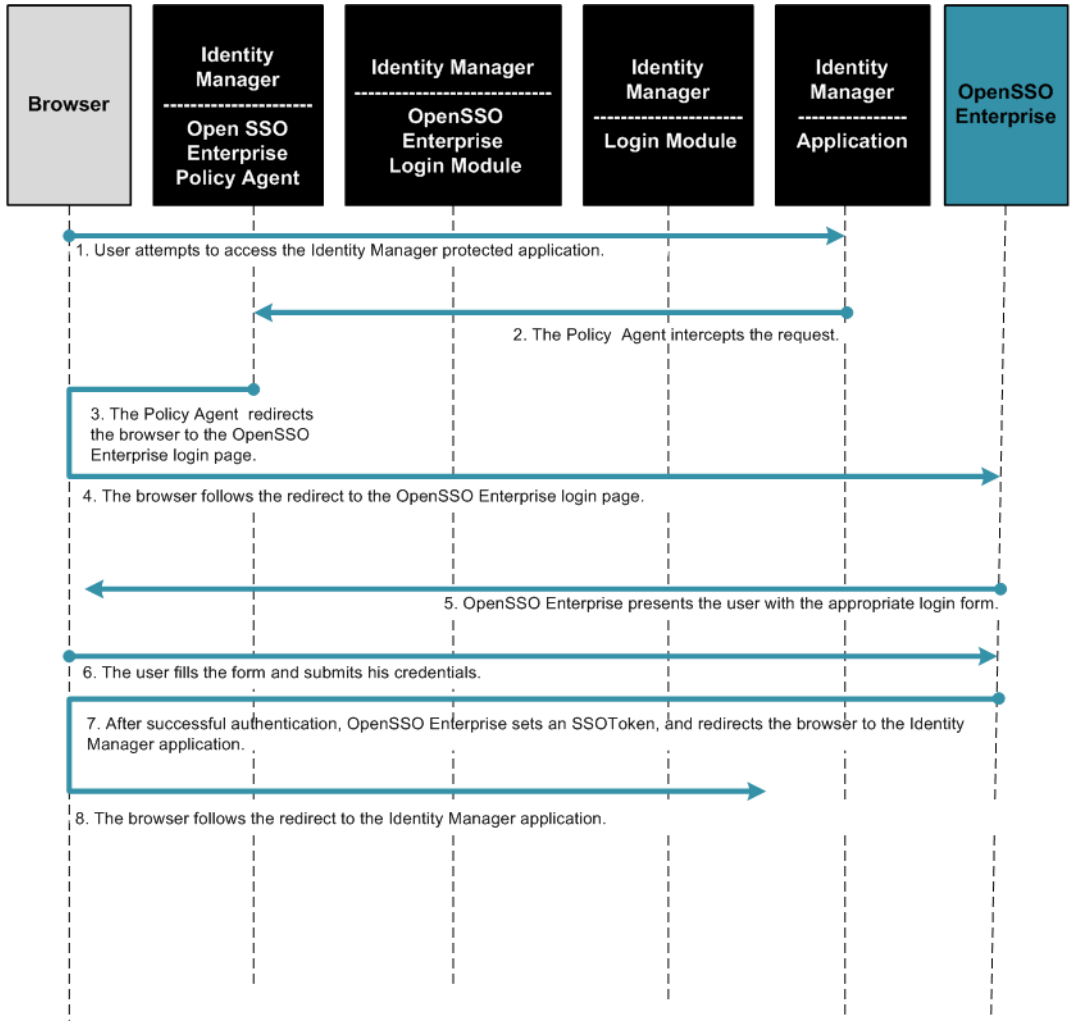


FIGURE 1–3 Single Sign-On Protocol Flow

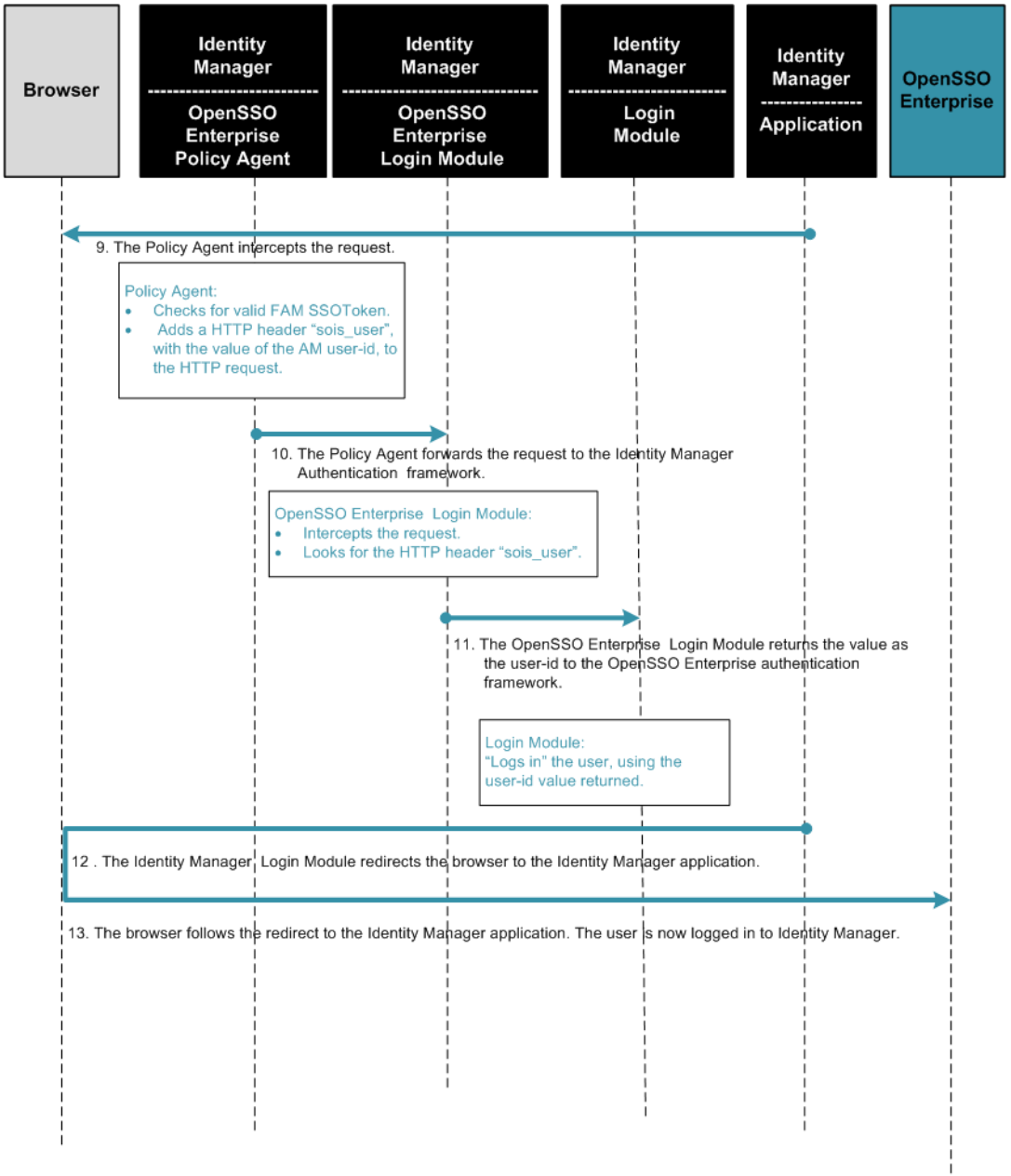


FIGURE 1–4 Single Sign-On Process Flow (continued)

▼ To Configure Identity Manager Login Module Groups

At this point, Identity Manager is not yet protected by the policy agent.

1. **Log in to the Identity Manager administrator interface using the following credentials:**

   User Name: configurator

   Password: configurator

2. **Navigate to the Security > Login tab**

3. **Click "Manage Login Module Groups."**

4. **In the Login Module Groups page, click New.**

5. **In the Create Login Module Group page, provide the following information:**

   Login Module Group Name: Sun OpenSSO Realm

   Assign Login Module: Sun OpenSSO Realm Login Module

   In the second dropdown list: `SunOpenSSORealm`

   The Modify Login Module screen is displayed.

6. **In the Modify Login Module screen, choose the following values:**

   Login success requirement: Sufficient

   Login correlation rule: Leave this field blank. Don't make a selection; leave it the entry at "Select..."

7. **Click Save.**

   The Create Login Module Group page is displayed. A new row is added to the table and describes the selections you made. You should now see one login module listed in the table.

8. **In the Assign Login Module dropdown list, choose "Identity System UserID/Password Login Module."**

   You are redirected to the Modify Login Module page.

9. **In the Modify Login Module page, enter the following values:**

   Login display name: PassThrough

   Login success requirement: sufficient

10 **Click Save.**

You are taken back to the Create Login Module Group. A new row is added to the table and describes the selections you made. You should now see two login modules listed in the table.

11 **Click Save.**

You are redirected to the Login Module Groups screen. Here you will see the custom group you added Sun OpenSSO Realm.

12 **Click "Return To Login Applications."**

▼ To Configure the Identity Manager User Login Interface

You are logged into the Identity Manager administrator interface, and are on the Security > Login tab.

1 **Click on the User Interface hyperlink.**

2 **Remove the "Default Identity System ID/Password Login Module Group. "**

Mark the checkbox beside the entry and click Delete.

3 **In the "Assign Login Module Groups" dropdown list, choose the Sun OpenSSO Realm login module.**

The Modify Login Module page is displayed. You should see just one login module group listed in the table, Sun OpenSSO Realm.

4 **Click Save.**

The Login Applications page is displayed. For the User Interface application, the Sun OpenSSO Realm login module group has been assigned to it.

5 **Log out of the Identity Manager administrator interface.**

▼ To Configure the Identity Manager Administrator Login Interface

At this point, Identity Manager is not yet protected by the policy agent.

1. **Log in to the Identity Manager administrator interface using these credentials:**

   User Name: configurator

   Password: configurator

2. **Navigate to the Security > Login tab.**

3. **Click the Administrator Interface hyperlink.**

4. **Remove the "Default Identity System ID/Password Login Module Group."**

   Mark the checkbox beside the entry and click Delete.

5. **In the Assign Login Module Groups dropdown list, choose the Sun OpenSSO Realm login module.**

   The Modify Login Module page is displayed. You should now see just one login module group listed in the table, Sun OpenSSO Realm.

   Click Save.

6. **Log out of Identity Manager administrator interface.**

Testing Single Sign-On from OpenSSO Enterprise to Identity Manager

To test single sign-on from OpenSSO Enterprise to Identity Manager, follow these steps:

1. Re-Enable Identity Manager protection by the OpenSSO Enterprise Policy Agent.
2. Test Admin-User Single Sign-On Between OpenSSO Enterprise and Identity Manager.

▼ To Re-Enable Identity Manager Protection by the OpenSSO Enterprise Policy Agent

1 Log in to the OpenSSO Enterprise console using the following credentials:

User Name: amadmin

Password: password

2 Navigate to Access Control > Top-Level Realm >Agents > J2EE > idmagent > Application.

3 For the property Not Enforced URI (`com.sun.identity.agents.config.notenforced.uri`), remove the entry you previously added:

```
/idm/*
/idm/*?*
```

4 Make sure these lines are present:

```
/idm/styles/*

/idm/includes/*

/idm/images/*
```

5 Click Save.

6 Log out of the OpenSSO Enterprise console.

▼ To Test End-User Single Sign-On Between OpenSSO Enterprise and Identity Manager

1 Go to the OpenSSO Enterprise login page.

In this example, go to `http://`*ApplicationServerHost:Port*`/idm/user`.

2 Log in using the following credentials:

User Name: idmuser

Password: password

The Identity Manager user page is displayed. You should be single signed-on to Identity Manager, and should not be prompted for login by Identity Manager.

3 Log out of the Identity Manager user page.

▼ To Test Admin-User Single Sign-On Between OpenSSO Enterprise and Identity Manager

1 Go to following Identity Manager URL:

```
http://host1.example.com:2080/idm
```

The OpenSSO Enterprise login page is displayed.

2 Log in using the following credentials:

User Name: idmadin

Password: password

The Identity Manager administrator interface is displayed. You should be single-signed onto Identity Manager, and should not be prompted for login by Identity Manager

3 Log out of Identity Manager.

Configuring Single-Logout Between Identity Manager and OpenSSO Enterprise

When the user logs out from the Identity Manager application, the user should automatically be logged out from OpenSSO Enterprise as well. This is called single-logout.

To configure single-logout between Identity Manager and OpenSSO Enterprise, complete the following steps:

1. Configure OpenSSO Enterprise for single-logout.
2. Test the single-logout configuration.

The following figure illustrates the process flow for single-logout.

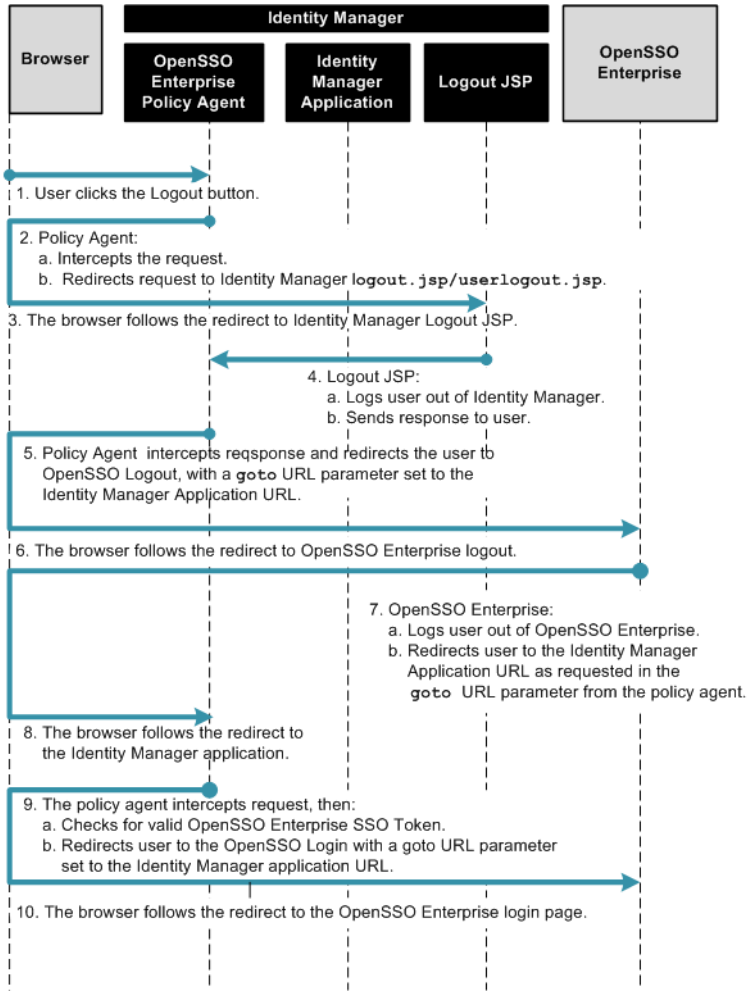


FIGURE 1–5 Process Flow for Single-Logout Between Identity Manager and OpenSSO Enterprise

▼ To Configure OpenSSO Enterprise for Single-Logout

1. **Log in to the OpenSSO Enterprise administration console.**
2. **Navigate to the Policy Agent Profile for the policy agent on Identity Manager.**
   From the Access Control tab, go to Top Level Realm > Agents > J2EE> idmagent.
3. **In the Policy Agent Profile, go to Application > Logout Processing.**

4. **Add the following values to the Application Logout URI property** `com.sun.identity.agents.config.logout.uri`:
   - `logout.uri[idm]=/idm/logout.jsp`
   - `logout.uri[idm/user]=/idm/user/userLogout.jsp`
5. **Add the following values to the Logout Entry URI property** `com.sun.identity.agents.config.logout.entry.uri :`
   - `entry.uri[idm]=/idm`
   - `entry.uri[idm/user]=/idm/user`
6. **Click Save at the top of the page.**

   The properties you have configured are "hot-swappable" properties; they do not require you to restart the server for changes to take effect.
7. **Log out of the OpenSSO Enterprise administration console.**

▼ To Test the Single-Logout Configuration

1. **Log in to the Identity Manager application.**
2. **In the Identity Manager application window, click Logout IDM.**

   You are logged out from both Identity Manager and OpenSSO Enterprise, and then redirected to the OpenSSO Enterprise login page.
3. **Log in to OpenSSO Enterprise.**

   You are automatically redirected to the specific Identity Manager application administrator or user you had logged out from in step 2.

   You have successfully tested that you were logged out from both Identity Manager and OpenSSO Enterprise when you logged out from your Identity Manager application. This is single logout. Additionally, you have verified that when you log in a second time, you are correctly redirected to the appropriate Identity Manager user area.

Configuring First-Time User Login Behavior

When a user logs into the protected application through OpenSSO Enterprise for the first time, after being provisioned, the user should be prompted to set his challenge/response answers. These answers can later be used to verify the user's identity when the user wants to reset a forgotten password.

To configure this first-time user login behavior, complete the following steps:

1. Configure OpenSSO Enterprise first-time user login behavior.
2. Develop a post-authentication plug-in for first-time user login.
3. Compile the post-authentication plug-in code.

The following figures illustrate the process flow for this first-time user login behavior.

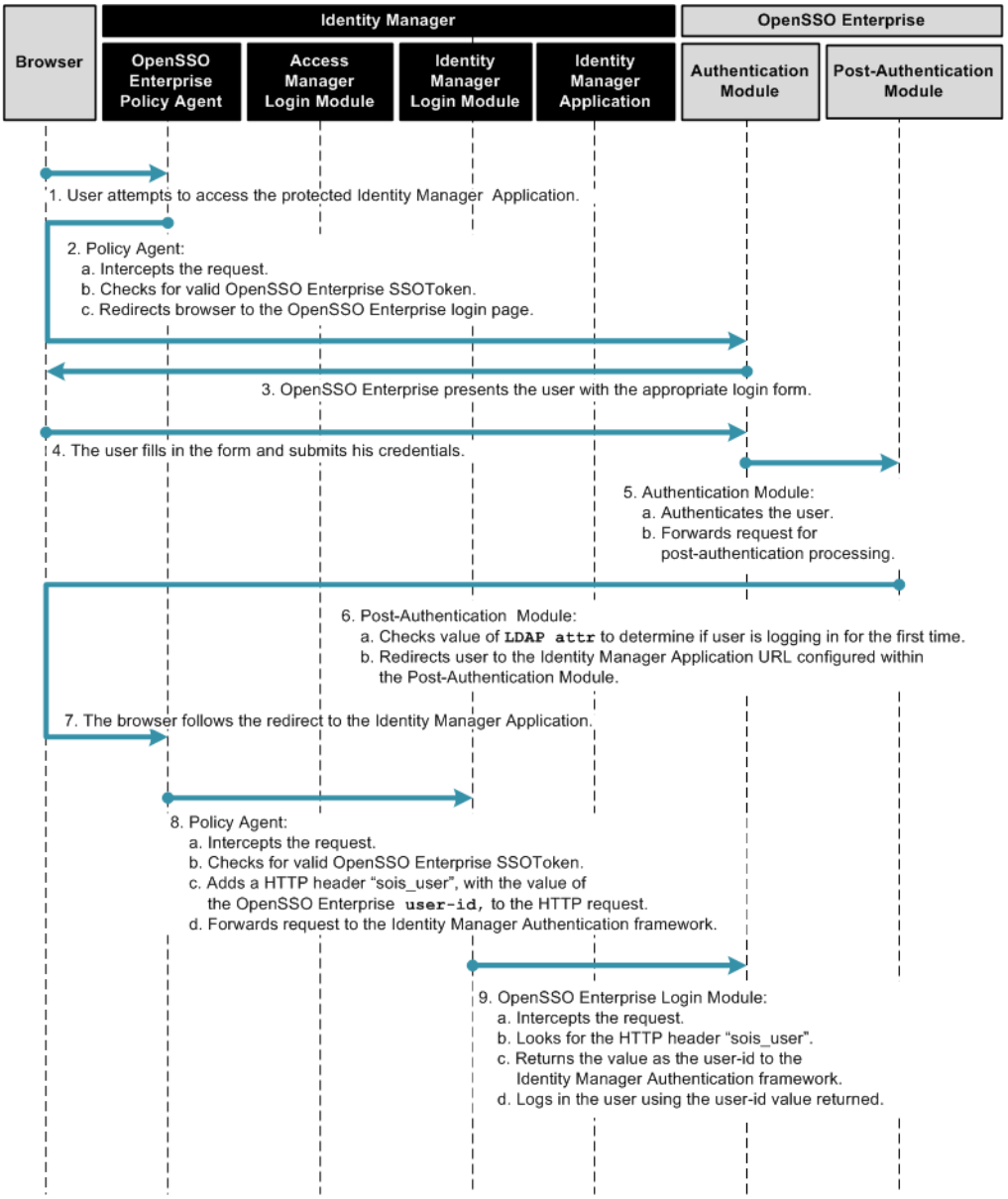


FIGURE 1–6 Process Flow for First-Time User Login

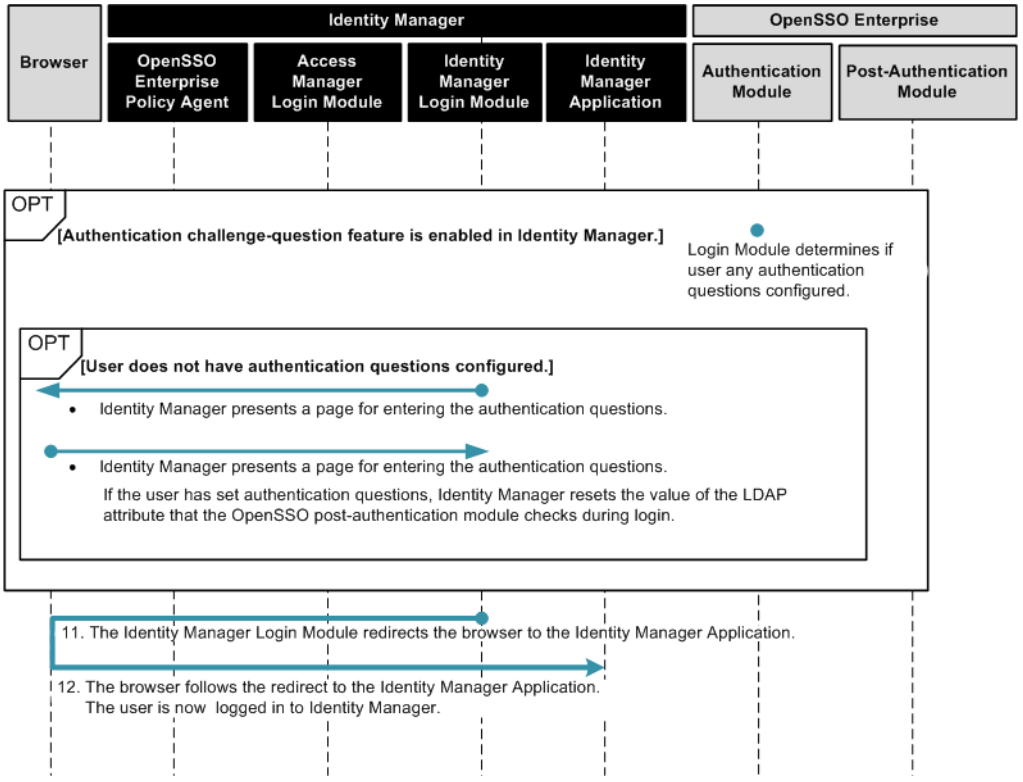


FIGURE 1–7 Process Flow for First-Time User Login (continued)

▼ To Configure OpenSSO Enterprise First-Time User Login Behavior

1 Develop a custom post-authentication plug-in.

You can write your own custom post authorization plug-in, or you can use the sample source code that comes with OpenSSO Enterprise. See "Developing a Post-Authentication Plug-In for First-Time User Login" on page 69 for more information.

2 Compile the post-authentication plug-in code.

See "To Compile the Post-Authentication Plug-In Code" on page 72 for detailed information.

3 Use the OpenSSO Enterprise console to modify first-time user login settings.

a. Log in to the OpenSSO Enterprise administration console.

b. Click the Access Control tab, and then navigate to *RealmName*> Data Stores > *DataStoreName*.

c. For the property LDAP User Attributes, add the LDAP attribute `employeeType`.

d. Click Save.

e. Click "Back to Data Stores."

f. Click the Authentication tab.

g. Click "All Core Settings."

h. For the property "Authentication Post Processing Classes," add the value `com.sun.identity.authentication.spi.FirstTimeLogin`.

i. Click Save.

j. Click "Back to Authentication," and then click "Back to Access Control".

k. Click the Configuration tab, and then navigate to Server & Sites > Default Server Settings > Advanced.

l. Click Add to add a new property. Example:

This is an example of an LDAP attribute name. Use your own LDAP attribute name here.

Property Name: `com.sun.identity.firsttime_login_attr_name`

Property Value: `employeeType`

m. Click Save.

The following warning message is displayed:

"Server Profile was updated. Unidentified property, `com.sun.identity.firsttime_login_attr_name`"

Ignore this warning.

n. Log out of the OpenSSO Enterprise console.

4 Copy your custom post-authentication plug-in classfile (example: `FirstTimeLogin.class`) to the following OpenSSO Enterprise `web-app` directory:

`WEB-INF/classes/com/sun/identity/authentication/spi`

Be sure to create directories that don't already exist to reflect the package. Example:
`/opt/SUNWappserver91/domains/opensso4idm/applications/j2ee-modules/opensso/WEB-INF/classes/com/sun/identity/authentication/spi`

5 **Restart the OpenSSO Enterprise web container for the changes to take effect.**

Developing a Post-Authentication Plug-In for First-Time User Login

Your custom post-authentication plug-in, or module, must minimally perform the following operations:

- Read the value of an LDAP attribute, its name to be specified in an OpenSSO Enterprise property `com.sun.identity.firsttime_login_attr_name`.
- If the value of the attribute is true, determine that the user is logging-in for the first time.
- If the user is logging in for the first time, redirect to the Identity Manager URL where the user can enter or set challenge questions.

Before you begin, determine the LDAP attribute you will use to identify a user who is logging in for the first time, and replace occurrences of `employeeType` in the following instructions with the custom LDAP attribute name. This attribute is ideally a boolean LDAP attribute that takes values `true` or `false`. In the procedures described below, the attribute is `employeeType`.

You can develop your own code based on the code sample made available in the `opensso.zip` distribution. Or you can use the source code that comes with OpenSSO Enterprise. Choose only one of the following procedures:

- "Writing Your Own Post-Authentication Plug-In" on page 69
- "Using the Post-Authentication Plug-In Sample Source Code" on page 72

Writing Your Own Post-Authentication Plug-In

The following code sample is a post-authentication plug-in. In this code sample, OpenSSO Enteprise redirects to an Identity Manager URL if the value of the configured LDAP attribute is `true`.

Replace occurrences of `com.sun.identity.authentication.spi.FirstTimeLogin` with the fully qualified name of your class.

EXAMPLE 1–1 Code Sample: Post-Authentication Plug-In for First-Time Login

```
package com.sun.identity.authentication.spi;
```

EXAMPLE 1–1 Code Sample: Post-Authentication Plug-In for First-Time Login *(Continued)*

```
import com.iplanet.am.util.Debug;
import com.iplanet.am.util.Misc;
import com.iplanet.am.util.SystemProperties;
import com.iplanet.sso.SSOToken;
import com.iplanet.sso.SSOException;
import com.sun.identity.authentication.service.AuthUtils;
import com.sun.identity.authentication.util.ISAuthConstants;
import com.sun.identity.idm.AMIdentity;
import com.sun.identity.idm.IdRepoException;
import com.sun.identity.idm.IdUtils;
import java.io.IOException;
import java.lang.System;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;



public class FirstTimeLogin implements AMPostAuthProcessInterface {

    //add this attribute as an advance property
    private static final String FIRSTTIME_LOGIN_ATTR_NAME =
        "com.sun.identity.firsttime_login_attr_name";

    private static Debug debug = Debug.getInstance("FirstTimeLogin");

    /**
     * Post processing on successful authentication.
     * @param requestParamsMap contains HttpServletRequest parameters
     * @param request HttpServlet  request
     * @param response HttpServlet response
     * @param ssoToken user's session
     * @throws AuthenticationException if there is an error while setting
     * the session paswword property
     */
    public void onLoginSuccess(Map requestParamsMap,
        HttpServletRequest request,
        HttpServletResponse response,
        SSOToken ssoToken) throws AuthenticationException {

        if (debug.messageEnabled()) {
            debug.message("FirstTimeLogin.onLoginSuccess called:
                 Req:" + request.getRequestURL());
```

EXAMPLE 1–1 Code Sample: Post-Authentication Plug-In for First-Time Login *(Continued)*

```
    }

    String strAttributeName = SystemProperties.get(FIRSTTIME_LOGIN_ATTR_NAME);

    try {

        if(strAttributeName != null && !strAttributeName.trim().equals("")){
            AMIdentity amIdentityUser = IdUtils.getIdentity(ssoToken);
            Map attrMap = amIdentityUser.getAttributes();
            String strAttributeValue = Misc.getMapAttr(
                attrMap, strAttributeName, null);
            if (debug.messageEnabled()) {
                debug.message("FirstTimeLogin.onLoginSuccess:
                     " + strAttributeName + "=" + strAttributeValue);
            }
    System.out.println("FirstTimeLogin.onLoginSuccess:
  " + strAttributeName + "=" + strAttributeValue);
          if(strAttributeValue != null && strAttributeValue.equalsIgnoreCase("true")){
              if (request != null){
              request.setAttribute(AMPostAuthProcessInterface.POST_PROCESS_LOGIN_SUCCESS_URL,
                        "http://localhost:8081/idm/user/main.jsp?goto=http://mail.yahoo.com");
                }
              }
        }

        if (debug.messageEnabled()) {
            debug.message("FirstTimeLogin.onLoginSuccess:
                FirstTimeLogin " + "concluded successfully");
        }
    } catch (IdRepoException ire) {
        debug.error("FirstTimeLogin.onLoginSuccess:
             IOException while " + "fetching user attributes: " + ire);
    } catch (SSOException sse) {
        debug.error("FirstTimeLogin.onLoginSuccess:
             SSOException while " + "setting session password property: " + sse);
    }
}

/**
 * Post processing on failed authentication.
 * @param requestParamsMap contains HttpServletRequest parameters
 * @param req HttpServlet request
 * @param res HttpServlet response
 * @throws AuthenticationException if there is an error
 */
```

EXAMPLE 1–1 Code Sample: Post-Authentication Plug-In for First-Time Login *(Continued)*

```
    public void onLoginFailure(Map requestParamsMap,
        HttpServletRequest req,
        HttpServletResponse res) throws AuthenticationException {
            debug.message("FirstTimeLogin.onLoginFailure: called");
    }

    /**
     * Post processing on Logout.
     * @param req HttpServlet request
     * @param res HttpServlet response
     * @param ssoToken user's session
     * @throws AuthenticationException if there is an error
     */
    public void onLogout(HttpServletRequest req,
        HttpServletResponse res,
        SSOToken ssoToken) throws AuthenticationException {
            debug.message("FirstTimeLogin.onLogout called");
    }
}
```

If you want to preserve the value of the OpenSSO Enterprise `goto` URL, and pass it on to Identity Manager, you can do that in the post-authentication plug-in. You can retrieve the original URL parameters from the HTTP request, and incorporate them into the request to the Identity Manager URL. See the "Adding Authentication Post Processing Features" in *Sun OpenSSO Enterprise 8.0 Developer's Guide*

Using the Post-Authentication Plug-In Sample Source Code

The sample source code is contained in file *opensso*`/integrations/idm/src/com/sun/identity/authentication/spi/FirstTimeLogin.java`. Replace occurrences of `com.sun.identity.authentication.spi.FirstTimeLogin` with the fully qualified name of your class. Replace the Identity Manager URL an appropriate URL to suit your deployment.

▼ To Compile the Post-Authentication Plug-In Code

1. **Download the Java EE SDK if you don't have it already through NetBeans or GlassFish.**

   From this SDK, you will need the Java servlet API classes (available in `javaee.jar`) to compile the post-authentication module source code.

2 **Set the Java home directory.**

```
# export JAVA_HOME=/export/software/jdk1.6.0_14
# export PATH=$JAVA_HOME/bin:$PATH
```

3 **Set the Java classpath.**

The following has been broken into multiple lines for readability purposes.

```
# export CLASSPATH=/opt/SUNWappserver91/lib/javaee.jar:
/opt/SUNWappserver91/domains/opensso4idm/applications/
  j2ee-modules/opensso/WEB-INF/lib/amserver.jar:
/opt/SUNWappserver91/domains/opensso4idm/applications/
  j2ee-modules/opensso/WEB-INF/lib/opensso-sharedlib.jar
```

The jar files used in this example are:

- `javaee.jar`

  Available in the Java EE SDK or in your GlassFish / NetBeans installation.

- `amserver.jar`

  Available in the `opensso.war`.

- `opensso-sharedlib.jar`

  Available in the `opensso.war`.

4 **Compile the source file.**

```
# javac FirstTimeLogin.java
```

The class file `FirstTimeLogin.class` is created in the current directory.

Configuring User-Initiated Password Reset

When Identity Manager and OpenSSO Enterprise are fully integrated, you can extend the end-user password reset, or forgotten password feature, to include Identity Manager. Configure challenge questions for each user account for identification purposes when a user needs to change or reset a configured password. If these questions are not answered correctly, password reset is not allowed. The following configurations will work if the user has already configured challenge questions and answers, or if the user needs to configure challenge questions and answers now.

To configure end-user password reset, complete the following steps:

1. Define Identity Manager URLs as Not Enforced.
2. Modify the OpenSSO Enterprise login page.
3. Configure the Identity Manager password controls.
4. Test the User-Initiated Password Reset configuration.

The following figures illustrate the process flow for end-user password reset.

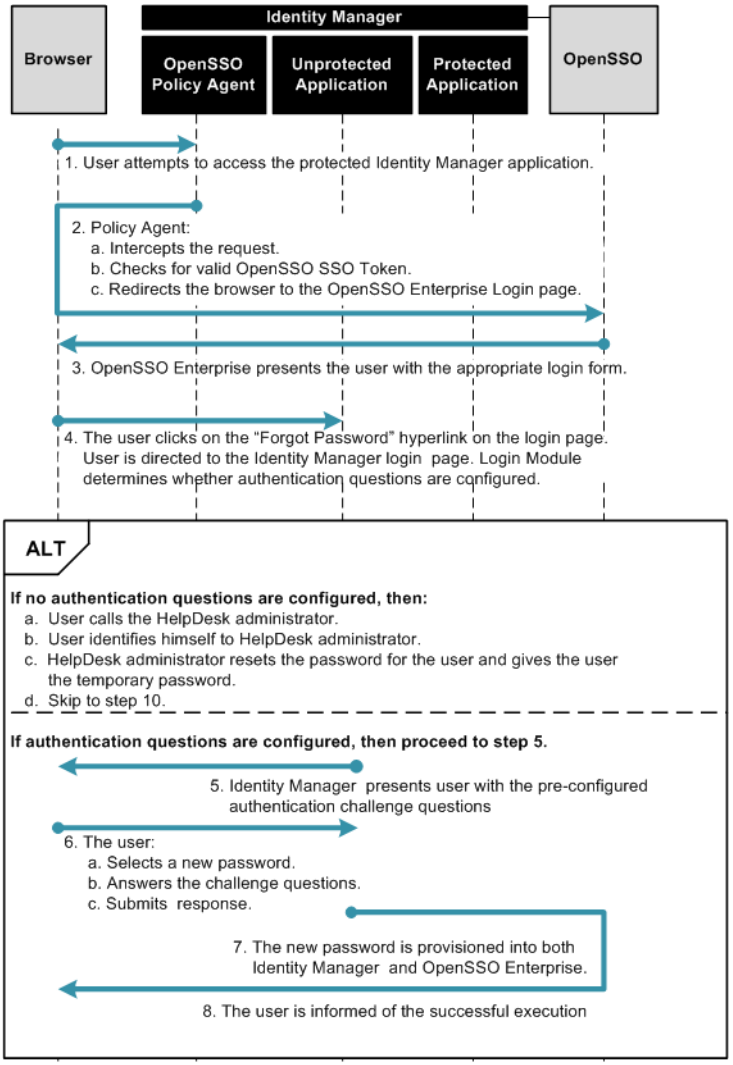


FIGURE 1–8 Process flow for User-Initiated Password Reset

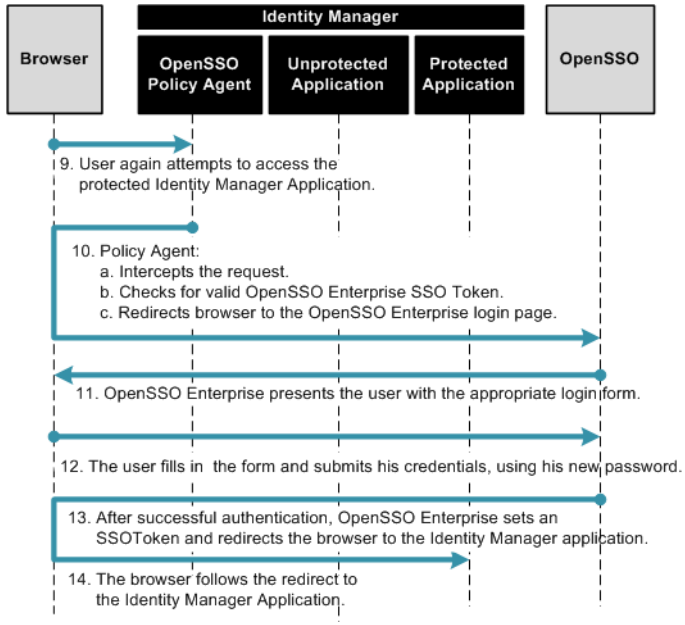


FIGURE 1–9 Process Flow for User-Initiated Password Reset (continued)

▼ To Define Identity Manager URLs as Not Enforced

1. **Log in to the OpenSSO Enterprise console as an administrator.**
2. **Click the Access Control tab.**
3. **Click the appropriate realm name and navigate to the Agents profile for the policy agent that protects Identity Manager.**
4. **Under the Policy Agent Profile, click the Application tab.**
5. **Add the following URIs to the Not Enforced URIs property.**
   - `/idm/authutil/`
   - `/idm/authutil/*`
   - `/idm/authutil/*?*`
6. **Click Save.**
7. **Log out of OpenSSO Enterprise.**

Modifying the OpenSSO Enterprise Login Page

Configure a "Register User" button on the OpenSSO login page. You can manually change the deployed `Login.jsp` file, or you can use the sample `Login.jsp` included with the `opensso.zip` download. Choose only one of the following procedures:

- "To Manually Modify a Deployed Login.jsp" on page 76
- "To Use the Sample `Login.jsp`" on page 79

▼ To Manually Modify a Deployed Login.jsp

1 Use a text editor to open the file

web-container-deploy-base`/opensso/config/auth/default/Login.jsp`

2 Add the sections highlighted in bold in this example:

```
...
<%

String ServiceURI = (String) viewBean.getDisplayFieldValue(viewBean.SERVICE_URI);
String encoded = "false";
String gotoURL = (String) viewBean.getValidatedInputURL(
request.getParameter("goto"), request.getParameter("encoded"), request);
String encodedQueryParams = (String) viewBean.getEncodedQueryParams(request);

if ((gotoURL != null) && (gotoURL.length() != 0)) {
encoded = "true";

}

String replaygotoURL = "";
String goToURL = request.getParameter("goto");
if(gotoURL != null && !gotoURL.equals("null") && (gotoURL.length() > 0)){
replaygotoURL = "&goto=" + goToURL;

}

System.out.println("replaygotoURL: " + replaygotoURL);
%>


<link rel="stylesheet" href="<%= ServiceURI %>/css/styles.css" type="text/css" />
<script language="JavaScript" src="<%= ServiceURI %>/js/browserVersion.js"></script>
<script language="JavaScript" src="<%= ServiceURI %>/js/auth.js"></script>


...
```

```
function LoginSubmit(value) {
aggSubmit();
var hiddenFrm = document.forms['Login'];


if (hiddenFrm != null) {
hiddenFrm.elements['IDButton'].value = value;
if (this.submitted) {
alert("The request is currently being processed");

}

else {
this.submitted = true;
hiddenFrm.submit();

}

}

}


function ForgotPassword() {
//alert("Inside ForgotPassword");
aggSubmit();
var hiddenFrm = document.forms['Login'];
if (hiddenFrm != null) {
if(hiddenFrm.elements[1].value == ""){
alert("Please enter User Name!");
placeCursorOnFirstElm();
}else{
window.location = "http://HostName.DomainName.com:6480/idm/authutil/
     questionLogin.jsp?accountId=
    " + hiddenFrm.elements[1].value + "<%=replaygotoURL%>";
}

}

}


...
```

```
<jato:content name="hasNoButton">
<tr>
<td><img src="<%= ServiceURI %>/images/dot.gif"
width="1" height="15" alt="" /></td>

<td>
<table border=0 cellpadding=0 cellspacing=0>
<script language="javascript">
markupButton(
'<jato:text name="lblSubmit" />',
"javascript:LoginSubmit('<jato:text name="lblSubmit" />')");
</script>

<script language="javascript">
markupButton(
'Forgot Password',
"javascript:ForgotPassword()");
</script>

</table>

</td>
</tr>

<!-- end of hasNoButton -->

</jato:content>


...


<input type="hidden" name="goto" value="<%= gotoURL %>">
<input type="hidden" name="SunQueryParamsString" value="<%= encodedQueryParams %>">
<input type="hidden" name="encoded" value="<%= encoded %>">

<input type="hidden" name="plaingoto" value="<%=
                        request.getParameter("goto") %>">
</auth:form>

</jato:content>


...
```

3. **Replace the beginning of the URL** `http://`*HostName.DomainName*`.com:6480/idm/authutil/questionLogin.jsp?accountId=` **with the specifics of your deployment.**

   The URL in the section of this page that ends `.../idm/authutil/questionLogin.jsp?accountId=` links to the Identity Manager JSP that will be displayed if the user does not have challenge questions configured.

4. **Remove the web container's temporary, compiled JSP to ensure that the changes made are picked up.**

   For example, if using GlassFish, the temporary, compiled classes can be found under *glassfish-home*`/domains/`*your-domain*`/generated/`.

5. **Restart the OpenSSO Enterprise web container after making the changes.**

▼ To Use the Sample `Login.jsp`

1. **Use a text editor to open the file** *opensso*`/integrations/idm/jsps/Login.jsp` **in the decompressed** `opensso.zip` **archive.**

2. **Replace the deployed** */web-container-deploy-base*`/opensso/config/auth/default/Login.jsp` **with the modified**`Login.jsp`**.**

   Choose one of the following options:

   You can replace your existing `Login.jsp` with the sample `Login.jsp`. If you choose this option, the following will occur:

   - You will lose any custom changes made to the existing `Login.jsp`.
   - You will inherit changes that might have been previously made to the sample `Login.jsp` to incorporate requirements for other use cases related to the OpenSSO integration with Identity Manager.
   - You must change the Identity Manager URL embedded in the sample to reflect the Identity Manager system URL of your architecture.

     You can search for the string `/idm` to locate the URLs.

   Alternatively, you can manually make changes to the file. If you choose this option, run the `diff` command to view the differences between the two files.

3. **Remove the web containers temporary, compiled JSP to ensure that the changes made are picked up.**

   For example, if using GlassFish, the temporary, compiled classes can be found under *glassfish-home*`/domains/`*your-domain*`/generated/`.

4 **Restart the OpenSSO Enterprise web container after making the changes.**

Configuring the Identity Manager Password Controls

When the Identity Manager password controls are configured for user-initiated password reset, the following occur:

- The Identity Manager Open SSO Resource Adaptor requires the old password.
- Identity Manager uses the basic change password form.
- Identity Manager self-change is enabled.

▼ To Configure the Identity Manager Password Controls

1 **Log in to Identity Manager as an administrator.**

2 **Navigate to the Configure tab.**

3 **Click on the link "Form and Process Mappings."**

4 **Search for the entry "`endUserChangePassword`. "**

In the text field, replace "End User Change Password Form" with "Basic Change Password Form."

5 **Save the changes.**

▼ To Test the Identity Manager Password Control Configuration

1 **Log in to Identity Manager as a regular user .**

2 **Under the "Profile" tab, go to the "Change password" page.**

You should see that `SunAccessManagerRealm` requires the old password.

3 **Enter the user's the old password, the new password, and confirmation of the new password,**

The user's password should be set in the Directory Server user data store as a "self-change" instead of am "admin-change." This is especially important if the `pwdMustChange` or `passwordMustChange` attributes had been earlier set on the user's profile on the Directory Server. If the self-change configuration is not implemented, when the user logs back into OpenSSO Enterprise, the user will be asked to change his password again

▼ To Test the User-Initiated Password Reset Configuration

1. **Access an Identity Manager URL.**
   You are redirected to the OpenSSO Enterprise login page.
2. **Enter a username and click the Forgot Password button.**
   You are redirected to the Identity Manager `questionLogin.jsp`.
3. **Enter answers to the challenge questions and click the Login button.**
   You are redirected to a second page.
4. **Enter your new password on this second page.**
   This is a temporary password you have received from contacting the HelpDesk.
5. **Select the option to update all resource accounts.**
   Ensure that both the Identity Manager and OpenSSO resources are selected.
6. **Select the option in the column "Forgot Old Password?" for the OpenSSO Resource.**
7. **Click the "Change Password" button.**
   The password is now changed. Use the new password next time you log in.

Configuring Administrator-Initiated Password Reset

By default, when a user password is close to expiring, Directory Server sends a warning to the user. This warning is sent based on the time configured in the password policy. The next time the user attempts to log in to a protected application, OpenSSO Enterprise redirects the user to Identity Manager where the user can change his password. If the user does not change his password, and lets his password expire, he must contact the HelpDesk administrator and request a password-reset.

When a HelpDesk administrator resets the user's password, a flag is set in the user profile. The HelpDesk administrator gives a temporary password to the user by email or over the phone. When the user logs in using the temporary password, the user is redirected to the Identity Manager user interface to reset the user password. After the user password has been reset, the flag that was set earlier is unset.

To configure OpenSSO Enterprise for administrator-initiated password reset, complete the following steps:

1. Configure Directory Server.
2. Configure OpenSSO Enterprise for Administrator-Initiated Password Reset.
3. Configure the Identity Manager password controls.
4. Test Administrator-Initiated Password Reset configurations.

The following figure illustrates in detail the process flow for administrator-initiated password reset.

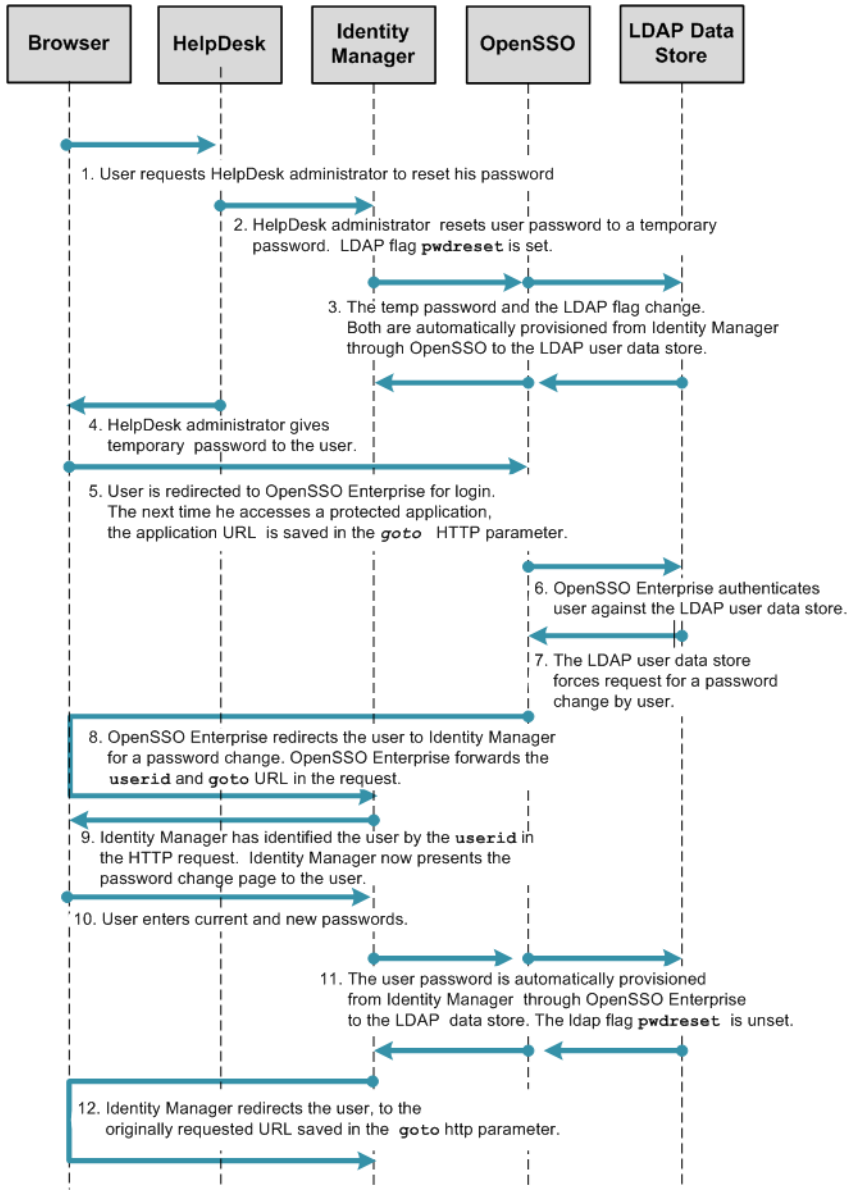


FIGURE 1–10 Process Flow for Administrator-Initiated Password Reset

Configuring Directory Server

Before you can configure OpenSSO Enterprise for administrator-initiated password reset, you must configure the Directory Server must to meet the following conditions:

- A password policy is configured and assigned to the test user's LDAP profile in the directory server. The password policy should have the following controls set:

  Set Password Expiration
  LDAP attributes: `passwordexp`, `passwordmaxage`

  Set Expiration Warning
  LDAP attribute: `passwordwarning`

  Warning Duration
  LDAP attribute: `passwordExpireWithoutWarning`)

- The following controls are set to allow for administrator-driven password reset:

  Require Password Change at First Login and After Reset
  LDAP attribute: `passwordchange`, `passwordmustchange`

  Allow Users to Change Their Passwords
  LDAP attribute: `pwdallowuserchange`

- The `passwordPolicySubentry` attribute in the test user's LDAP profile is set with the DN of the password policy. This indicates that the password policy has been assigned to this user. Example:

  ```
  cn=idm_integration,dc=sun,dc=com
  ```

See the *Sun Java System Directory Server Enterprise Edition 6.3 Administration Guide* for detailed instructions on configuring these settings.

Important Information About Using Sun Directory Server 6.3

After you install Sun Directory Server Enterprise Edition 6.3, Directory Server uses Legacy mode for its password policy syntax, which works for both Directory Server 5.x and Directory Server 6.x. However, Directory Server 6.3 maintains two sets of password attributes for both password policies and the user's computed password attributes. This may trigger other potential issues. Unless you are planning to use Directory Server 5.x password policies, a good practice is to migrate a new Directory Server 6.3 instance to the Directory Server 6-Only mode. Doing so removes redundancies and avoids any potential problems.

Here is an example of how you can verify which mode the Directory Server is running in, and how you can enable Directory Server 6–Only mode.

```
# DirectoryServer-base/ds6/bin/dsconf get-server-prop -p 1389 -D "cn=directory manager"
-w mypass -c -e pwd-compat-mode
pwd-compat-mode  :  DS5-compatible-mode

# DirectoryServer-base/ds6/bin>dsconf pwd-compat -p 1389 -D "cn=directory manager"
-w mypass -c -e to-DS6-migration-mode
## Beginning password policy compatibility changes.
## Password policy compatibility changes finished.
```

```
Task completed (slapd exit code: 0).

# DirectoryServer-base/ds6/bin/dsconf pwd-compat -p 1389 -D "cn=directory manager"
-w mypass -c -e to-DS6-mode
## Beginning password policy compatibility changes.
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).

# DirectoryServer-base/ds6/bin/dsconf get-server-prop -p 1389 -D "cn=directory manager"
-w mypass -c -e pwd-compat-mode
pwd-compat-mode  :  DS6-mod
```

Configuring OpenSSO Enterprise for Administrator-Initiated Password Reset

Only the OpenSSO Enterprise LDAP authentication module supports the password change controls enforced by most directory servers.

To configure OpenSSO Enterprise for administrator-initiated password reset, complete the following steps:

1. Enable LDAP authentication.
2. Define Identity Manager URLs as Not Enforced.
3. Create a Custom `ChangePassword.jsp` file.
4. Modify the LDAP authentication module XML service file.
5. Modify the OpenSSO login page.

▼ To Enable LDAP Authentication

1 Log in to the OpenSSO Enterprise console as an administrator.

2 Click the Access Control tab.

3 Navigate to Top Level Realm > Authentication > Authentication Chaining.

4 In the Authentication Chaining section, click New.

5 Enter a name for the chain and click OK.

For this example: `idmauth`.

6 On the new chain's Properties page, add the LDAP module as REQUIRED, and click Save.

7. **Click Back to Authentication.**
8. **For the Organization Authentication Configuration property value, choose the service you just created.**
9. **In the Module Instances section, choose LDAP.**
10. **Provide the following information to about the LDAP user data store:**

Primary LDAP Server:	Use the form *server.domain:port*
DN to Start User Search:	Branch of the LDAP tree from which the user-search begins
DN for Root User Bind:	DN to use when binding to the LDAP user data store
Password for Root User Bind:	Password for the user binding to the LDAP user data store
Password for Root User Bind (confirm):	Type the password again

11. **Save the changes.**
12. **Log out of the OpenSSO Enterprise console.**

Next Steps After completing this configuration:

- Use `/opensso/console` to log in to the OpenSSO Enterprise console; do not `/opensso/UI/Login`. This ensures that the authentication module configured for the OpenSSO Enterprise administrator is used when logging into the administration console, and that the LDAP module just configured for realm users is not used.
- Make sure this configuration hasn't affected how you can view the objects inside the OpenSSO Enterprise resource inside Identity Manager.

  Log in to the Identity Manager console and expand the OpenSSO Enterprise resource listing to view the OpenSSO Enterprise roles and groups inside it. If you receive an error, you may need to reconfigure the OpenSSO Enterprise adaptor to use a delegated administrator instead of `amadmin` to connect to OpenSSO Enterprise. The Identity Manager adaptor for OpenSSO Enterprise authenticates to OpenSSO Enterprise using the authentication configuration for the realm which is now different from the configuration for the OpenSSO Enterprise console. The `amadmin` will no longer work.

  To create the delegated administrator:

  1. Create a user.
  2. Assign this user to a group.
  3. Assign administrator privileges to this group.

See "Delegating Administrator Privileges" in *Sun OpenSSO Enterprise 8.0 Administration Guide* for detailed information on delegating administrative privileges to a group.

▼ To Define Identity Manager URLs as Not Enforced

1. **Log in to the OpenSSO Enterprise console as an administrator.**
2. **Click the Access Control tab.**
3. **Click the appropriate realm name and navigate to the Policy Agent Profile for the policy agent that protects Identity Manager.**
4. **Under the Policy Agent Profile, click the Application tab.**
5. **Add the following URIs to the Not Enforced URIs property:**
   - `/idm/authutil/`
   - `/idm/authutil/*`
   - `/idm/authutil/*?*`
6. **Click Save.**
7. **Log out of OpenSSO Enterprise.**

Creating a Custom `ChangePassword.jsp` File

By default, the user is directed to the OpenSSO Enterprise password change page. Create a custom JSP file, `ChangePassword.jsp`, that redirects a user to Identity Manager for password change events. The new `ChangePassword.jsp` forwards the following information to Identity Manager:

- The original URL requested by the user and defined as the value of the `goto` parameter
- The user identifier defined as the value of the `accountId` parameter

This customized `ChangePassword.jsp` file is referenced in the section "Modifying the LDAP Authentication Module XML Service File" on page 89.

To create a custom `ChangePassword.jsp` file, choose only one of the following procedures:

- "To Create a New `ChangePassword.jsp` File" on page 88
- "To Use the Sample Source Code" on page 89

▼ To Create a New `ChangePassword.jsp` File

1 **Create the file `config/auth/default/ChangePassword.jsp` in the OpenSSO Enterprise `web-app` directory.**

Example:

```
<html>

<%@page info="Login" language="java"%>
<%@taglib uri="/WEB-INF/jato.tld" prefix="jato"%>
<%@taglib uri="/WEB-INF/auth.tld" prefix="auth"%>
<jato:useViewBean className="com.sun.identity.authentication.UI.LoginViewBean">


<%@ page contentType="text/html" %>

<head>
<title><jato:text name="htmlTitle_Login" /></title>

<%
String ServiceURI = (String) viewBean.getDisplayFieldValue(viewBean.SERVICE_URI);
%>

<link rel="stylesheet" href="<%= ServiceURI %>/css/styles.css" type="text/css" />
<script language="JavaScript" src="<%= ServiceURI %>/js/browserVersion.js"></script>
<script language="JavaScript" src="<%= ServiceURI %>/js/auth.js"></script>
</head>

<%
    System.out.println("AccountId: " + request.getParameter("IDToken1"));
    System.out.println("goto: " + request.getParameter("goto"));
        System.out.println("plaingoto: " + request.getParameter("plaingoto"));

    String accountId = request.getParameter("IDToken1");
    String gotoURL = request.getParameter("plaingoto");

    String redirectURL =
        "http://HostName.DomainName:6480/idm/authutil/anonResetPassword.jsp";
if(accountId != null){
        redirectURL = redirectURL + "?accountId=" + accountId;
        }
if(gotoURL != null && !gotoURL.equals("null") && (gotoURL.length() > 0)){
        if(accountId == null){
            redirectURL = redirectURL + "?goto=" + gotoURL;
                }else{
            redirectURL = redirectURL + "&goto=" + gotoURL;
```

```
                }
            }
        System.out.println("Redirect URL is:" + redirectURL);
            response.sendRedirect(redirectURL);
%>

</jato:useViewBean>

</html>
```

2 **Customize the URL to the Identity Manager page that performs the password-reset functions.**

The URL is highlighted in bold above. Determine this URL with help from your Identity Manager administrator, and customize the URL for your deployment.

▼ To Use the Sample Source Code

1 **Copy the sample file** *opensso*`/integrations/idm/jsps/ChangePassword.jsp` **in the** `opensso.zip` **distribution to the directory** *web-container-deploy-base*`/opensso/config/auth/default`**.**

2 **Customize the URL to the Identity Manager page that performs the password-reset functions.**

Determine this URL with help from your Identity Manager administrator, and customize the URL for your deployment. See step 2 of "Creating a Custom `ChangePassword.jsp` File " on page 87.

Modifying the LDAP Authentication Module XML Service File

By default, upon receiving a directory-server request for the user to change his password, OpenSSO Enterprise directs the user to its own password-change page. Configure OpenSSO Enterprise to use a custom JSP for password-change events. Modify the `LDAP.xml`file to use `ChangePassword.jsp` that you created in "Creating a Custom `ChangePassword.jsp` File " on page 87.

You can manually change the deployed `LDAP.xml` file, or you can use the sample `LDAP.xml` included with the `opensso.zip` download. Choose only one of the following procedures:

- "To Manually Modify a Deployed `LDAP.xml` File" on page 90
- "To Use the Sample `LDAP.xml`" on page 90

▼ To Manually Modify a Deployed `LDAP.xml` File

1. **Use a text editor to open the** */web-container-deploy-base*`/opensso/config/auth/default/LDAP.xml` **file, and add the section of code highlighted in bold in the following example:**

```
PasswordCallback echoPassword="false" >
<Prompt> Password: </Prompt>
</PasswordCallback>

</Callbacks>
<Callbacks length="4" order="2" timeout="120" template="ChangePassword.jsp"
header="Change Password&lt;BR&gt;&lt;/BR&gt;#REPLACE#&lt;BR&gt;&lt;/BR&gt;" >

<PasswordCallback echoPassword="false" >
<Prompt>Old Password </Prompt>
</PasswordCallback>
```

2. **Use a text editor to open the** */web-container-deploy-base*`/opensso/config/auth/default_en/LDAP.xml` **file, and make the same change as in step 1.**

▼ To Use the Sample `LDAP.xml`

1. **Change to the** `opensso/integrations/idm/xml/` **directory in the decompressed** `opensso.zip` **to access the sample** `LDAP.xml`**.**

2. **Replace your deployed** */web-container-deploy-base*`/opensso/config/auth/default/LDAP.xml` **with the a custom**`LDAP.xml`**.**

   Choose only one of the following options:

   - Replace your existing `LDAP.xml` file with the sample `LDAP.xml` file. If you choose this option, you will lose any custom changes you may have made to this file earlier.
   - Run the `diff` command to compare the files, then manually make the necessary changes.

Modifying the OpenSSO Login Page

Embed code into the OpenSSO Enterprise `Login.jsp` file that will save the URL in the HTTP request parameter `goto`. This URL is required by the `ChangePassword.jsp` that you created in the section "To Create a New `ChangePassword.jsp` File" on page 88. Once saved, the URL can be passed onto Identity Manager. Identity Manager later redirects the user to that URL.

The URL in the HTTP request parameter `goto` is the original URL requested by the user, before he was redirected to OpenSSO Enterprise for login.

You can manually change the deployed `Login.jsp` file, or you can use the sample `Login.jsp` included with the `opensso.zip` download. Choose only one of the following procedures:

- "To Modify a Deployed `Login.jsp` File" on page 91
- "To Use the Sample `Login.jsp`" on page 92

▼ To Modify a Deployed `Login.jsp` File

1 **Use a text editor to open the file** *web-container-deploy-base*`/opensso/config/auth/default/Login.jsp` **and add the sections of code displayed in bold the following example:**

```
...

<%

String ServiceURI = (String) viewBean.getDisplayFieldValue(viewBean.SERVICE_URI);
String encoded = "false";
String gotoURL = (String) viewBean.getValidatedInputURL(
request.getParameter("goto"), request.getParameter("encoded"), request);
String encodedQueryParams = (String) viewBean.getEncodedQueryParams(request);
if ((gotoURL != null) && (gotoURL.length() != 0)) {
encoded = "true";

}

String replaygotoURL = "";
String goToURL = request.getParameter("goto");
if(gotoURL != null && !gotoURL.equals("null") && (gotoURL.length() > 0)){
replaygotoURL = "&goto=" + goToURL;

}

System.out.println("replaygotoURL: " + replaygotoURL);
%>


<link rel="stylesheet" href="<%= ServiceURI %>/css/styles.css" type="text/css" />
<script language="JavaScript" src="<%= ServiceURI %>/js/browserVersion.js"></script>
<script language="JavaScript" src="<%= ServiceURI %>/js/auth.js"></script>

...


<input type="hidden" name="goto" value="<%= gotoURL %>">
<input type="hidden" name="SunQueryParamsString" value="<%= encodedQueryParams %>">
<input type="hidden" name="encoded" value="<%= encoded %>">
```

```
<input type="hidden" name="plaingoto" value="<%= request.getParameter("goto") %>">
</auth:form>

</jato:content>


...
```

2. **Remove the web container's temporary, compiled JSP to ensure that the changes made are picked up.**

   For example, if you are using GlassFish, the temporary, compiled classes are in the *glassfish-home*`/domains/`*your-domain*`/generated/` directory.

3. **Restart the OpenSSO Enterprise web container after making the changes.**

▼ To Use the Sample `Login.jsp`

1. **Locate the file** *opensso*`/integrations/idm/jsps/Login.jsp` **in the decompressed** `opensso.zip` **distribution.**

2. **Replace the deployed** *web-container-deploy-base*`/opensso/config/auth/default/Login.jsp` **file.**

   Choose only one of the following two options:

   You can replace your existing `Login.jsp` with the sample `Login.jsp`. If you choose this option, the following will occur:

   - You will lose any custom changes made to the existing `Login.jsp`.
   - You will inherit changes that might have been previously made to the sample `Login.jsp` to incorporate requirements for other use cases related to the OpenSSO integration with Identity Manager.
   - You must change the Identity Manager URL embedded in the sample `Login.jsp` to reflect the Identity Manager system URL of your architecture.

     You can search for the string `/idm` to locate the URLs.

   Alternatively, you can make the changes manually. If you choose this option, first run the `diff` command to view the differences between the deployed `Login.jsp` and the sample `Login.jsp`, and then make the changes manually.

3 **Remove the web containers temporary, compiled JSP to ensure that the changes made are picked up.**

For example, if you are using GlassFish, the temporary, compiled classes is in the *glassfish-home*/domains/*your-domain*/generated/ directory.

4 **Restart the OpenSSO web container after making the changes.**

Configuring the Identity Manager Password Controls

When the Identity Manager password controls are configured for administrator-initiated password reset, the following occur:

- The Identity Manager Open SSO Resource Adaptor requires the old password.
- Identity Manager uses the basic change password form.
- Identity Manager self-change is enabled.

▼ To Configure the Identity Manager Password Controls

1 **Log in to Identity Manager as an administrator.**

2 **Navigate to the Configure tab.**

3 **Click on the link "Form and Process Mappings."**

4 **Search for the entry "**endUserChangePassword**. "**

In the text field, replace "End User Change Password Form" with "Basic Change Password Form."

5 **Save the changes.**

▼ To Test the Identity Manager Password Control Configuration

1 **Login to Identity Manager as a regular user .**

2 **Under the "Profile" tab, go to the "Change password" page.**

You should see that SunAccessManagerRealm requires the old password.

3 **Enter the user's the old password, the new password, and confirmation of the new password,**

The user's password should be set in the Directory Server user data store as a "self-change" instead of am "admin-change." This is especially important if the pwdMustChange or passwordMustChange attributes had been earlier set on the user's profile on the Directory

Server. If the self-change configuration is not implemented, when the user logs back into OpenSSO Enterprise, the user will be asked to change his password again

Testing Administrator-Initiated Password Reset Configurations

To verify the behavior for each stage of this use case, perform the following validation tests in this exact order:

1. Test the password expiration warning.
2. Test the password expiration.
3. Test the administrator-initiated password reset.

▼ To Test the Password Expiration Warning

Complete the following steps after the time for the password expiration warning, as defined in the password policy, would take effect.

1 Access a URL protected by OpenSSO Enterprise .

The OpenSSO login page is displayed.

2 Enter the test user name and password.

You are redirected to Identity Manager to change your password. Note the following about the Identity Manager URL:

- The URL is the one configured in `ChangePassword.jsp`.
- The user will be forwarded to the value of the `goto` parameter after the password has been successfully changed.
- The value of the `accountId` parameter determines the account for which the password needs to be changed. Identity Manager will make the changes to the password on both Identity Manager and OpenSSO Enterprise .

▼ To Test the Password Expiration

Complete the following steps after the time the password should have expired, as defined in the password policy.

1 Access a URL protected by OpenSSO Enterprise.

The OpenSSO Enterprise login page is displayed.

2 **Enter the test user name and password.**

An error page is displayed indicating the test user that the password has expired. The user is then instructed to ask the administrator to reset the password.

▼ To Test Administrator-Initiated Password Reset

Before You Begin The Directory Server must have logging and auditing features enabled. Use these features to monitor the Directory Server audit log as you complete the test. See the *Sun Java System Directory Server Enterprise Edition 6.3 Administration Guide* .

1 **Log in as the Directory Administrator, and change the password for a test user.**

This simulates the password reset by a HelpDesk administrator.

2 **Verify that the user's `userPassword` attribute was modified, and that the `pwdreset` attribute was set to TRUE using the audit log.**

The `pwdreset` attribute will force the user to change the password at the next login. The audit log might resemble this sample:

```
time: 20090713074720
dn: uid=idmuser1,dc=sun,dc=com
changetype: modify
replace: userPassword
userPassword: {SSHA}4Bgy/HF9SGN9nnS4Ii6/KJj9ktFdAxQUIDvwVQ==
-
replace: modifiersname
modifiersname: cn=admin,cn=administrators,cn=dscc
-
replace: modifytimestamp
modifytimestamp: 20090713144720Z
-
replace: passwordexpirationtime
passwordexpirationtime: 19700101000000Z
-
replace: pwdreset
pwdreset: TRUE
```

3 **Access the Identity Manager user URL.**

You are redirected to OpenSSO Enterprise for login.

4 **Enter the test user name and password.**

You are redirected to Identity Manager to change your password. Note the following about the Identity Manager URL:

- The URL is the one configured in `ChangePassword.jsp`.

- The user is forwarded to the value of the `goto` parameter after the password has been successfully changed.
- The value of the `accountId` parameter determines the account for which the password needs to be changed. Identity Manager will make the changes to the password on both Identity Manager and OpenSSO Enterprise.

Troubleshooting Administrator-Initiated Account Unlock

If you cannot log in to OpenSSO Enterprise, verify that you are using the correct userid and password. The Directory Administrator who reset your password should have communicated to you the temporary password for the user account.

Monitor the Directory Server's access log, during login. You should see successful SRCH and BIND operations, for the user. Example:

```
[15/Jul/2009:09:32:12 -0700] conn=158 op=9 msgId=269 -
SRCH base="dc=sun,dc=com" scope=2 filter="(uid=idmuser1)" attrs="dn uid"

[15/Jul/2009:09:32:12 -0700] conn=158 op=9 msgId=269 -
RESULT err=0 tag=101 nentries=1 etime=0

[15/Jul/2009:09:32:12 -0700] conn=160 op=5 msgId=270 -
BIND dn="uid=idmuser1,dc=sun,dc=com" method=128 version=3

[15/Jul/2009:09:32:12 -0700] conn=160 op=5 msgId=270 -
RESULT err=0 tag=97 nentries=0 etime=0 dn="uid=idmuser1,dc=sun,dc=com"
```

The string `err=0` in the entries above indicates success for that operation.

After you log in to OpenSSO Enterprise , if you are not redirected to the Identity Manager page, check the following :

- Be sure that your OpenSSO Enterprise web-container is using the changed or new files, that you configured above. The web-container may be using an old pre-compiled version of the default JSP files.
- Be sure the IDM URLs that you embedded in the JSP files are accurate and don't contain typographic errors.
- Browse through the OpenSSO Enterprise web-container logs and look for any reported errors.
- Browse through the OpenSSO Enterprise debug logs, especially the Authentication and IdRepo logs, to check for any reported errors or exceptions.
- Browse through the OpenSSO Enterprise Authentication debug log to determine which `LDAP.xml` file is being looked up, and be sure that specific file was actually modified by you. Depending upon your browser configuration for localization, OpenSSO Enterprise might

be looking for `LDAP.xml` in a different directory. For example, you may have modified just the `config/auth/default/LDAP.xml` file, but OpenSSO Enterprise might be using the `/config/auth/default_en/LDAP.xml` file.

Configuring User-Initiated Account Unlock

You can configure OpenSSO Enterprise so that when an end-user account is locked, the user can unlock the account without intervention from an administrator. The OpenSSO LDAP authentication module supports the account lockout controls enforced by most directory servers.

To configure OpenSSO Enterprise for user-initiated account unlock, complete the following steps:

1. Configure the Directory Server.
2. Configure OpenSSO Enterprise for user-initiated account unlock.
3. Test the user-initiated account unlock configurations.

A user can be locked out of his or her account, or unable to log in, due to memory account lockout or physical account lockout.

Memory account lockout
This occurs when the user has exceeded the allowed number of failed attempts to log in as configured in the password policy. The user may remain locked out for a set period of time and can only reset the password after that period has passed. The locked state of the user account is maintained in memory and no information is written to the user's LDAP profile.

Physical account lockout
This occurs when the status of a specified LDAP attribute in the user's profile is explicitly changed to Inactive, either by an administrator or as a result of some automated processes. The specified LDAP attribute is defined as the value of the Lockout Attribute Name attribute in the Core authentication module. By default it is `inetuserstatus`.

The following figures illustrate the process flow for user-initiated account unlock.

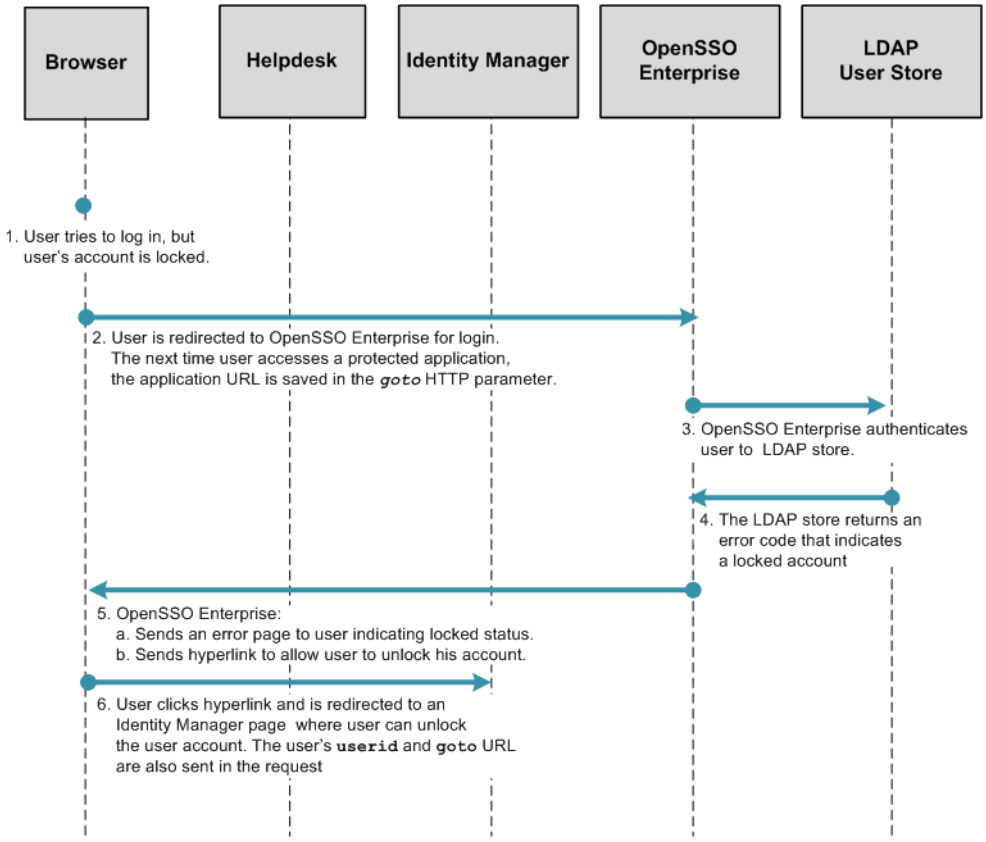


FIGURE 1–11 Process Flow for User-Initiated Account Unlock

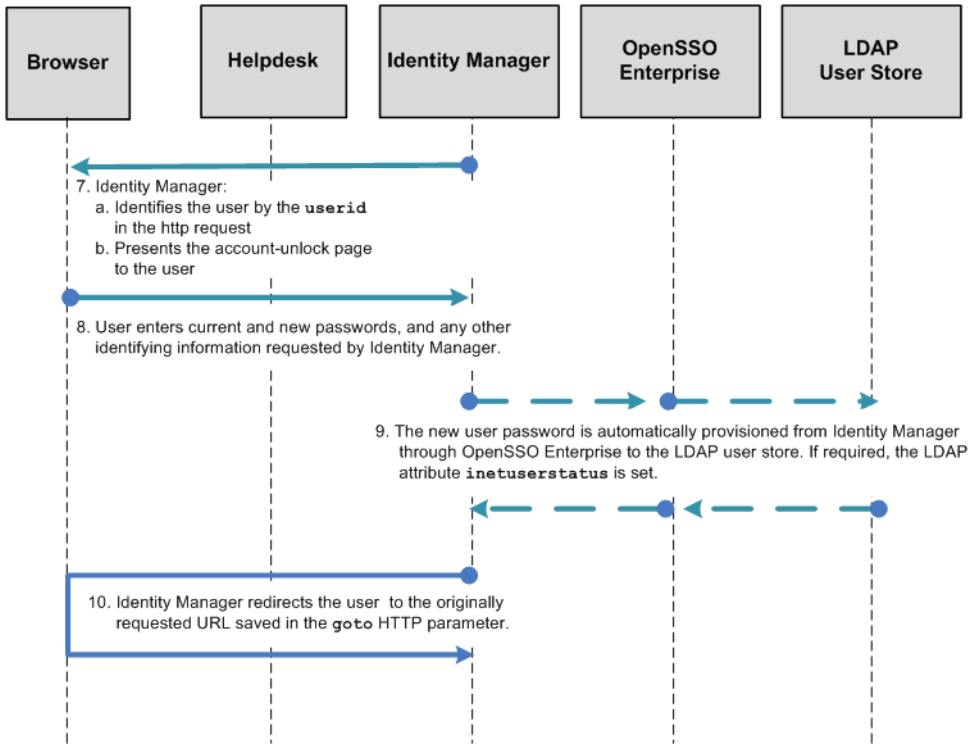


FIGURE 1–12 Process Flow for User-Initiated Account Unlock (continued)

Configuring the Directory Server

Before you can configure OpenSSO Enterprise for administrator-initiated password reset, you must configure the Directory Server must to meet the following conditions:

- A password policy is configured and assigned to the test user's LDAP profile in the directory server. The password policy should have the following controls set:

  Enable Account Lockout
  LDAP attribute: `passwordLockout`

  Failures Before Lockout
  LDAP attribute: `passwordMaxFailure`

  Failure Count Reset
  LDAP attribute: `passwordResetFailureCount`

  Set Limit on Lockout Duration
  LDAP attribute: `passwordUnlock`

Lockout Duration
LDAP attribute: `passwordLockoutDuration`

- The `passwordPolicySubentry` attribute in the test user's LDAP profile is set with the DN of the password policy. This indicates that the that the password policy has been assigned to this user. Example:

  `cn=idm_integration,dc=sun,dc=com`

See the *Sun Java System Directory Server Enterprise Edition 6.3 Administration Guide*for detailed instructions on configuring these settings.

Configuring OpenSSO Enterprise for User-Initiated Account Unlock

The integration between OpenSSO Enterprise and Identity Manager supports only LDAP authentication for this user-initiated account unlock. The LDAP authentication module in OpenSSO supports account-lockout controls enforced by most directory servers.

To configure OpenSSO Enterprise for user-initiated account unlock, complete the following steps:

1. Enable LDAP authentication.
2. Define Identity Manager URLs as Not Enforced.
3. Modify the Account Lockout Message page.

▼ To Enable LDAP Authentication

Before You Begin Memory account lockout in OpenSSO Enterprise must be disabled because the account lockout controls in the user data store will be used. You can use the OpenSSO Enterprise console to disable memory account lockout. See "Enabling Account Lockout" in *Sun OpenSSO Enterprise 8.0 Administration Guide* for detailed information.

1 Log in to the OpenSSO Enterprise console as administrator.

2 Click the Access Control tab.

3 Navigate to Top Level Realm > Authentication > Authentication Chaining.

4 In the Authentication Chaining section, click New.

5 Enter a name for the chain and click OK.

For this example: `idmauth`.

6 On the new chain's Properties page, add the LDAP module as REQUIRED, and click Save.

7. **Click Back to Authentication.**
8. **For the value of Organization Authentication Configuration, choose the service just created .**
9. **Save changes and log out of OpenSSO Enterprise.**

Next Steps After completing this configuration, use `/opensso/console` to log in to the OpenSSO Enterprise console; do not `/opensso/UI/Login`. This ensures that the authentication module configured for the OpenSSO Enterprise administrator is used when logging into the OpenSSO Enterprise console, and that the LDAP module just configured is not used.

▼ To Define Identity Manager URLs as Not Enforced

1. **Log in to the OpenSSO Enterprise console as administrator.**
2. **Click the Access Control tab.**
3. **Click the appropriate realm name and navigate to the Policy Agents profile for the policy agent that protects Identity Manager.**
4. **Under the policy agent profile, click the Application tab.**
5. **Add the following URIs to the Not Enforced URIs property.**
   - `/idm/authutil/`
   - `/idm/authutil/*`
   - `/idm/authutil/*?*`
6. **Click Save.**
7. **Log out of OpenSSO Enterprise.**

Modifying the OpenSSO Enterprise Login Page

Embed code into the OpenSSO Enterprise `Login.jsp` file that will save the URL in the HTTP request parameter `goto`. This URL is required by the `user_inactive.jsp` that you will create in a subsequent procedure. The URL is saved so that it can be passed onto Identity Manager. Identity Manager ultimately redirects the user to the URL. This is the original URL requested by the user, before the user was redirected to OpenSSO for login.

Choose only one of the following procedures:

- "To Manually Modify a Deployed Login.jsp" on page 102
- "To Use the Sample `Login.jsp`" on page 102

▼ To Manually Modify a Deployed Login.jsp

1. **Use a text editor to open the file** *web-container-deploy-base*/opensso/config/auth/default/Login.jsp**.**

2. **Modify the file as in the following sample.**

   Changes are highlighted in bold.

```
...

<%
String ServiceURI = (String) viewBean.getDisplayFieldValue(viewBean.SERVICE_URI);
String encoded = "false";
String gotoURL = (String) viewBean.getValidatedInputURL(
request.getParameter("goto"), request.getParameter("encoded"), request);
String encodedQueryParams = (String) viewBean.getEncodedQueryParams(request);
if ((gotoURL != null) && (gotoURL.length() != 0)) {
encoded = "true";
}
String replaygotoURL = "";
String goToURL = request.getParameter("goto");
if(gotoURL != null && !gotoURL.equals("null") && (gotoURL.length() > 0)){
replaygotoURL = "&goto=" + goToURL;
}
System.out.println("replaygotoURL: " + replaygotoURL);%>

<link rel="stylesheet" href="<%= ServiceURI %>/css/styles.css" type="text/css" />
<script language="JavaScript" src="<%= ServiceURI %>/js/browserVersion.js"></script>
<script language="JavaScript" src="<%= ServiceURI %>/js/auth.js"></script>

...

<input type="hidden" name="goto" value="<%= gotoURL %>">
<input type="hidden" name="SunQueryParamsString" value="<%= encodedQueryParams %>">
<input type="hidden" name="encoded" value="<%= encoded %>">
<input type="hidden" name="plaingoto" value="<%= request.getParameter("goto") %>"></auth:form>
</jato:content>

...
```

▼ To Use the Sample Login.jsp

1. **Use a text editor to open the file** *opensso*/integrations/idm/jsps/Login.jsp **in the decompressed** opensso.zip **distribution.**

2 **Replace the deployed** */web-container-deploy-base*`/opensso/config/auth/default/Login.jsp` **with the modified**`Login.jsp`**.**

Choose one of the following options:

You can replace your existing `Login.jsp` with the sample `Login.jsp`. If you choose this option, the following will occur:

- You will lose any custom changes made to the existing `Login.jsp`.
- You will inherit changes that might have been previously made to the sample `Login.jsp` to incorporate requirements for other use cases related to the OpenSSO integration with Identity Manager.
- You must change the Identity Manager URL embedded in the sample to reflect the Identity Manager system URL of your architecture.

  You can search for the string `/idm` to locate the URLs.

Alternatively, you can manually make changes to the file. If you choose this option, run the `diff` command to view the differences between the two files.

3 **Remove the web containers temporary, compiled JSP to ensure that the changes made are picked up.**

For example, if using GlassFish, the temporary, compiled classes can be found under *glassfish-home*`/domains/`*your-domain*`/generated/`.

4 **Restart the OpenSSO Enterprise web container after making the changes.**

Modifying the Account Lockout Message Page

Modify the `user_inactive.jsp` file. This file notifies the user that the account is locked. Modify the page to include a redirect to an Identity Manager page the user can use to unlock the account. The `user_inactive.jsp` file forwards the following information to Identity Manager:

- The original URL requested by the user and defined as the value of the `goto` parameter.
- The user identifier defined as the value of the `accountId` parameter.

The Identity Manager URL used in the following sample refers to `anonResetPassword.jsp`. You might, however, direct the user to `questionLogin.jsp` the forgotten password page. If a user has accidentally locked an account it may be because of a forgotten password.

You can manually change the deployed `user_inactive.jsp` file, or you can use the sample `user_inactive.jsp` included with the `opensso.zip` download. Choose only one of the following procedures:

- "To Manually Modify the Account Lockout Message Page" on page 104
- "To Use the Sample Account Lockout Message Page" on page 105

▼ To Manually Modify the Account Lockout Message Page

1 **Use a text editor to open the file**
web-container-deploy-base/opensso/config/auth/default/user_inactive.jsp.

2 **Add the sections of code highlighted in bold in the following example:**

```
...
<head>
<title><jato:text name="htmlTitle_UserInactive" /></title>
<%
String ServiceURI = (String) viewBean.getDisplayFieldValue(viewBean.SERVICE_URI);
System.out.println("AccountId: " + request.getParameter("IDToken1"));
System.out.println("goto: " + request.getParameter("goto"));
System.out.println("plaingoto: " + request.getParameter("plaingoto"));
String accountId = request.getParameter("IDToken1");
String gotoURL = request.getParameter("plaingoto");
String redirectURL =
     "http://HostName.domainName.com:6480/idm/authutil/anonResetPassword.jsp";
         if(accountId != null){
redirectURL = redirectURL + "?accountId=" + accountId;
}

if(gotoURL != null && !gotoURL.equals("null") && (gotoURL.length() > 0)){
if(accountId == null){
redirectURL = redirectURL + "?goto=" + gotoURL;
}else{

redirectURL = redirectURL + "&goto=" + gotoURL;

}

}

%>


...
<div class="AlrtMsgTxt">
<auth:resBundle bundleName="amAuthUI" resourceKey="contactadmin" />

<!-- hyperlink -->

<jato:content name="ContentHref">
<p><a href="<%= redirectURL %>">Unlock Yourself</a></p>
</jato:content>
</div>
</td></tr></table></div></td>
```

```
</tr>
<tr>
<td> </td>
</tr>
<tr>
<td><img src="<%= ServiceURI %>/images/dot.gif"
width="1" height="33" alt="" /></td>
<td> </td>
</tr>
...
```

3 **Embedded in the JSP, you will see the URL to the Identity Manager page that allows the account unlock. Modify this URL as per your deployment.**

4 **Remove the web containers temporary, compiled JSP to ensure that the changes made are picked up.**

For example, if using GlassFish, the temporary, compiled classes can be found under *glassfish-home*/`domains`/*your-domain*/`generated/`.

5 **Restart the OpenSSO Enterprise web container after making the changes.**

▼ To Use the Sample Account Lockout Message Page

1 **Locate the file** *opensso*`/integrations/idm/jsps/user_inactive.jsp` **in the** `opensso.zip` **distribution.**

2 **Replace the deployed** */web-container-deploy-base*`/opensso/config/auth/default/user_inactive.jsp` **with the modified file.**

Choose only one of the following options:

You can replace your existing `user_inactive.jsp` with the sample `user_inactive.jsp`. If you choose this option, the following will occur:

- You will lose any custom changes made to the existing `user_inactive.jsp`.
- You will inherit changes that might have been previously made to the sample `user_inactive.jsp` to incorporate requirements for other use cases related to the OpenSSO integration with Identity Manager.
- You must change the Identity Manager URL embedded in the `user_inactive.jsp` to reflect your Identity Manager system URL. Search for the string `/idm` in the sample `Login.jsp` file to locate these URLs.

The Identity Manager URL used in the sample mentions the page `anonResetPassword.jsp`. However, you might want to direct the user to the forgotten-password page `questionLogin.jsp` instead. If the user has accidentally locked his account, he may not know his password to be able to reset it.

Alternatively, you can manually make the changes to the file. If you choose this option, run the `diff` command to view the differences between the two files.

3 **Remove the web containers temporary, compiled JSP to ensure that the changes made are picked up.**

For example, if using GlassFish, the temporary, compiled classes can be found under *glassfish-home*`/domains/`*your-domain*`/generated/`.

4 **Restart the OpenSSO Enterprise web container after making the changes.**

To Test the User-Initiated Account Unlock Configurations

Complete the following tests:

- "To Test Memory Account Unlock" on page 106
- "To Test Physical Account Unlock" on page 107

▼ To Test Memory Account Unlock

In this case, the user account is locked by maintaining a *state* of the account in memory. No information is stored in the user's LDAP profile to indicate the status of the account (active or inactive) when the account is locked in memory.

1 **Configure the password policy and assign the policy to the test user.**

See the *Sun Java System Directory Server Enterprise Edition 6.3 Administration Guide*for detailed instructions on configuring these settings.

2 **Access a resource protected by OpenSSO Enterprise to be redirected to the login page.**

3 **Log in to OpenSSO Enterprise using an incorrect password.**

Do this repeatedly until the account is locked and the error page is displayed. The account is locked based on the number of attempts configured in the password policy.

4 **Click the hyperlink on the page.**

You are redirected to an Identity Manager page on which are required to change your password. Note that the URL is the one configured in the `user_inactive.jsp`.

5 **Change your password.**

Identity Manager determines the account from the `accountID` parameter and changes the password on both OpenSSO Enterprise and on Identity Manager. After a successful modification, the user is redirected to the original URL defined in the `goto` parameter.

▼ To Test Physical Account Unlock

1 **In the test user's profile, set the value of the `inetuserstatus` attribute in the user data store to Inactive.**

2 **Access a resource protected by OpenSSO Enterprise to get redirected to the login page.**

3 **Log in to OpenSSO Enterprise.**

An error page is displayed informing you that the account has been locked.

4 **Click the hyperlink on the page.**

You are redirected to an Identity Manager page on which you are required to change your password. Note that the URL is the one configured in the `user_inactive.jsp`.

5 **Change your password.**

Identity Manager determines the account from the `accountID` parameter and will change the password on both OpenSSO and Identity Manager. After a successful modification, the user is redirected to the original URL defined in the `goto` parameter.

Configuring Identity Manager End-User Self-Registration

You can configure OpenSSO Enterprise so that end-users can create their own accounts when accessing Identity Manager for the first time. This is called self-registration or anonymous enrollment. The user provides the minimum required account details, and the user account is created on Identity Manager. This user account is automatically provisioned into OpenSSO Enterprise. A user account created in this manner is the most basic account with the minimum privileges available.

In the Identity Manager WAR, `/idm` is the base context of the deployment. An OpenSSO Enterprise policy agent must be present and protecting Identity Manager.

To configure OpenSSO, complete the following steps:

1. Configure OpenSSO Enterprise for end-user self-registration.
2. Modify the Identity Manager registration work flow.
3. Test the configurations.

The following figures illustrate the process flow for end-user self-registration.

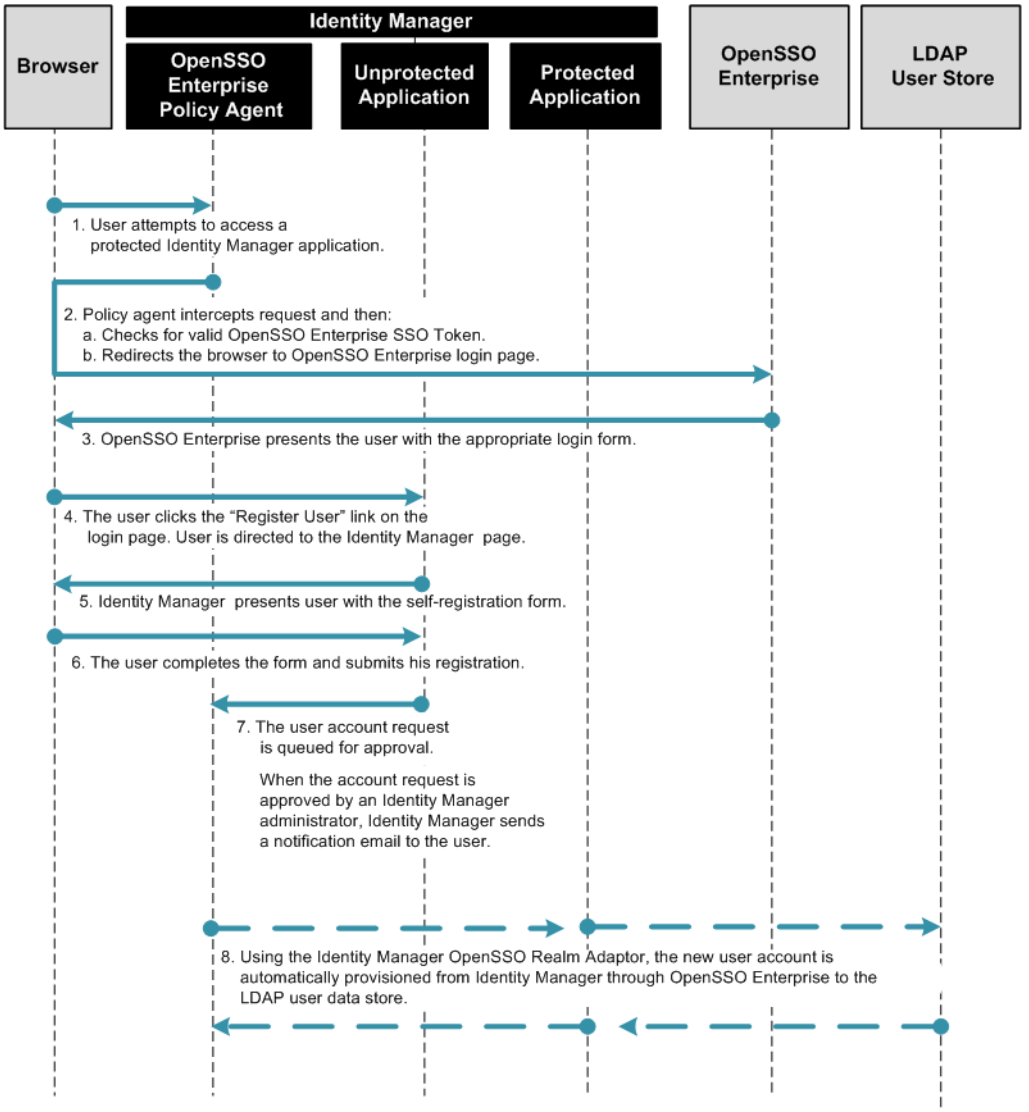


FIGURE 1–13 Process Flow for End-User Self-Registration

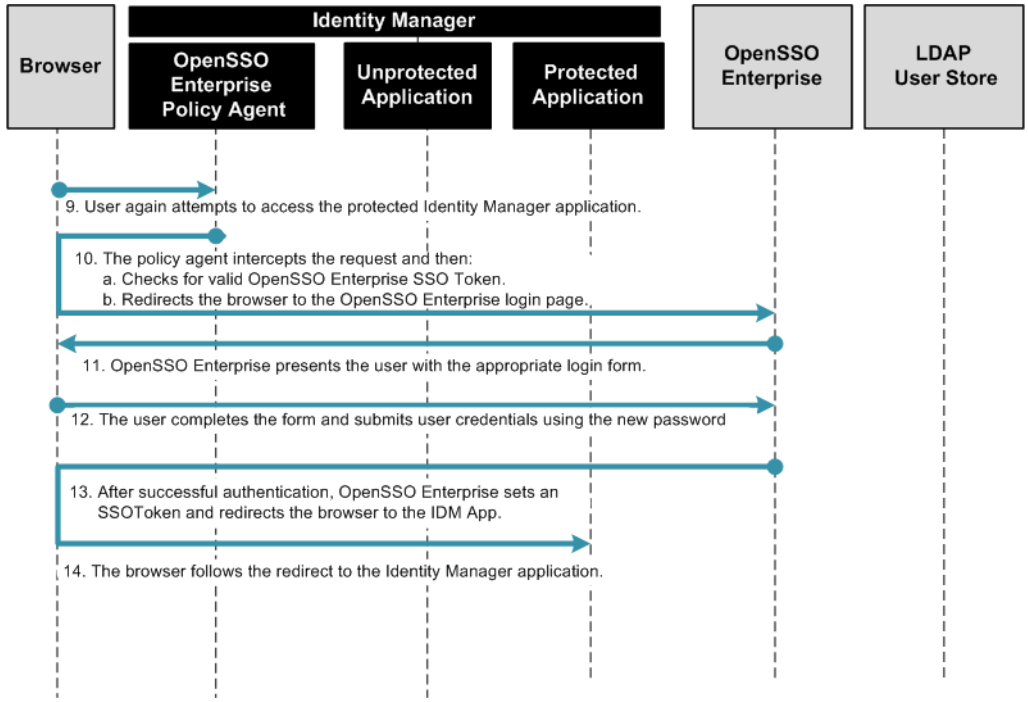


FIGURE 1–14 Process Flow for End-User Self-Registration (continued)

Configuring OpenSSO Enterprise for End-User Self-Registration

Complete the following steps:

- "To Define Identity Manager URLs as Not Enforced" on page 109
- "Modifying the OpenSSO Enterprise Login Page" on page 110

▼ To Define Identity Manager URLs as Not Enforced

1. **Log in to the OpenSSO Enterprise console as an administrator.**
2. **Click the Access Control tab.**
3. **Click the appropriate realm name and navigate to the Policy Agent Profile for the policy agent that protects Identity Manager.**
4. **Under the Policy Agent Profile, click the Application tab.**

5 **Add the following URIs to the "Not Enforced URIs" property.**

- `/idm/authutil/`
- `/idm/authutil/*`
- `/idm/authutil/*?*`

6 **Click Save.**

7 **Log out of OpenSSO Enterprise.**

Modifying the OpenSSO Enterprise Login Page

Modify the *opensso*`/config/auth/default/Login.jsp` page to include a Register User button. You can manually change the deployed `Login.jsp` file, or you can use the sample `Login.jsp` included with the `opensso.zip` download. Choose only one of the following procedures:

- "To Manually Modify a Deployed `Login.jsp`" on page 110
- "To Use the Sample `Login.jsp`" on page 113

▼ To Manually Modify a Deployed `Login.jsp`

1 **Use a text editor to open the file** */web-container-deploy-base*`/opensso/config/auth/default/Login.jsp.`

2 **Add the sections of code displayed in bold in the following example:**

```
...


<%

String ServiceURI = (String) viewBean.getDisplayFieldValue(viewBean.SERVICE_URI);
String encoded = "false";
String gotoURL = (String) viewBean.getValidatedInputURL(
request.getParameter("goto"), request.getParameter("encoded"), request);
String encodedQueryParams = (String) viewBean.getEncodedQueryParams(request);
if ((gotoURL != null) && (gotoURL.length() != 0)) {

encoded = "true";

}

String replaygotoURL = "";
String goToURL = request.getParameter("goto");
if(gotoURL != null && !gotoURL.equals("null") && (gotoURL.length() > 0)){

replaygotoURL = "&goto=" + goToURL;
```

```
}

System.out.println("replaygotoURL: " + replaygotoURL);
%>


<link rel="stylesheet" href="<%= ServiceURI %>/css/styles.css" type="text/css" />
<script language="JavaScript" src="<%= ServiceURI %>/js/browserVersion.js"></script>
<script language="JavaScript" src="<%= ServiceURI %>/js/auth.js"></script>


...


function LoginSubmit(value) {
aggSubmit();
var hiddenFrm = document.forms['Login'];

if (hiddenFrm != null) {
hiddenFrm.elements['IDButton'].value = value;

if (this.submitted) {
alert("The request is currently being processed");

}

else {

this.submitted = true;
hiddenFrm.submit();

}

}

}


function RegisterUser() {
window.location = "http://HostName.DomainNamecom:6480/idm/authutil/
      anonEnrollment.jsp?id=endUserAnonymousEnrollment<%=replaygotoURL%>";

}


...
```

```
<jato:content name="hasNoButton">

<tr>

<td><img src="<%= ServiceURI %>/images/dot.gif"
width="1" height="15" alt="" /></td>

<td>
<table border=0 cellpadding=0 cellspacing=0>
<script language="javascript">
markupButton(
'<jato:text name="lblSubmit" />',
"javascript:LoginSubmit('<jato:text name="lblSubmit" />')");
</script>

<script language="javascript">
markupButton(
'Register User',
"javascript:RegisterUser()");
</script>
</table>
</td>

</tr>

<!-- end of hasNoButton -->

</jato:content>


...


<input type="hidden" name="goto" value="<%= gotoURL %>">
<input type="hidden" name="SunQueryParamsString" value="<%= encodedQueryParams %>">
<input type="hidden" name="encoded" value="<%= encoded %>">
<input type="hidden" name="plaingoto" value="<%= request.getParameter("goto") %>">
</auth:form>

</jato:content>


...
```

3. **Remove the web container's temporary, compiled JSP to ensure that the changes made are picked up.**

   For example, if using GlassFish, the temporary, compiled classes is under *glassfish-home*/`domains`/*your-domain*/`generated/`.

4. **Restart the OpenSSO Enterprise web container after making the changes.**

▼ To Use the Sample `Login.jsp`

1. **Locate the file** *opensso*/`integrations/idm/jsps/Login.jsp` **in the** `opensso.zip` **distribution.**

2. **Replace the deployed** */web-container-deploy-base*/`opensso/config/auth/default/Login.jsp` **with the modified** `Login.jsp`**.**

   Choose one of the following options:

   You can you replace your existing `Login.jsp` with the sample `Login.jsp`. If you choose this option, the following will occur:

   - You will lose any custom changes made to the existing `Login.jsp`.
   - You will inherit changes that might have been previously made to the sample `Login.jsp` to incorporate requirements for other use cases related to the OpenSSO integration with Identity Manager.
   - You must change the Identity Manager URL embedded in the `Login.jsp` to reflect your Identity Manager system URL. Search for the string `/idm` in the sample `Login.jsp` file to locate these URLs.

   Alternatively, you can manually make the changes to the file. If you choose this option, run the `diff`command to view the differences between the two files.

3. **Remove the web container's temporary, compiled JSP to ensure that the changes made are picked up.**

   For example, if using GlassFish, the temporary, compiled classes are under `glassfish-home/domains/your-domain/generated/`.

4. **Restart the OpenSSO Enterprise web container after making the changes.**

Modifying the Identity Manager Registration Work Flow

You can use the Identity Manager plug-in for NetBeans IDE or, you can use the Identity Manager Debug Pages. Choose only one of the following procedures:

- "To Change the Registration Work Flow Using NetBeans IDE" on page 114
- "To Use the Identity Manager Debug Pages" on page 116

▼ To Change the Registration Work Flow Using NetBeans IDE

Before You Begin

- NetBeans IDE must be already downloaded and installed.

  See downloaded and installed NetBeans IDE for more information.
- The Identity Manager Plug-In for NetBeans must be already downloaded and installed.

  See downloaded and installed the Identity Manager Plug-in for NetBeans for more information.

1. **Create (or open) an Identity Manager Project in NetBeans.**

   You can use this procedure with either integrated or remote NetBeans projects. Use the online help available in NetBeans to create the Identity Manager project if necessary. See the Identity Manager IDE website for more information.

2. **From the NetBeans Project Window, right-click on the Custom Identity Manager Objects Node and select IDM > Open Object.**

3. **In the Open Object dialog box, enter the object name "End User Anonymous Enrollment," and click OK.**

4. **Right-click on the file in the Project Window, and choose IDM > Clone Object(s) to clone the object for safe keeping.**

5. **Name the new object** *End User Anonymous Enrollment Orig***.**

6. **Click on the tab in the Editor window containing the file "End User Anonymous Enrollment" work flow.**

   This will put the file in focus.

7. **Expand the tree in the Navigator Window to locate the Activity Assimilate User View.**

8. **Add the OpenSSO Enterprise resource to the map of options for the "assimilate" invocation.**

   The changes are displayed in bold in the following example:

```
<Activity id='2' name='Assimilate User View'>
<Action id='0' hidden='true'>

<expression>
<invoke name='assimilate'>
<ref>view</ref>
<map>
```

```
<s>waveset.accountId</s>
<ref>accountId</ref>

<s>to</s>
<ref>accountId</ref>

<s>global.firstname</s>
<ref>firstname</ref>

<s>global.lastname</s>
<ref>lastname</ref>

<s>global.fullname</s>

<concat>
<ref>firstname</ref>
<s> </s>
<ref>lastname</ref>
</concat>

<s>fullname</s>

<concat>
<ref>firstname</ref>

<s> </s>
<ref>lastname</ref>
</concat>

<s>accounts[Lighthouse].fullname</s>

<concat>
<ref>firstname</ref>
<s> </s>
<ref>lastname</ref>
</concat>

<s>global.email</s>
<ref>email</ref>

<s>global.idmManager</s>
<ref>idmManager</ref>

<s>accounts[Lighthouse].idmManager</s>
<ref>idmManager</ref>

<s>waveset.password</s>
```

```
<ref>userPassword</ref>

<s>waveset.organization</s>
<ref>organization</ref>

<s>waveset.resources</s>

<list>
<s>OpenSSO</s>
</list>

</map>

</invoke>

</expression>

<Variable name='view'>
<ref>user</ref>
</Variable>

<Return from='view' to='user'/>
<Return from='WF_ACTION_ERROR' to='error'/>
</Action>

<Transition to='Create Process View'/>
</Activity>
```

The name of the OpenSSO resource (OpenSSO in `self_registration_idm_anon_enroll.html`) is the name assigned when the resource was created. To verify the name, navigate to the "Resources | List Resources" tab in the Identity Manager administration console and expand the "Sun Access Manager Realm" branch.

9. **Save the changes.**

10. **Right-click on the file and select IDM > Upload Object(s) to upload the file.**

▼ To Use the Identity Manager Debug Pages

1. **Log in to the Identity Manager console as an administrator.**

2. **Go to the debug URL at** *protocol*`://`*IDM-host-machine*`:`*port*`/idm/debug`**.**

3. **Select the object Task Definition in the list next to the List Objects button.**

4. **Click the List Objects button.**

5 Search for the object "End User Anonymous Enrollment.

Export the existing definition to back it up. Then click Edit.

6 Add the OpenSSO Enterprise resource to the Activity "Assimilate User View."

In the following example, the changes are displayed in bold:

```
<Activity id='2' name='Assimilate User View'>
<Action id='0' hidden='true'>

<expression>
<invoke name='assimilate'>
<ref>view</ref>
<map>

<s>waveset.accountId</s>
<ref>accountId</ref>

<s>to</s>
<ref>accountId</ref>

<s>global.firstname</s>
<ref>firstname</ref>

<s>global.lastname</s>
<ref>lastname</ref>

<s>global.fullname</s>

<concat>
<ref>firstname</ref>
<s> </s>
<ref>lastname</ref>
</concat>

<s>fullname</s>

<concat>
<ref>firstname</ref>

<s> </s>
<ref>lastname</ref>
</concat>

<s>accounts[Lighthouse].fullname</s>

<concat>
<ref>firstname</ref>
```

```
<s> </s>
<ref>lastname</ref>
</concat>

<s>global.email</s>
<ref>email</ref>

<s>global.idmManager</s>
<ref>idmManager</ref>

<s>accounts[Lighthouse].idmManager</s>
<ref>idmManager</ref>

<s>waveset.password</s>
<ref>userPassword</ref>

<s>waveset.organization</s>
<ref>organization</ref>

<s>waveset.resources</s>
<list>
<s>OpenSSO</s>
</list>

</map>

</invoke>

</expression>

<Variable name='view'>
<ref>user</ref>
</Variable>

<Return from='view' to='user'/>
<Return from='WF_ACTION_ERROR' to='error'/>
</Action>

<Transition to='Create Process View'/>
</Activity>
```

The name of the OpenSSO Enterprise resource (OpenSSO in `self_registration_idm_anon_enroll.html`) is the name assigned when the resource was created. To verify the name, navigate to the "Resources | List Resources" tab in the Identity Manager administration console and expand the "Sun Access Manager Realm" branch.

7 Log out of the console.

Testing Configurations for End-User Self-Registration

Complete the following tests in this exact order:

1. Test end-user self-registration.
2. Test approval of the new user account.
3. Verify Provisioning of the new user account.
4. Verify activation of the new user account.

▼ To Test End-User Self-Registration

1 Go to the OpenSSO Enterprise login URL at
protocol`://`*OpenSSO-HostName*`:`*port*`/opensso/UI/Login`**.**

2 Click the Register User button to register a test user.

3 Go through the registration process, and click Register to save changes.

A message is displayed indicating the registration request is being processed.

▼ To Test Approval of New User Account

1 Log in to the Identity Manager console as an administrator.

The Create User task is displayed as pending task.

2 Navigate to the Work Items > Approvals tab.

3 Choose the provisioning task for the new user-id and click the Approve button.

4 Confirm the approval.

5 Log out of the Identity Manager console.

▼ Verify Provisioning Of New User Account

1 Log in to the OpenSSO Enterprise console as an administrator.

2 Navigate to the Access Control > Realm > Subjects tab.

The approved user is displayed indicating that the profile was successfully registered and provisioned.

▼ Verify Activation Of New User Account

1. **Go to the OpenSSO Enterprise login URL at** *protocol*`://`*OpenSSO-HostName*`:`*port*`/opensso/UI/Login` **and login as the new user.**

2. **Log out of OpenSSO Enterprise.**

Troubleshooting Identity Manager Integration

To troubleshoot problems with any procedure in this chapter, try the following:

▼ To Enable Trace in Identity Manager

1. **Login to the Identity Manager debug interface:**

   Go to the following URL:

   `http://`*ApplicationServerHost:Port*`/idm/debug`

   Provide the following credentials:

   UserName: configurator

   Password: configurator

2. **Click the Show Trace button.**

3. **In the Edit Trace Configuration window, mark Trace Enabled checkbox.**

4. **Add the following classes, each with a trace level of 4:**

   - `com.waveset.adapter.ResourceAdapterBase`
   - `com.waveset.adapter.SunAccessManagerRealmResourceAdapter`

5. **Note the default trace file location and name in the property** `Trace File`**.**

   You can customize the trace file location or name.

6. **Click Save.**

7 Log out of the Identity Manager debug interface.

To Inspect Log Files

For the installation described in this chapter, these are the log locations:

- Application Server logs

  `/opt/SUNWappserver91/domains/domain1/logs`

  In the file:

  `/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm/config/Waveset.properties`

  enable this property:

  `exception.trace=true`

  After you enable this property, the exception trace logging will be logged to the trace log that is mentioned in the section above. However, the property does not take effect until the GlassFish server is restarted.
- OpenSSO Enterprise Client SDK debug logs

  `/opt/SUNWappserver91/domains/domain1/logs/am_debug`
- OpenSSO Enterprise Server debug logs

  `/opt/fam80-idm80/opensso/debug`
- OpenSSO Enterprise Policy Agent debug logs

  `/opt/j2ee_agents/appserver_v9_agent/Agent_001/logs/debug`

To View or Change Identity Manager System Settings

Using the Identity Manager Debug Administrator Interface

The following information refers to the examples in this book. For detailed information about using the Debug Administrator Interface, see the *Identity Manager Tuning, Troubleshooting, and Error Messages* at `http://docs.sun.com/app/docs/coll/1514.5`.

URL: `http://`*ApplicationServerHost:Port*`/idm/debug`

User Name: configurator

Password: configurator

The following Identity Manager objects were created/modified in this document:

Object Type	Name or ID
Resource	SunOpenSSORealm
LoginModGroup	Sun OpenSSO Realm
LoginApp	UI_LOGIN_CONFIG_DISPLAY_NAME_USER_INTERFACE
	UI_LOGIN_CONFIG_DISPLAY_NAME_ADMIN_INTERFACE

You can view, edit, or export Identity Manager objects in xml format, or through the Get Object and List Objects buttons.

Using the Identity Manager IDE Interface

Before you begin, disable the Policy Agent on the Identity Manager server. This enables NetBeans to connect to the Identity Manager server.

1. Download NetBeans IDE 6.0. Web & Java EE edition.
2. Download the Identity Manager IDE plug-in .
3. Follow the instructions for installing the IDE Plug-in in NetBeans. Go to the following URL:

   https://identitymanageride.dev.java.net/netbeans-setup.html

   A new IDM menu item appears in the NetBeans menubar.

▼ To Inspect an Identity Manager Object

1 Choose the "Custom Identity Manager Objects" in the Project window.

2 In the NetBeans menu, choose IDM / Open Object.

3 In the Open Object page, do this:

```
Object Name: *
Object Type: <select an object, ex:Resource>
```

You will see that the list of Matching Objects gets populated with the objects of the selected type. In this example, if you select SunAccessManagerRealm, and OK, the object-definition will be downloaded to the project.

To Update an Identity Manager Object

To modify the object, and upload the changed object, right-click on the object in the Project window and select Identity Manager / Upload Object. The following Identity Manager objects were created or modified in this chapter:

Resource	SunAccessManagerRealm
`LoginModGroup`	Sun Access Manager Realm
`LoginApp`	User Interface
	Administrator Interface

To Consult Forums and Mailing Lists

- Sun Identity Manager Forum questions

  `identityManager-technical-questions-ext@sun.com`
- Sun Identity Manager IDE Plug-in related questions

  `https://identitymanageride.dev.java.net/servlets/ProjectMailingListList`

  `https://identitymanageride.dev.java.net/servlets/ProjectForumView`
- Sun OpenSSO Enterprise / Policy Agent questions

  `amfm-technical-ext@sun.com`
- OpenSSO questions

  `https://opensso.dev.java.net/servlets/ProjectForumView`

Sample Output

EXAMPLE 1–2 Sample Output from `lh` Log Program

```
Import init.xml Identity Manager configuration
Getting new session...

 * * * * * * *
Importing file '/opt/SUNWappserver91/domains/domain1/applications/
  j2ee-modules/idm/sample/init.xml':

Including file 'sample/sysconfig.xml'.
PKCS#5 encryption set. Server encryption keys re-encrypted.
Restored Configuration:System Configuration
Including file 'sample/certdata.xml'.
Restored UserForm:CertificateDataMainForm
Restored UserForm:CertificateDataAddCertForm
Restored UserForm:CertificateDataForm
Including file 'sample/changelogconfig.xml'.
Restored Configuration:ChangeLog Configuration
Including file 'sample/admingroups.xml'.
Added Configuration:AuthorizationTypes
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored AdminGroup:Admin
Restored AdminGroup:List Admin Roles
Restored AdminGroup:Connect Admin Roles
Restored AdminGroup:Admin Role Administrator
Restored AdminGroup:Approver Administrator
Restored AdminGroup:Organization Approver
Restored AdminGroup:Role Approver
Restored AdminGroup:Resource Approver
Restored AdminGroup:List Capabilities
Restored AdminGroup:Connect Capabilities
Restored AdminGroup:Capability Administrator
Restored AdminGroup:EndUser
Restored AdminGroup:End User Administrator
Restored AdminGroup:Import/Export Administrator
Restored AdminGroup:License Administrator
Restored AdminGroup:Login Administrator
Restored AdminGroup:List Organizations
Restored AdminGroup:Connect Organizations
Restored AdminGroup:Organization Administrator
Restored AdminGroup:List Policies
Restored AdminGroup:Connect Policies
Restored AdminGroup:Policy Administrator
Restored AdminGroup:Reconcile Administrator
Restored AdminGroup:Reconcile Request Administrator
Restored AdminGroup:View Meta View
Restored AdminGroup:Meta View Administrator
Restored AdminGroup:Configure Audit
Restored AdminGroup:Configure Certificates
Restored AdminGroup:Run Report Refs
Restored AdminGroup:Run Admin Report
Restored AdminGroup:Admin Report Administrator
Restored AdminGroup:Run Audit Report
Restored AdminGroup:Audit Report Administrator
Restored AdminGroup:Run Reconcile Report
Restored AdminGroup:Reconcile Report Administrator
Restored AdminGroup:Run Resource Report
Restored AdminGroup:Resource Report Administrator
Restored AdminGroup:Run Risk Analysis
Restored AdminGroup:Risk Analysis Administrator
Restored AdminGroup:Run Role Report
Restored AdminGroup:Role Report Administrator
Restored AdminGroup:Run User Report
Restored AdminGroup:User Report Administrator
Restored AdminGroup:Run Task Report
Restored AdminGroup:Task Report Administrator
Restored AdminGroup:Report Administrator
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored AdminGroup:List Resources
Restored AdminGroup:Connect Resources
Restored AdminGroup:Access Resource UI
Restored AdminGroup:Resource Administrator
Restored AdminGroup:Bulk Resource Administrator
Restored AdminGroup:Resource Object Administrator
Restored AdminGroup:Change Resource Password Administrator
Restored AdminGroup:Bulk Change Resource Password Administrator
Restored AdminGroup:Reset Resource Password Administrator
Restored AdminGroup:Bulk Reset Resource Password Administrator
Restored AdminGroup:Resource Password Administrator
Restored AdminGroup:Bulk Resource Password Administrator
Restored AdminGroup:Change Active Sync Resource Administrator
Restored AdminGroup:Control Active Sync Resource Administrator
Restored AdminGroup:List Resource Groups
Restored AdminGroup:Connect Resource Groups
Restored AdminGroup:Resource Group Administrator
Restored AdminGroup:List Roles
Restored AdminGroup:Connect Roles
Restored AdminGroup:Role Administrator
Restored AdminGroup:List Rules
Restored AdminGroup:Connect Rules
Restored AdminGroup:Connect Capabilities Rules
Restored AdminGroup:Connect Login Constraint Rules
Restored AdminGroup:Connect Controlled Organizations Rules
Restored AdminGroup:Connect Login Correlation Rules
Restored AdminGroup:Connect New User Name Rules
Restored AdminGroup:List User Members Rules
Restored AdminGroup:Connect User Members Rules
Restored AdminGroup:Connect Excluded Accounts Rules
Restored AdminGroup:Connect User Is Assigned Admin Role Rules
Restored AdminGroup:Connect SPE User Is Assigned Admin Role Rules
Restored AdminGroup:Connect SPE Users Search Context Rules
Restored AdminGroup:Connect SPE Users Search Filter Rules
Restored AdminGroup:Connect SPE Users After Search Filter Rules
Restored AdminGroup:Connect Capabilities On SPE User Rules
Restored AdminGroup:View UserUIConfig
Restored AdminGroup:Access User UI
Restored AdminGroup:List User Refs
Restored AdminGroup:User Refs
Restored AdminGroup:List Users
Restored AdminGroup:View User
Restored AdminGroup:Connect Users
Restored AdminGroup:Create User
Restored AdminGroup:Bulk Create User
Restored AdminGroup:Update User
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored AdminGroup:Bulk Update User
Restored AdminGroup:Delete IDM User
Restored AdminGroup:Deprovision User
Restored AdminGroup:Unlink User
Restored AdminGroup:Unassign User
Restored AdminGroup:Delete User
Restored AdminGroup:Bulk Delete IDM User
Restored AdminGroup:Bulk Unassign User
Restored AdminGroup:Bulk Unlink User
Restored AdminGroup:Bulk Deprovision User
Restored AdminGroup:Bulk Delete User
Restored AdminGroup:Enable User
Restored AdminGroup:Bulk Enable User
Restored AdminGroup:Disable User
Restored AdminGroup:Bulk Disable User
Restored AdminGroup:Unlock User
Restored AdminGroup:Rename User
Restored AdminGroup:Change Password Administrator
Restored AdminGroup:Reset Password Administrator
Restored AdminGroup:Password Administrator
Restored AdminGroup:Change Password Administrator (Verification Required)
Restored AdminGroup:Reset Password Administrator (Verification Required)
Restored AdminGroup:Password Administrator (Verification Required)
Restored AdminGroup:Import User
Restored AdminGroup:User Account Administrator
Restored AdminGroup:Bulk User Account Administrator
Restored AdminGroup:Change User Account Administrator
Restored AdminGroup:Bulk Change User Account Administrator
Restored AdminGroup:Assign User Capabilities
Restored AdminGroup:SPML Access
Restored AdminGroup:Account Administrator
Restored AdminGroup:Bulk Account Administrator
Restored AdminGroup:Change Account Administrator
Restored AdminGroup:Bulk Change Account Administrator
Restored AdminGroup:List UserForms
Restored AdminGroup:Connect UserForms
Restored AdminGroup:Security Administrator
Restored AdminGroup:Waveset Administrator
Restored AdminGroup:Remedy Integration Administrator
Restored AdminGroup:Auditor View User
Restored AdminGroup:List Audit Policies
Restored AdminGroup:Connect Audit Policies
Restored AdminGroup:Assign User Audit Policies
Restored AdminGroup:Assign Organization Audit Policies
Restored AdminGroup:Assign Audit Policies
Restored AdminGroup:Audit Policy Administrator
```

EXAMPLE 1–2 Sample Output from lh Log Program *(Continued)*

```
Restored AdminGroup:Run Audited Attribute Report
Restored AdminGroup:Audited Attribute Report Administrator
Restored AdminGroup:Run User Access Report
Restored AdminGroup:User Access Report Administrator
Restored AdminGroup:Run AuditLog Report
Restored AdminGroup:AuditLog Report Administrator
Restored AdminGroup:Run Policy Summary Report
Restored AdminGroup:Policy Summary Report Report Administrator
Restored AdminGroup:Run Audit Policy Scan Report
Restored AdminGroup:Audit Policy Scan Report Administrator
Restored AdminGroup:Run AuditPolicy Violation History
Restored AdminGroup:AuditPolicy Violation History Administrator
Restored AdminGroup:Run Organization Violation History
Restored AdminGroup:Organization Violation History Administrator
Restored AdminGroup:Run Resource Violation History
Restored AdminGroup:Resource Violation History Administrator
Restored AdminGroup:Run Violation Summary Report
Restored AdminGroup:Violation Summary Report Administrator
Restored AdminGroup:Run Separation of Duties Report
Restored AdminGroup:Separation of Duties Report Administrator
Restored AdminGroup:Run Access Review Summary Report
Restored AdminGroup:Access Review Summary Report Administrator
Restored AdminGroup:Run Access Review Detail Report
Restored AdminGroup:Access Review Detail Report Administrator
Restored AdminGroup:Run Auditor Report
Restored AdminGroup:Auditor Report Administrator
Restored AdminGroup:Auditor Remediator
Restored AdminGroup:Auditor Attestor
Restored AdminGroup:Auditor Access Scan Administrator
Restored AdminGroup:Auditor Periodic Access Review Administrator
Restored AdminGroup:Auditor Administrator
Including file 'sample/adminroles.xml'.
Restored AdminRole:User
Including file 'sample/admins.xml'.
Restored User:Configurator
Restored User:Administrator
Restored User:Reset
Including file 'sample/enduserobjects.xml'.
Restored ObjectGroup:End User
Restored Rule:End User Controlled Organizations
Including file 'sample/loginconfig.xml'.
Restored LoginConfig:Waveset Login Configuration
Added LoginModGroup:Default Lighthouse Id/Pwd Login Module Group
Added LoginModGroup:Default Lighthouse Id/Questions Login Module Group
Added LoginModGroup:Default Lighthouse X509 Cert Login Module Group
Added LoginApp:Administrator Interface
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Added LoginApp:BPE
Added LoginApp:User Interface
Added LoginApp:Secondary Authentication Interface
Added LoginApp:Command Line Interface
Added LoginApp:DefaultUser
Added LoginApp:IVR Interface
Including file 'sample/auditconfig.xml'.
Preserving object Configuration #ID#Configuration:AuditConfiguration
Restored Configuration:Audit Configuration
Restored Configuration:WorkflowDetailsRecordForm
Restored Configuration:LogRecordForm
Restored UserForm:AuditMainForm
Restored UserForm:AuditGroupEditForm
Restored UserForm:AuditPublisherForm
Restored Configuration:AuditConfigForm
Including file 'sample/remedyconfig.xml'.
Added Configuration:Remedy Workflow Process
Added TaskDefinition:Test Remedy Template Workflow
Restored Configuration:RemedyTemplateForm
Including file 'sample/serverkeys.xml'.
Restored TaskDefinition:Server Encryption
Including file 'sample/metaView.xml'.
Restored MetaView:User Meta View
Including file 'sample/auditorforms.xml'.
Preserving object Rule #ID#Rule:ViolationPriority
Preserving object Rule #ID#Rule:ViolationSeverity
Restored UserForm:AuditPolicyLibrary
Restored UserForm:AuditorFormLibrary
Added UserForm:Audit Policy List
Added UserForm:Audit Policy Delete Confirmation Form
Restored UserForm:Audit Policy Form
Restored UserForm:Update Audit Policy Form
Added UserForm:Remediation Library
Restored UserForm:Bulk Remediation
Restored UserForm:Sign Bulk Remediation
Restored Rule:ViolationPriority
Restored Rule:ViolationSeverity
Restored UserForm:Remediation List
Restored Configuration:AuditorOrgForm
Restored UserForm:Violation Detail Form
Restored UserForm:Compliance Violation Summary Form
Restored UserForm:Conflict Violation Details Form
Restored UserForm:Auditor Tab
Restored UserForm:Remediation Form
Restored Configuration:User Extended Attributes
Restored AttributeDefinition:accountId
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored AttributeDefinition:password
Restored AttributeDefinition:fullname
Restored AttributeDefinition:email
Restored AttributeDefinition:lastname
Restored AttributeDefinition:firstname
Restored UserForm:Example Form
Restored TaskDefinition:Password Expiration
Restored Configuration:SoapConfig
Restored ResourceAction:Example Login Action
Restored ResourceAction:Example Logoff Action
Restored TaskDefinition:LoadTask
Restored TaskDefinition:ImportTask
Restored Configuration:Reconcile Configuration
Including file 'sample/userSearchDefaults.xml'.
Added UserForm:User Search Defaults
Including file 'sample/userActionsConfig.xml'.
Restored Configuration:User Actions Configuration
Including file 'sample/findObjectsDefaults.xml'.
Restored Configuration:Find Objects Defaults
Including file 'sample/approvalforms.xml'.
Including file 'sample/AdminDashboard.xml'.
Restored UserForm:Admin Dashboard
Including file 'sample/otherWorkItems.xml'.
Restored UserForm:Other Work Item List
Including file 'sample/emailTemplates.xml'.
Restored EmailTemplate:Password Reset
Restored EmailTemplate:Temporary Password Reset
Restored EmailTemplate:Request Resource
Restored EmailTemplate:Retry Notification
Restored EmailTemplate:Risk Analysis
Restored EmailTemplate:Report
Restored EmailTemplate:User ID Recovery
Including file 'sample/policy.xml'.
Preserving object Policy #ID#PasswordPolicy
Preserving object Policy #ID#AccountIdPolicy
Preserving object Policy #ID#Windows2000PasswordPolicy
Preserving object Policy #ID#Policy.DefaultLighthouseAccount
Preserving object Policy #ID#Policy.LdapDnAccount
Restored Policy:Password Policy
Restored Policy:AccountId Policy
Restored Policy:Windows 2000 Password Policy
Restored Policy:Default Identity Manager Account Policy
Restored Policy:LDAP DN Account Policy
Including file 'sample/deferredtasks.xml'.
Added TaskDefinition:Deferred Task Scanner
Added TaskDefinition:Start Date
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Added TaskDefinition:Sunset Date
Including file 'sample/formlib.xml'.
Added UserForm:Default User Library
Added UserForm:Password Library
Added UserForm:Questions Library
Added UserForm:Account Summary Library
Added UserForm:Account Link Library
Added UserForm:Administrator Library
Added UserForm:User Library
Added UserForm:Organization Library
Added UserForm:Locale Selection Library
Added UserForm:Approval Library
Added UserForm:Scalable Selection Library
Added Rule:Role Names
Added Rule:Organization Names
Added Rule:Resource Names
Including file 'sample/treetableLibrary.xml'.
Added Configuration:Tree Table Library
Including file 'sample/forms.xml'.
Not saving object UserForm #ID#UserForm:DefaultUserForm: not found
Not saving object UserForm #ID#UserForm:RenameUserForm: not found
Not saving object UserForm #ID#UserForm:ChangeUserPasswordForm: not found
Not saving object UserForm #ID#UserForm:ReprovisionForm: not found
Not saving object UserForm #ID#UserForm:DeprovisionForm: not found
Not saving object UserForm #ID#UserForm:DisableForm: not found
Not saving object UserForm #ID#UserForm:EnableForm: not found
Not saving object UserForm #ID#UserForm:UnlockForm: not found
Not saving object UserForm #ID#UserForm:ResetUserPasswordForm: not found
Not saving object UserForm #ID#UserForm:UserFormLibrary: not found
Not saving object Configuration #ID#Configuration:UserFormLibrary: not found
Restored UserForm:Default User Form
Added UserForm:Default View User Form
Restored UserForm:Select Accounts Form
Added UserForm:Tabbed User Form
Added UserForm:Tabbed View User Form
Added UserForm:Wizard User Form
Added UserForm:Wizard View User Form
Restored UserForm:Reprovision Form
Restored UserForm:Deprovision Form
Restored UserForm:Disable Form
Restored UserForm:Enable Form
Restored UserForm:Unlock Form
Restored UserForm:Change My Password Form
Restored UserForm:Change User Password Form
Restored UserForm:Reset User Password Form
Restored UserForm:Rename User Form
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored UserForm:User Form Library
Restored Configuration:User Form Library
Restored Configuration:Role Rename Form
Restored Configuration:Role Form
Restored Configuration:Resource Group Rename Form
Restored Configuration:Application Form
Added UserForm:List Resource Groups
Restored Configuration:Lighthouse Policy Rename Form
Restored Configuration:Lighthouse Policy Form
Restored UserForm:Change User Capabilities Form
Restored UserForm:Change User Audit Policies Form
Restored UserForm:Change Organization Audit Policies Form
Restored Configuration:Organization Rename Form
Restored Configuration:Organization Form
Restored Configuration:Directory Junction Form
Restored Configuration:Virtual Organization Form
Restored Configuration:Virtual Organization Refresh Form
Restored Configuration:Capability Form
Restored Configuration:Admin Role Form
Restored UserForm:Managed Resources Form
Restored UserForm:Form and Process Mappings Form
Restored UserForm:Empty Form
Restored Configuration:Find Objects Form
Restored Configuration:Find Objects Results Form
Restored UserForm:User Interface Configuration Form
Restored UserForm:Delegate WorkItems
Restored UserForm:Lookup UserId
Including file 'sample/configforms.xml'.
Added UserForm:Confirm Deletes
Added UserForm:List Rules
Added UserForm:Expression Editor
Added UserForm:Edit Rule
Added UserForm:Edit Argument
Added UserForm:List Forms
Added UserForm:Edit Form
Added UserForm:Display Component Fields
Added UserForm:Edit Field
Including file 'sample/sysforms.xml'.
Not saving object UserForm #ID#UserForm:TaskScheduleForm: not found
Restored UserForm:Work Item Confirmation
Restored UserForm:Work Item List
Restored UserForm:Task Schedule Form
Restored Configuration:SyslogRecordForm
Including file 'sample/userFind.xml'.
Not saving object UserForm #ID#UserForm:UserSearchLibrary: not found
Not saving object UserForm #ID#UserForm:AdvancedFindUserForm: not found
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Not saving object UserForm #ID#UserForm:FindUserForm: not found
Not saving object UserForm #ID#UserForm:FindUserResultsForm: not found
Not saving object UserForm #ID#UserForm:UserSelectionForm: not found
Not saving object UserForm #ID#UserForm:FindAccountOwnerForm: not found
Not saving object UserForm #ID#UserForm:AccountOwnerSelectionForm: not found
Restored UserForm:User Search Library
Restored UserForm:Advanced Find User Form
Restored UserForm:Find User Form
Restored UserForm:Find User Results Form
Restored UserForm:User Selection Form
Restored UserForm:Find Account Owner Form
Restored UserForm:Account Owner Selection Form
Including file 'sample/userListForm.xml'.
Restored UserForm:User List Library
Restored UserForm:User List Form
Including file 'sample/accountFind.xml'.
Not saving object UserForm #ID#UserForm:AccountSearchLibrary: not found
Not saving object UserForm #ID#UserForm:FindAccountForm: not found
Not saving object UserForm #ID#UserForm:FindAccountResultsForm: not found
Restored UserForm:Account Search Library
Restored UserForm:Find Account Form
Restored UserForm:Find Account Results Form
Including file 'sample/loginForms.xml'.
Added UserForm:Login App List
Restored Configuration:Login App Rename Form
Restored Configuration:Login App View Form
Added UserForm:Login Mod Group List
Restored Configuration:Login Mod Group Rename Form
Restored Configuration:Login Mod Group View Form
Restored Configuration:Login Module Edit Form
Added Rule:Sample On Local Network
Including file 'sample/metaViewForms.xml'.
Restored UserForm:Edit Meta View
Restored UserForm:Edit Meta Events
Restored UserForm:Meta View Library
Restored UserForm:List Builder Library
Restored UserForm:Edit Meta View Attribute
Restored UserForm:Edit Meta View Attribute Target
Restored UserForm:Edit Meta Event
Restored UserForm:Edit Meta Event Response
Restored UserForm:Confirm Meta View Attribute Deletes
Restored UserForm:Import Meta View
Restored UserForm:Configure MetaView from Resource Changes
Restored UserForm:Continue To Meta View From Resource
Restored UserForm:Meta View Password Generation
Including file 'sample/changelogconfigForms.xml'.
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored UserForm:Edit ChangeLog Configuration
Restored UserForm:Edit ChangeLog Policy
Restored UserForm:Edit ChangeLog
Including file 'sample/workflow.xml'.
Not saving object ProvisioningTask Create User: not found
Not saving object TaskDefinition Update User: not found
Not saving object TaskDefinition Delete User: not found
Not saving object TaskDefinition Disable User: not found
Not saving object TaskDefinition Enable User: not found
Not saving object TaskDefinition Rename User: not found
Not saving object TaskDefinition Change User Password: not found
Not saving object TaskDefinition Reset User Password: not found
Not saving object TaskDefinition Delete Resource Account: not found
Not saving object TaskDefinition Password Login: not found
Not saving object TaskDefinition Question Login: not found
Not saving object TaskDefinition Create Resource Object: not found
Not saving object TaskDefinition Update Resource Object: not found
Not saving object TaskDefinition Delete Resource Object: not found
Not saving object UserForm #ID#UserForm:ApprovalForm: not found
Including file 'sample/wfutil.xml'.
Added Configuration:Rename Task
Added Configuration:Parse Result
Added Configuration:Update View
Added Configuration:Update User View
Added Configuration:Set Password
Added Configuration:Update User Object
Added Configuration:Move User
Added Configuration:Sunrise Via Work Item
Added Configuration:Sunset
Added Configuration:Derive Date
Added Configuration:Data Transformation
Including file 'sample/wfapproval.xml'.
Preserving object EmailTemplate #ID#EmailTemplate:ProvisioningApproval
Preserving object EmailTemplate #ID#EmailTemplate:DeprovisioningApproval
Preserving object EmailTemplate #ID#EmailTemplate:ProvisioningNotification
Preserving object EmailTemplate #ID#EmailTemplate:DeprovisioningNotification
Preserving object EmailTemplate #ID#EmailTemplate:AccountUpdateNotification
Added Configuration:Approval
Restored UserForm:Approval Form
Added Rule:Approval Transaction Message
Added Rule:Approval Transaction Message Helper
Restored EmailTemplate:Account Creation Approval
Restored EmailTemplate:Account Deletion Approval
Restored EmailTemplate:Account Creation Notification
Restored EmailTemplate:Account Deletion Notification
Restored EmailTemplate:Account Update Notification
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Added Configuration:Multi Approval
Added Configuration:Approval Evaluator
Added Configuration:Notify
Added Configuration:Approval Notification Evaluator
Added Configuration:Notification Evaluator
Added Configuration:Lighthouse Approvals
Added Configuration:Provisioning Notification
Restored UserForm:Sunrise Form
Restored TaskDefinition:Approver Report
Including file 'sample/wfprovisioning.xml'.
Added Configuration:Provision With Retries
Updated EmailTemplate:Retry Notification
Added Configuration:Provision
Added Configuration:DeProvision
Added Configuration:Bulk Provision
Including file 'sample/wfrecon.xml'.
Added TaskDefinition:Audit Native Change To Account Attributes
Added TaskDefinition:Notify Reconcile Start
Added TaskDefinition:Notify Reconcile Response
Added TaskDefinition:Notify Reconcile Finish
Including file 'sample/wfresource.xml'.
Added Configuration:Resource Object Retries
Restored Configuration:Resource Policy
Added TaskDefinition:Create Resource Object
Added TaskDefinition:Update Resource Object
Added TaskDefinition:Delete Resource Object
Added TaskDefinition:Create Resource Group
Added TaskDefinition:Update Resource Group
Added TaskDefinition:Delete Resource Group
Added TaskDefinition:Create Resource Organization
Added TaskDefinition:Update Resource Organization
Added TaskDefinition:Delete Resource Organization
Added TaskDefinition:Create Resource Organizational Unit
Added TaskDefinition:Update Resource Organizational Unit
Added TaskDefinition:Delete Resource Organizational Unit
Added TaskDefinition:Create Resource Person
Added TaskDefinition:Update Resource Person
Added TaskDefinition:Delete Resource Person
Added TaskDefinition:Create Resource User
Added TaskDefinition:Update Resource User
Added TaskDefinition:Delete Resource User
Added Rule:Unix Excluded Resource Accounts
Added Rule:Windows Excluded Resource Accounts
Added Rule:Microsoft SQL Server Excluded Resource Accounts
Added Rule:Sun Access Manager Excluded Resource Accounts
Including file 'sample/wfuser.xml'.
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Added ProvisioningTask:Create User
Added TaskDefinition:Update User
Added TaskDefinition:Delete User
Added TaskDefinition:Disable User
Added TaskDefinition:Move User
Added TaskDefinition:Enable User
Added TaskDefinition:Unlock User
Added TaskDefinition:Rename User
Added TaskDefinition:Change User Password
Added TaskDefinition:Reset User Password
Added TaskDefinition:Password Login
Added TaskDefinition:Question Login
Added TaskDefinition:Change Resource Account Password
Added TaskDefinition:Handle LDAP Modify DN
Added TaskDefinition:Complete Sunrise Account Deferred
Including file 'sample/wfsystem.xml'.
Added TaskDefinition:Manage Role
Added TaskDefinition:Manage Resource
Including file 'sample/taskconfig.xml'.
Preserving object UserForm #ID#TaskTemplate:CreateUser
Preserving object UserForm #ID#TaskTemplate:UpdateUser
Preserving object UserForm #ID#TaskTemplate:DeleteUser
Including file 'sample/forms/TemplateFormLibrary.xml'.
Restored UserForm:Template Form Library
Including file 'sample/forms/CreateUserTaskTemplateForm.xml'.
Restored UserForm:Create User Template Form
Restored TaskTemplate:Create User Template
Including file 'sample/forms/DeleteUserTaskTemplateForm.xml'.
Restored UserForm:Delete User Template Form
Restored TaskTemplate:Delete User Template
Including file 'sample/forms/UpdateUserTaskTemplateForm.xml'.
Added UserForm:Update User Template Form
Restored TaskTemplate:Update User Template
Including file 'sample/enduser.xml'.
Not saving object UserForm #ID#UserForm:EndUserMenu: not found
Not saving object UserForm #ID#UserForm:AnonymousUserMenu: not found
Not saving object UserForm #ID#UserForm:End User Form: not found
Not saving object UserForm #ID#UserForm:AnonymousUserLogin: not found
Not saving object UserForm #ID#UserForm:ChangePasswordForm: not found
Not saving object UserForm #ID#UserForm:ResetPasswordForm: not found
Not saving object UserForm #ID#UserForm:ExpiredLoginForm: not found
Not saving object UserForm #ID#UserForm:QuestionLoginForm: not found
Not saving object Configuration EndUserRuleLibrary: not found
Preserving object Configuration #ID#Configuration:EndUserResources
Preserving object Configuration #ID#Configuration:EndUserTasks
Including file 'sample/enduserlib.xml'.
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Added Configuration:EndUserRuleLibrary
Including file 'sample/rules/ResourceFormRules.xml'.
Added Configuration:ResourceFormRuleLibrary
Including file 'sample/rules/RegionalConstants.xml'.
Added Configuration:Regional Constants
Including file 'sample/forms/ADUserForm.xml'.
Added UserForm:AD User Form
Including file 'sample/forms/AIXUserForm.xml'.
Added UserForm:AIX User Form
Including file 'sample/forms/HP-UXUserForm.xml'.
Added UserForm:HP-UX User Form
Including file 'sample/forms/LDAPUserForm.xml'.
Added UserForm:LDAP User Form
Including file 'sample/forms/NDSUserForm.xml'.
Added UserForm:NDS User Form
Including file 'sample/forms/SolarisUserForm.xml'.
Added UserForm:Solaris User Form
Including file 'sample/forms/SUSELinuxUserForm.xml'.
Added UserForm:SUSE Linux User Form
Including file 'sample/forms/RedHatLinuxUserForm.xml'.
Added UserForm:Red Hat Linux User Form
Restored Configuration:End User Resources
Restored Configuration:End User Tasks
Restored Configuration:Anonymous User Tasks
Restored UserForm:End User Empty Form
Restored UserForm:End User Menu
Restored UserForm:Anonymous User Menu
Restored UserForm:Anonymous User Login
Restored UserForm:End User Form
Restored UserForm:Basic Change Password Form
Restored UserForm:Change Password Form
Restored UserForm:Expired Login Form
Restored UserForm:Login Change User Answers Form
Restored UserForm:Question Login Form
Restored UserForm:Change User Answers Form
Added UserForm:End User Field Library
Added UserForm:End User Dynamic Resource Forms
Added UserForm:Self Discovery
Added UserForm:End User Launch List
Added UserForm:End User Work Item List
Added UserForm:End User Other Work Item List
Added UserForm:End User Work Item Edit
Added UserForm:End User Work Item List Ext
Added UserForm:End User Work Item Confirmation Ext
Restored UserForm:End User Approvals List
Restored UserForm:End User Approvals Confirmation
```

EXAMPLE 1–2 Sample Output from lh Log Program *(Continued)*

```
Added UserForm:End User Task List
Added UserForm:End User Task Results
Restored UserForm:End User View WorkItem Delegations
Restored UserForm:End User Past WorkItem Delegations
Restored UserForm:End User Delegate WorkItems
Restored UserForm:End User Access Privileges
Restored UserForm:End User Dashboard
Restored UserForm:End User Navigation
Restored UserForm:End User Request Menu
Added Configuration:End User Update View
Added TaskDefinition:End User Update My Resources
Added TaskDefinition:End User Update My Roles
Added TaskDefinition:End User Update Resources
Added TaskDefinition:End User Update Roles
Restored UserForm:End User Anonymous Enrollment Completed Form
Restored UserForm:End User Anonymous Enrollment Validation Form
Restored UserForm:End User Anonymous Enrollment Completion Form
Restored UserForm:End User Anonymous Enrollment Form
Added TaskDefinition:End User Anonymous Enrollment
Restored EmailTemplate:End User Anonymous Enrollment Template
Including file 'sample/UserUIConfig.xml'.
Restored Configuration:UserUIConfig
Including file 'sample/reporttasks.xml'.
Added UserForm:Report Form Library
Added UserForm:Syslog Form Library
Restored TaskDefinition:AuditLog Report
Restored TaskDefinition:Historical User Changes Report
Restored TaskDefinition:Syslog Report
Restored TaskDefinition:Usage Report
Restored TaskDefinition:Role Report
Restored TaskDefinition:Admin Role Report
Restored TaskDefinition:User Report
Restored TaskDefinition:User Question Report
Restored TaskDefinition:Administrator Report
Restored TaskDefinition:Task Report
Restored TaskDefinition:LogTamperingReport
Restored TaskDefinition:Resource User Report
Restored TaskDefinition:Resource Group Report
Restored TaskDefinition:Default User Audit Report
Restored TaskDefinition:Account Index Summary
Restored TaskDefinition:Workflow Summary Report
Restored TaskDefinition:AuditLog Maintenance Task
Restored TaskDefinition:System Log Maintenance Task
Restored TaskDefinition:Resource Status Report
Including file 'sample/synchronization.xml'.
Added TaskDefinition:Resource Role Synchronizer
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored TaskDefinition:SourceAdapterTask
Restored TaskDefinition:IDM Synchronization
Restored TaskDefinition:IDMXUser Synchronization
Added Rule:supportedSyncObjectTypesForResource
Added Rule:getAvailableServerOptions
Restored UserForm:Synchronization Policy Edit
Including file 'sample/defaultreports.xml'.
Preserving object TaskTemplate #ID#TaskTemplate:PasswordChangeUsage
Preserving object TaskTemplate #ID#TaskTemplate:PasswordResetUsage
Preserving object TaskTemplate #ID#TaskTemplate:AccountsDeletedAudit
Preserving object TaskTemplate #ID#TaskTemplate:AccountsCreatedUsage
Preserving object TaskTemplate #ID#TaskTemplate:AccountsDeletedUsage
Preserving object TaskTemplate #ID#TaskTemplate:AllRoles
Preserving object TaskTemplate #ID#TaskTemplate:AllAdminRoles
Preserving object TaskTemplate #ID#TaskTemplate:AllAdministrators
Preserving object TaskTemplate #ID#TaskTemplate:AllUsers
Preserving object TaskTemplate #ID#TaskTemplate:WeeklyActivityAudit
Preserving object TaskTemplate #ID#TaskTemplate:DailyActivityAudit
Preserving object TaskTemplate #ID#TaskTemplate:ResPasswordResetAudit
Preserving object TaskTemplate #ID#TaskTemplate:ResAcctCreateAudit
Preserving object TaskTemplate #ID#TaskTemplate:ResPasswordChangeAudit
Preserving object TaskTemplate #ID#TaskTemplate:WeeklySystemMessages
Preserving object TaskTemplate #ID#TaskTemplate:MyDirectEmployeeSummary
Preserving object TaskTemplate #ID#TaskTemplate:MyDirectIndirectEmployeeSummary
Preserving object TaskTemplate #ID#TaskTemplate:MyDirectEmployeeDetail
Preserving object TaskTemplate #ID#TaskTemplate:MyDirectIndirectEmployeeDetail
Preserving object TaskTemplate #ID#TaskTemplate:HistoricalUserChangesTemplate
Restored TaskTemplate:Resource Accounts Deleted List
Restored TaskTemplate:All Administrators
Restored TaskTemplate:All Roles
Restored TaskTemplate:All Admin Roles
Restored TaskTemplate:All Users
Restored TaskTemplate:Today's Activity
Restored TaskTemplate:Resource Accounts Created List
Restored TaskTemplate:Resource Password Change List
Restored TaskTemplate:Resource Password Resets List
Restored TaskTemplate:Historical User Changes Report
Restored TaskTemplate:Weekly Activity
Restored TaskTemplate:Password Change Chart
Restored TaskTemplate:Password Reset Chart
Restored TaskTemplate:Created Resource Accounts Chart
Restored TaskTemplate:Deleted Resource Accounts Chart
Restored TaskTemplate:Recent System Messages
Restored TaskTemplate:My Direct and Indirect Employee Detail
Restored TaskTemplate:My Direct Employee Detail
Restored TaskTemplate:My Direct Employee Summary
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored TaskTemplate:My Direct and Indirect Employee Summary
Including file 'sample/tickerconfig.xml'.
Restored Configuration:TickerConfig
Including file 'sample/redirectFilterConfig.xml'.
Restored Configuration:Redirect Filter Configuration
Including file 'sample/resourceforms.xml'.
Including file 'sample/forms/AccessManagergroupcreate.xml'.
Added ResourceForm:Access Manager Create Group Form
Including file 'sample/forms/AccessManagergroupupdate.xml'.
Added ResourceForm:Access Manager Update Group Form
Including file 'sample/forms/ADgroupcreate.xml'.
Added ResourceForm:Windows Active Directory Create Group Form
Including file 'sample/forms/ADgroupupdate.xml'.
Added ResourceForm:Windows Active Directory Update Group Form
Including file 'sample/forms/ADorganizationalunitupdate.xml'.
Added ResourceForm:Windows Active Directory Update Organizational Unit Form
Including file 'sample/forms/ADorganizationalunitcreate.xml'.
Added ResourceForm:Windows Active Directory Create Organizational Unit Form
Including file 'sample/forms/ADcontainercreate.xml'.
Added ResourceForm:Windows Active Directory Create Container Form
Including file 'sample/forms/ADcontainerupdate.xml'.
Added ResourceForm:Windows Active Directory Update Container Form
Including file 'sample/forms/ADpersoncreate.xml'.
Added ResourceForm:Windows Active Directory Create User Form
Including file 'sample/forms/ADpersonupdate.xml'.
Added ResourceForm:Windows Active Directory Update User Form
Including file 'sample/forms/AIXgroupcreate.xml'.
Added ResourceForm:AIX Create Group Form
Including file 'sample/forms/AIXgroupupdate.xml'.
Added ResourceForm:AIX Update Group Form
Including file 'sample/forms/SP2groupcreate.xml'.
Added ResourceForm:SP2 Create Group Form
Including file 'sample/forms/SP2groupupdate.xml'.
Added ResourceForm:SP2 Update Group Form
Including file 'sample/forms/HP-UXgroupcreate.xml'.
Added ResourceForm:HP-UX Create Group Form
Including file 'sample/forms/HP-UXgroupupdate.xml'.
Added ResourceForm:HP-UX Update Group Form
Including file 'sample/forms/LDAPgroupcreate.xml'.
Added ResourceForm:LDAP Create Group Form
Including file 'sample/forms/LDAPgroupupdate.xml'.
Added ResourceForm:LDAP Update Group Form
Including file 'sample/forms/LDAPorganizationcreate.xml'.
Added ResourceForm:LDAP Create Organization Form
Including file 'sample/forms/LDAPorganizationupdate.xml'.
Added ResourceForm:LDAP Update Organization Form
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Including file 'sample/forms/LDAPorganizationalunitcreate.xml'.
Added ResourceForm:LDAP Create Organizational Unit Form
Including file 'sample/forms/LDAPorganizationalunitupdate.xml'.
Added ResourceForm:LDAP Update Organizational Unit Form
Including file 'sample/forms/LDAPpersoncreate.xml'.
Added ResourceForm:LDAP Create Person Form
Including file 'sample/forms/LDAPpersonupdate.xml'.
Added ResourceForm:LDAP Update Person Form
Including file 'sample/forms/LDAPPosixGroupCreate.xml'.
Added ResourceForm:LDAP Create Posix Group Form
Including file 'sample/forms/LDAPPosixGroupUpdate.xml'.
Added ResourceForm:LDAP Update Posix Group Form
Including file 'sample/forms/SAPPortalgroupcreate.xml'.
Added ResourceForm:SAP Enterprise Portal Create Group Form
Including file 'sample/forms/SAPPortalgroupupdate.xml'.
Added ResourceForm:SAP Enterprise Portal Update Group Form
Including file 'sample/forms/SAPPortalroleupdate.xml'.
Added ResourceForm:SAP Enterprise Portal Update Role Form
Including file 'sample/forms/SunAMCreateFilteredGroupForm.xml'.
Added ResourceForm:Sun Access Manager Create Filtered Group Form
Including file 'sample/forms/SunAMUpdateFilteredGroupForm.xml'.
Added ResourceForm:Sun Access Manager Update Filtered Group Form
Including file 'sample/forms/SunAMCreateDynamicGroupForm.xml'.
Added ResourceForm:Sun Access Manager Create Dynamic Subscription Group Form
Including file 'sample/forms/SunAMUpdateDynamicGroupForm.xml'.
Added ResourceForm:Sun ONE Identity Server Update Dynamic Subscription Group Form
Including file 'sample/forms/SunAMCreateStaticGroupForm.xml'.
Added ResourceForm:Sun Access Manager Create Static Subscription Group Form
Including file 'sample/forms/SunAMUpdateStaticGroupForm.xml'.
Added ResourceForm:Sun Access Manager Update Static Subscription Group Form
Including file 'sample/forms/SunAMCreateRoleForm.xml'.
Added ResourceForm:Sun Access Manager Create Role Form
Including file 'sample/forms/SunAMUpdateRoleForm.xml'.
Added ResourceForm:Sun Access Manager Update Role Form
Including file 'sample/forms/SunAMCreateOrganizationForm.xml'.
Added ResourceForm:Sun Access Manager Create Organization Form
Including file 'sample/forms/SunAMUpdateOrganizationForm.xml'.
Added ResourceForm:Sun Access Manager Update Organization Form
Including file 'sample/forms/SunAMRealmCreateGroupForm.xml'.
Added ResourceForm:Sun Access Manager Realm Create Group Form
Including file 'sample/forms/SunAMRealmUpdateGroupForm.xml'.
Added ResourceForm:Sun Access Manager Realm Update Group Form
Including file 'sample/forms/SunAMRealmCreateRoleForm.xml'.
Added ResourceForm:Sun Access Manager Realm Create Role Form
Including file 'sample/forms/SunAMRealmUpdateRoleForm.xml'.
Added ResourceForm:Sun Access Manager Realm Update Role Form
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Including file 'sample/forms/SunAMRealmCreateFilteredRoleForm.xml'.
Added ResourceForm:Sun Access Manager Realm Create Filtered Role Form
Including file 'sample/forms/SunAMRealmUpdateFilteredRoleForm.xml'.
Added ResourceForm:Sun Access Manager Realm Update Filtered Role Form
Including file 'sample/forms/NDSgroupcreate.xml'.
Added ResourceForm:Netware NDS Create Group Form
Including file 'sample/forms/NDSgroupupdate.xml'.
Added ResourceForm:NetWare NDS Update Group Form
Including file 'sample/forms/NDSorganizationcreate.xml'.
Added ResourceForm:NetWare NDS Create Organization Form
Including file 'sample/forms/NDSorganizationupdate.xml'.
Added ResourceForm:NetWare NDS Update Organization Form
Including file 'sample/forms/NDSorganizationalunitcreate.xml'.
Added ResourceForm:NetWare NDS Create Organizational Unit Form
Including file 'sample/forms/NDSorganizationalunitupdate.xml'.
Added ResourceForm:NetWare NDS Update Organizational Unit Form
Including file 'sample/forms/NISgroupcreate.xml'.
Added ResourceForm:NIS Create Group Form
Including file 'sample/forms/NISgroupupdate.xml'.
Added ResourceForm:NIS Update Group Form
Including file 'sample/forms/NTgroupcreate.xml'.
Added ResourceForm:Windows NT Create Group Form
Including file 'sample/forms/NTgroupupdate.xml'.
Added ResourceForm:Windows NT Update Group Form
Including file 'sample/forms/RedHatLinuxgroupcreate.xml'.
Added ResourceForm:Red Hat Linux Create Group Form
Including file 'sample/forms/RedHatLinuxgroupupdate.xml'.
Added ResourceForm:Red Hat Linux Update Group Form
Including file 'sample/forms/Siebelpositioncreate.xml'.
Added ResourceForm:Siebel Create Position Form
Including file 'sample/forms/Siebelpositionupdate.xml'.
Added ResourceForm:Siebel Update Position Form
Including file 'sample/forms/SiteMinderLDAPgroupcreate.xml'.
Added ResourceForm:SiteMinderLDAP Create Group Form
Including file 'sample/forms/SiteMinderLDAPgroupupdate.xml'.
Added ResourceForm:SiteMinderLDAP Update Group Form
Including file 'sample/forms/SiteMinderLDAPorganizationcreate.xml'.
Added ResourceForm:SiteMinderLDAP Create Organization Form
Including file 'sample/forms/SiteMinderLDAPorganizationupdate.xml'.
Added ResourceForm:SiteMinderLDAP Update Organization Form
Including file 'sample/forms/SiteMinderLDAPorganizationalunitcreate.xml'.
Added ResourceForm:SiteMinderLDAP Create Organizational Unit Form
Including file 'sample/forms/SiteMinderLDAPorganizationalunitupdate.xml'.
Added ResourceForm:SiteMinderLDAP Update Organizational Unit Form
Including file 'sample/forms/Solarisgroupcreate.xml'.
Added ResourceForm:Solaris Create Group Form
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Including file 'sample/forms/Solarisgroupupdate.xml'.
Added ResourceForm:Solaris Update Group Form
Including file 'sample/forms/SUSELinuxgroupcreate.xml'.
Added ResourceForm:SUSE Linux Create Group Form
Including file 'sample/forms/SUSELinuxgroupupdate.xml'.
Added ResourceForm:SUSE Linux Update Group Form
Including file 'sample/forms/SunJSCSActiveSyncForm.xml'.
Added UserForm:Sun Java System Communications Services ActiveSync Form
Including file 'sample/forms/SunJSCSGroupCreate.xml'.
Added ResourceForm:Sun Java System Communications Services Create Group Form
Including file 'sample/forms/SunJSCSGroupUpdate.xml'.
Added ResourceForm:Sun Java System Communications Services Update Group Form
Including file 'sample/forms/SunJSCSOrganizationalUnitCreate.xml'.
Added ResourceForm:Sun Java System Communications Services Create
  Organizational Unit Form
Including file 'sample/forms/SunJSCSOrganizationalUnitUpdate.xml'.
Added ResourceForm:Sun Java System Communications Services Update
  Organizational Unit Form
Including file 'sample/forms/SunJSCSOrganizationCreate.xml'.
Added ResourceForm:Sun Java System Communications Services Create Organization Form
Including file 'sample/forms/SunJSCSOrganizationUpdate.xml'.
Updated ResourceForm:Sun Java System Communications Services Update Group Form
Including file 'sample/resourceAccountChangePassword.xml'.
Added ResourceForm:Change Resource Account Password Form
Including file 'sample/resourceAccountResetPassword.xml'.
Added ResourceForm:Reset Resource Account Password Form
Including file 'sample/resourcePolicyModify.xml'.
Added ResourceForm:Edit Resource Policy Form
Including file 'sample/resourceGroupDelete.xml'.
Added UserForm:Delete Group Form
Including file 'sample/resourceObjectFind.xml'.
Added ResourceForm:Find Resource Object Form
Including file 'sample/resourceObjectFindResults.xml'.
Added ResourceForm:Find Resource Object Results Form
Including file 'sample/resourceObjectRename.xml'.
Added ResourceForm:Rename Resource Object Form
Including file 'sample/resourceWizardForms.xml'.
Restored UserForm:Resource Wizard Library
Restored UserForm:Resource Wizard
Restored UserForm:Resource Rename Form
Restored UserForm:Database Resource Wizard Library
Restored UserForm:Resource Wizard Database Table
Restored UserForm:Resource Wizard ScriptedJDBC
Restored UserForm:Resource Wizard Microsoft Identity Integration Server
Restored UserForm:Resource Wizard PeopleSoft Component Interface
Restored UserForm:Resource Wizard JMS Listener
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored UserForm:Resource Wizard Sun Access Manager
Restored UserForm:Resource Wizard FlatFileActiveSync
Including file 'sample/resourceActiveSyncWizardLibrary.xml'.
Restored UserForm:Resource Active Sync Wizard Library
Including file 'sample/resourceActiveSyncWizardForms.xml'.
Including file 'sample/resourceActiveSyncWizardLibrary.xml'.
Updated UserForm:Resource Active Sync Wizard Library
Including file 'sample/conditionDialog.xml'.
Restored UserForm:Condition Dialog
Restored UserForm:Resource Active Sync Wizard
Including file 'sample/resourceList.xml'.
Restored UserForm:Resource List Library
Restored UserForm:Resource List Form
Including file 'sample/resourceTableTasks.xml'.
Added UserForm:Resource Create Form
Added UserForm:Resource Delete Form
Added UserForm:Default Resource Rename Form
Added UserForm:Resource Saveas Form
Added UserForm:Resource Object Create Form
Added UserForm:Resource Object Rename Form
Added UserForm:Resource Object Saveas Form
Added UserForm:Resource Object Delete Form
Including file 'sample/raforms.xml'.
Restored RiskReportTask:Windows NT Risk Analysis
Restored RiskReportTask:Windows Active Directory Risk Analysis
Restored RiskReportTask:NetWare NDS Risk Analysis
Restored RiskReportTask:AIX Risk Analysis
Restored RiskReportTask:Solaris Risk Analysis
Restored RiskReportTask:HPUX Risk Analysis
Restored RiskReportTask:Red Hat Linux Risk Analysis
Added RiskReportTask:Windows Active Directory Inactive Account Scan
Added RiskReportTask:Windows NT Inactive Account Scan
Added RiskReportTask:OS/400 Inactive Account Scan
Added RiskReportTask:ACF2 Inactive Account Scan
Including file 'sample/reconconfig.xml'.
Restored TaskDefinition:Reconcile
Restored TaskDefinition:Reconcile Requestor
Preserving object EmailTemplate #ID#EmailTemplate:ReconcileResourceEvent
Preserving object EmailTemplate #ID#EmailTemplate:ReconcileAccountEvent
Preserving object EmailTemplate #ID#EmailTemplate:ReconcileSummary
Restored EmailTemplate:Reconcile Resource Event
Restored EmailTemplate:Reconcile Account Event
Restored EmailTemplate:Reconcile Summary
Added Configuration:ReconConfigProxy
Including file 'sample/reconRules.xml'.
Added Rule:USER_NAME_MATCHES_ACCOUNT_ID
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Added Rule:USER_OWNS_MATCHING_ACCOUNT_ID
Added Rule:USER_EMAIL_MATCHES_ACCOUNT_EMAIL_CORR
Added Rule:USER_EMAIL_MATCHES_ACCOUNT_EMAIL_CONF
Added Rule:USER_FIRST_AND_LAST_NAMES_MATCH_ACCOUNT
Added Rule:SCHEDULING_RULE_ACCEPT_ALL_DATES
Including file 'sample/components.xml'.
Added Configuration:Component Properties
Including file 'sample/messages.xml'.
Added Configuration:defaultTestCustomCatalog
Including file 'sample/bulkoptask.xml'.
Preserving object TaskDefinition #ID#TaskDefinition:BulkOpTask
Restored TaskDefinition:Bulk Actions Task
Including file 'sample/dictionaryconfig.xml'.
Added Rule:TestDictionary
Added Rule:InsertDictionaryWord
Added Rule:CheckDictionaryWord
Preserving object Configuration #ID#Configuration:DictionaryConfig
Restored Configuration:Dictionary Configuration
Restored TaskDefinition:DictionaryLoader
Restored Configuration:DictionaryConfigForm
Including file 'sample/attrparse.xml'.
Added AttrParse:Default ACF2 AttrParse
Added AttrParse:Default VMS AttrParse
Added AttrParse:Default TopSecret ListUser CICS AttrParse
Added AttrParse:Default TopSecret ListUser TSO AttrParse
Added AttrParse:Default TopSecret TSO Segment AttrParse
Added AttrParse:Default TopSecret OMVS Segment AttrParse
Added AttrParse:Default TopSecret CICS Segment AttrParse
Added AttrParse:Default TopSecret ListAllObjects AttrParse
Added AttrParse:Default RACF ListUser AttrParse
Added AttrParse:Default LDAP RACF ListUser AttrParse
Added AttrParse:Default RACF TSO Segment AttrParse
Added AttrParse:Default RACF OMVS Segment AttrParse
Added AttrParse:Default RACF CICS Segment AttrParse
Added AttrParse:Default RACF NETVIEW AttrParse
Added AttrParse:Default Natural ListUser AttrParse
Added AttrParse:Default Natural ListUser Groups AttrParse
Added AttrParse:Default Natural ListAllObjects AttrParse
Including file 'sample/serverForms.xml'.
Not saving object Configuration #ID#Form:EmailTemplateSettingsForm: not found
Not saving object Configuration #ID#Form:ReconcilerSettingsForm: not found
Not saving object Configuration #ID#Form:SchedulerSettingsForm: not found
Not saving object Configuration #ID#Form:ServerSettingsForm: not found
Not saving object Configuration #ID#Form:JMXSettingsForm: not found
Restored Configuration:Reconciler Settings Form
Restored Configuration:Scheduler Settings Form
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored Configuration:Email Template Settings Form
Restored Configuration:JMX Settings Form
Restored Configuration:Server Settings Form
Including file 'sample/scripttasks.xml'.
Added TaskDefinition:Script Interpreter
Including file 'sample/workItemTypes.xml'.
Restored Configuration:WorkItemTypes
Including file 'sample/rules/AlphaNumeric.xml'.
Added Configuration:Alpha Numeric Rules
Including file 'sample/rules/DateLibrary.xml'.
Added Configuration:Date Library
Including file 'sample/rules/NamingRules.xml'.
Added Rule:Fullname - Last comma First
Added Rule:Fullname - First space Last
Added Rule:Fullname - First space MI space Last
Added Rule:AccountName - First dot Last
Added Rule:AccountName - First initial Last
Added Rule:AccountName - First underscore Last
Added Rule:Email
Including file 'sample/rules/RegionalConstants.xml'.
Updated Configuration:Regional Constants
Including file 'sample/rules/LoginCorrelationRules.xml'.
Added Rule:Correlate via X509 Certificate SubjectDN
Added Rule:Correlate via LDAP Uid
Including file 'sample/rules/NewUserNameRules.xml'.
Added Rule:Use SubjectDN Common Name
Including file 'sample/rules/ActiveSyncRules.xml'.
Added Rule:ActiveSync has isDeleted set
Including file 'sample/PeopleSoftComponentInterfaces.xml'.
Restored Configuration:PeopleSoft Component Interfaces
Including file 'sample/wfpwsync.xml'.
Preserving object EmailTemplate #ID#EmailTemplate:PasswordSyncNotice
Preserving object EmailTemplate #ID#EmailTemplate:PasswordSyncFailureNotice
Added TaskDefinition:Synchronize User Password
Restored EmailTemplate:PasswordSyncFailureNotification
Restored EmailTemplate:PasswordSyncNotification
Including file 'sample/adsyncfailover.xml'.
Added TaskDefinition:Active Directory Synchronization Recovery Collector
Added TaskDefinition:Active Directory Synchronization Failover
Including file 'sample/reportConfig.xml'.
Restored Configuration:Reports Configuration
Restored Configuration:Tracked Events Configuration
Added UserForm:Reports Config Library
Added UserForm:Tracked Events Config Library
Restored UserForm:Reports Configuration Form
Including file 'sample/auditor.xml'.
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Including file 'sample/auditorrules.xml'.
Added Rule:Review Everyone
Added Rule:Review Changed Users
Added Rule:Reject Changed Users
Added Rule:Default Remediator
Added Rule:Default Attestor
Added Rule:Default Escalation Attestor
Added Rule:All Non-Administrators
Added Rule:All Administrators
Added Rule:Users Without a Manager
Added Rule:Compare Accounts to Roles
Added Rule:Compare Roles to actual Resource values
Added AuditPolicy:IdM Role Comparison
Added AuditPolicy:IdM Account Accumulation
Including file 'sample/auditorforms.xml'.
Preserving object Rule #ID#Rule:ViolationPriority
Preserving object Rule #ID#Rule:ViolationSeverity
Updated UserForm:AuditPolicyLibrary
Updated UserForm:AuditorFormLibrary
Updated UserForm:Audit Policy List
Updated UserForm:Audit Policy Delete Confirmation Form
Updated UserForm:Audit Policy Form
Updated UserForm:Update Audit Policy Form
Updated UserForm:Remediation Library
Updated UserForm:Bulk Remediation
Updated UserForm:Sign Bulk Remediation
Ignoring changes to preserved object 'Rule:ViolationPriority'
Ignoring changes to preserved object 'Rule:ViolationSeverity'
Updated UserForm:Remediation List
Updated Configuration:AuditorOrgForm
Updated UserForm:Violation Detail Form
Updated UserForm:Compliance Violation Summary Form
Updated UserForm:Conflict Violation Details Form
Updated UserForm:Auditor Tab
Updated UserForm:Remediation Form
Including file 'sample/accessreviewforms.xml'.
Restored UserForm:Access Review Library
Restored UserForm:Bulk Attestation
Restored UserForm:Sign Bulk Attestation
Restored UserForm:Access Approval List
Restored UserForm:Access Review Dashboard
Restored UserForm:Access Review Delete Confirmation Form
Restored UserForm:Access Review Abort Confirmation Form
Restored UserForm:Attestation Form
Restored UserForm:Access Review Summary
Restored UserForm:Access Review Detail
```

EXAMPLE 1–2 Sample Output from lh Log Program *(Continued)*

```
Restored UserForm:Access Scan Form
Added UserForm:Access Scan List
Added UserForm:Access Scan Delete Confirmation Form
Restored UserForm:UserEntitlementForm
Restored UserForm:UserEntitlement Summary Form
Restored UserForm:Access Review Remediation Form
Restored UserForm:Access Scan Rename Form
Including file 'sample/auditortasks.xml'.
Restored TaskDefinition:Detailed User Report
Restored TaskDefinition:Audited Attribute Report
Restored TaskDefinition:Violation Summary Report
Restored TaskDefinition:Audit Policy Summary Report
Restored TaskDefinition:Audit Policy Scan
Restored TaskDefinition:Audit Policy Rescan
Added TaskDefinition:Applet Audit Policy Scan
Added UserForm:Policy Violation Report Library
Restored TaskDefinition:Organization Violation History
Restored TaskDefinition:Resource Violation History
Restored TaskDefinition:AuditPolicy Violation History
Restored TaskDefinition:Default Compliance Audit Report
Restored TaskDefinition:Audit Policy System Scan
Restored TaskDefinition:Separation of Duties
Restored TaskDefinition:Access Scan
Restored TaskDefinition:Access Review Rescan
Restored TaskDefinition:Access Review Detail Report
Restored TaskDefinition:Access Scan User Scope Report
Restored TaskDefinition:Access Review Coverage Report
Restored TaskDefinition:Access Review Summary Report
Restored TaskDefinition:Abort Access Review
Restored TaskDefinition:Recover Access Review
Restored TaskDefinition:Delete Access Review
Including file 'sample/auditorwfs.xml'.
Preserving object EmailTemplate #ID#EmailTemplate:PolicyViolationNotice
Preserving object EmailTemplate #ID#EmailTemplate:AttestationNotice
Preserving object EmailTemplate #ID#EmailTemplate:BulkAttestationNotice
Preserving object EmailTemplate #ID#EmailTemplate:AccessScanBeginNotice
Preserving object EmailTemplate #ID#EmailTemplate:AccessScanEndNotice
Added Configuration:Remediation
Added Configuration:Access Review Remediation
Added Configuration:Attestation
Added Rule:Remediation Transaction Message
Added Rule:Remediation Transaction Message Helper
Added Rule:Attestation Transaction Message
Added Rule:Attestation Transaction Message Helper
Added Rule:Attestation Remediation Transaction Message
Added Rule:Attestation Remediation Transaction Message Helper
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored EmailTemplate:Policy Violation Notice
Added Configuration:Multi Remediation
Added TaskDefinition:Standard Remediation
Restored TaskDefinition:Remediation Report
Added Configuration:Update Compliance Violation
Restored EmailTemplate:Attestation Notice
Restored EmailTemplate:Access Review Remediation Notice
Restored EmailTemplate:Bulk Attestation Notice
Restored EmailTemplate:Access Scan Begin Notice
Restored EmailTemplate:Access Scan End Notice
Added TaskDefinition:Standard Attestation
Added TaskDefinition:Access Review
Added Configuration:Launch Access Scan
Added Configuration:Launch Entitlement Rescan
Added Configuration:Launch Violation Rescan
Added TaskDefinition:ScanNotification
Including file 'sample/auditorDefaultReports.xml'.
Preserving object TaskTemplate #ID#TaskTemplate:DefOrgViolationHistory
Preserving object TaskTemplate #ID#TaskTemplate:DefAuditPolicyViolationHistory
Preserving object TaskTemplate #ID#TaskTemplate:DefResourceViolationHistory
Preserving object TaskTemplate #ID#TaskTemplate:AllComplianceViolations
Preserving object TaskTemplate #ID##TaskTemplate:AllAuditPolicies
Preserving object TaskTemplate #ID#AllSeparationofDutiesViolations
Preserving object TaskTemplate #ID#AllAccessReviewSummary
Restored TaskTemplate:Default Organization Violation History
Restored TaskTemplate:Default AuditPolicy Violation History
Restored TaskTemplate:Default Resource Violation History
Restored TaskTemplate:All Compliance Violations
Restored TaskTemplate:All Audit Policies
Restored TaskTemplate:All Separation of Duties Violations
Restored TaskTemplate:All Access Review Summary
Including file 'sample/speInit.xml'.
Preserving object Configuration #ID#IDMXConfiguration
Preserving object Configuration #ID#IDMXTrackedEventsConfig
Preserving object Configuration #ID#IDMXTransactionManagerConfig
Preserving object Resource #ID#Resource:SPEEndUserDirectory
Preserving object Policy #ID#Policy:SPE
Restored Configuration:SPE
Restored UserForm:SPE Browse
Restored UserForm:SPE User Form
Restored UserForm:SPE Example End User Form
Restored Configuration:SPE SPML
Restored TaskDefinition:SPE Migration
Including file 'sample/speRules.xml'.
Restored Rule:SPE Example Is Account Locked Rule
Restored Rule:SPE Example Lock Account Rule
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Restored Rule:SPE Example Unlock Account Rule
Restored Rule:SPE Example Correlation Rule Returning Single Identity
Restored Rule:SPE Example Correlation Rule Returning List of Identities
Restored Rule:SPE Example Correlation Rule for LDAP Returning Option Map
Restored Rule:SPE Example Correlation Rule for Simulated Returning Option Map
Restored Rule:SPE Example Confirmation Rule Returning First Candidate
Restored Rule:SPE Example Confirmation Rule Rejecting All Candidates
Restored Rule:SPE Example Confirmation Rule Selecting Candidates Using AccountId
Including file 'sample/speConfigForm.xml'.
Restored UserForm:SPE Configuration
Including file 'sample/speSearchForms.xml'.
Restored UserForm:SPE Search Confirmation
Restored UserForm:SPE Bulk User Results
Restored UserForm:SPE Search
Restored UserForm:SPE Search Config
Restored UserForm:SPE User Delete
Including file 'sample/speTrackedEventConfig.xml'.
Restored Configuration:SPE Tracked Events Configuration
Including file 'sample/speTransactionManagerConfig.xml'.
Restored Configuration:SPE Transaction Manager Configuration
Including file 'sample/dashboardGraphConfig.xml'.
Restored UserForm:Dashboard Graph Configuration
Including file 'sample/dashboardConfig.xml'.
Restored UserForm:Dashboard Configuration
Including file 'sample/speDashboardExamples.xml'.
Added Configuration:Today's Completed Transactions by Type
Added Configuration:Recent Directory Search Duration
Added Configuration:Recent Per Server Total Locked Transactions
Added Configuration:Today's Concurrent Administrators (Sample Data)
Added Configuration:Today's Resource Operations (Sample Data)
Added Configuration:Recent Directory Searches
Added Configuration:Recent Transaction Searches
Added Configuration:Today's Concurrent Users (Sample Data)
Added Configuration:Recent Resource Operation Failures (Sample Data)
Added Configuration:Recent Administrator Operations by Type
Added Configuration:Recent Resource Operations (Sample Data)
Added Configuration:Recent Per Server Pending Retry Transactions
Added Configuration:Today's Active Sync Poll Durations
Added Configuration:Recent Concurrent Users (Sample Data)
Added Configuration:Recent Per Server Runnable Transactions
Added Configuration:Today's Resource Operations by Resource
Added Configuration:Recent Max Memory Usage by Server
Added Configuration:Today's Active Sync Errors by Resource and Type
Added Configuration:Recent Transaction Search Duration
Added Configuration:Today's Transaction Failures by Type
Added Configuration:Monthly Self-Service Operations (Sample Data)
```

EXAMPLE 1–2 Sample Output from `lh` Log Program *(Continued)*

```
Added Configuration:Today's Active Sync Operations by Resource
Added Configuration:Recent Completed Transactions by Server
Added Configuration:Today's Resource Operations by Resource and Result
Added Configuration:Recent Provisioning Operation Duration (Sample Data)
Added Configuration:Monthly Resource Operations (Sample Data)
Added Configuration:Recent Thread Count by Server
Added Configuration:Today's In Process Transactions
Added Configuration:Today's Resource Operations by Type (Sample Data)
Added Configuration:Today's Registration Requests (Sample Data)
Added Configuration:Recent Concurrent Administrators (Sample Data)
Added Configuration:Today's Active Sync Activity
Added Configuration:Recent Administrator Activity
Added Configuration:Today's Provisioning Activity
Added Configuration:Recent Server Activity
Added Configuration:Resource Operations (Sample Data)
Added Configuration:Today's Activity (Sample Data)
Added Configuration:Recent Activity (Sample Data)
Including file 'sample/speAdminGroups.xml'.
Restored AdminGroup:Service Provider View User
Restored AdminGroup:Service Provider Update User
Restored AdminGroup:Service Provider Create User
Restored AdminGroup:Service Provider Delete User
Restored AdminGroup:Service Provider User Administrator
Restored AdminGroup:Service Provider Administrator
Restored AdminGroup:Service Provider Admin Role Administrator
Including file 'sample/speTransactionSearch.xml'.
Restored UserForm:SPE Transaction Search
Including file 'sample/speTransactionConfigForm.xml'.
Restored UserForm:SPE Transaction Configuration Form
Including file 'sample/speEndUserForms.xml'.
Restored Configuration:SPEUserPages
Restored UserForm:SPE End-User Login
Restored UserForm:SPE End-User Question Login Form
Restored UserForm:SPE End-User Forms Library
Restored UserForm:SPE End-User Forgot Username
Restored UserForm:SPE End-User Forgot Password
Restored UserForm:SPE End-User Change UserId
Restored UserForm:SPE End-User Change Notifications
Restored UserForm:SPE End-User Change Password
Restored UserForm:SPE End-User Reset Password
Restored UserForm:SPE End-User Change Challenge Answers
Restored UserForm:SPE Enrollment Main Form
Restored UserForm:SPE Enrollment Validation Form
Restored UserForm:SPE Enrollment Form
Restored Configuration:SPE End-User Pages Library
Restored Configuration:SPE Sample Users
```

EXAMPLE 1–2 Sample Output from lh Log Program *(Continued)*

```
Including file 'sample/speEmailTemplates.xml'.
Restored EmailTemplate:SPE End-User Username Recovery
Restored EmailTemplate:SPE End-User Profile Locked
Restored EmailTemplate:SPE End-User Reset Password
Restored EmailTemplate:SPE End-User Update Authentication Answers
Restored EmailTemplate:SPE End-User Change Notifications
Restored EmailTemplate:SPE End-User Change Notifications Old Address
Restored EmailTemplate:SPE End-User Change Password
Restored EmailTemplate:SPE End-User Change User Id
Restored EmailTemplate:SPE End-User Registration Template
Restored EmailTemplate:SPE Update Template
Restored EmailTemplate:SPE Cancellation Template
Including file 'sample/speEndUserResource.xml'.
Restored Resource:SPE End-User Directory
Including file 'sample/spePolicy.xml'.
Restored Policy:SPE Policy
Including file 'sample/speLoginConfig.xml'.
Restored LoginModGroup:Default SPE Id/Pwd Login Module Group
Restored LoginApp:SPE User Interface
Including file 'sample/speLinkingPolicyForm.xml'.
Restored UserForm:SPE Linking Policy Form

Successfully imported file '/opt/SUNWappserver91/domains/domain1/applications/
  j2ee-modules/idm/sample/init.xml'.

 * * * * * * *

Configure demo forms, tasks, and policies
Create demo users
Configure email preferences


Successfully configured Sun Java[tm] System Identity Manager. You can now start
your application server and login to Identity Manager.
More information:
Log File: /opt/SUNWappserver91/domains/domain1/applications/j2ee-modules/idm/
  patches/logs/SaveConfigurationLog5229log
```

◆ ◆ ◆ CHAPTER 2

Integrating CA SiteMinder

Computer Associates (CA) SiteMinder (formerly Netegrity SiteMinder) is an enterprise infrastructure product that enables centralized, secure Web access management. Its features include user authentication and single sign-on, policy-based authorization, and identity federation. One of the first single sign-on products to arrive on the market, legacy SiteMinder installations still exist to protect enterprise applications in many company networks.

This chapter describes options for integrating CA SiteMinder with Sun OpenSSO Enterprise. The chapter also provides instructions for using OpenSSO Enterprise to configure end-to-end CA SiteMinder single sign-on.

The following topics are included in this chapter:

About CA SiteMinder

CA SiteMinder consists of two core components that are used for access control and single sign-on:

- Policy Server
- Policy Agents

The SiteMinder Policy Server provides policy management, authentication, authorization and accounting. The Policy Server core engine was developed in C/C++ and the core components run like process daemons with predefined TCP/IP ports. The policy user interface is a Java

applet-based console. A supported web server configured with a SiteMinder NSAPI plug-in provides the front-end HTTP interface. The policy user interface enables the user to create policies, domains, and realms, as well as to configure authentication schemes. The policy user interface also provides centralized agent configuration. SiteMinder also has a local Java applet-based console utility for managing system configuration such as authentication and authorization settings, port numbers, and so forth.

The SiteMinder Web Agent acts as a filter for protecting enterprise applications. SiteMinder provides various policy agents to access Web applications and content according to predefined security policies:

- Web policy agents
- SAML affiliated policy agents
- Application server policy agents
- RADIUS database policy agents
- TransactionMinder XML policy agents
- Custom policy agents (Any policy agent that is written using the SiteMinder Agent API)

The SAML Affiliated agent is part of CA Federated Security Services. The Security Services provide single sign-on from a producer site, such as a portal, to a SAML consumer acting as an affiliate in a federated network. The communication between the SAML Affiliated policy agent and SiteMinder at the producer site results in the generation of a SAML Assertion. The TransactionMinder XML Agent is an XML-enabled version of the SiteMinder Web Agent that authenticates and authorizes web services-bound URLs.

Authentication and Authorization

SiteMinder supports several authentication schemes as part of its authentication framework. Authentication schemes provide a way to collect credentials and determine the identity of a user. SiteMinder Credential Collector is an application within the web policy agent that gathers specific information about a user's credentials, and then sends the information to the Policy Server. For form-based authentication, credentials are acquired by the Forms Credential Collector (FCC) process. The default extension for FCC files is `.fcc`. FCC process files are composed in a simple mark-up language that includes HTML and some custom notation. The following describes a simple authentication scheme flow using a form-based authentication scheme:

1. A user requests a resource that is protected by a policy agent and contained in a realm. The realm is protected by an HTML form-based authentication scheme.
2. SiteMinder contacts the Policy Server and determines that the user request must be redirected to the credential collector.
3. The policy agent redirects to the URL of the Credential Collector file.
4. The Credential Collector displays the form described in the `.fcc` file of the user's browser.

5. The user fills out the custom form and submits it. The Credential Collector processes the credentials by submitting the form to the Policy Server.
6. If the user is authenticated, Credential Collector creates a session cookie and sends it to the browser. The browser redirects the user to the resource that the user originally requested.
7. The web policy agent handles user authorization by using the user's session.

User Sessions

SiteMinder supports persistent and non-persistent sessions. The standard SiteMinder sessions are non-persistent and contain user session data but no other user-specific data. For example, a session does not contain attributes unless configured to do so. The SiteMinder user session is created by the SiteMinder server upon successful authentication.

The servers send the user session SiteMinder Web Agent to set in the browser. The policy agent is responsible for validating the cookie and enforcing session timeouts. The cookie named `SMSESSION` contains the following parameters by default:

- Device name or host name
- User's full DN
- User Name
- Session idle timeout
- Session maximum timeout
- Session creation time

Understanding the SiteMinder User Cases

This chapter describes three use cases, all built upon legacy SiteMinder deployments. In each use case, SiteMinder continues to provide authentication service for legacy applications even after OpenSSO Enterprise is installed to protect the same enterprise applications. SiteMinder and OpenSSO Enterprise typically co-exist in the following use cases:

- Simple Single Sign-On
- Federated Single Sign-On in a Service Provider Environment
- Federated Single Sign-On in an Identity Provider Environment

Single logout for any these of these use cases can be implemented in many ways. The logout for federation use cases must have a link in the partner portal for the following URL:

```
http:<sphost>:<spport>/opensso/saml2/jsp/spSingleLogoutInit.jsp?metaAlias=
<metaalias>&idpEntityID=<idp entityid>&RelayState=<integrated product logout url>
```

Single logout can also be achieved using Identity Provider-initiated single logout.

Simple Single Sign-On Use Case

In this use case, a SiteMinder instance is already deployed and configured to protect some of the enterprise applications in a company intranet. In the architecture figure below, the legacy application is contained in the Protected Resource . The company wants to continue leveraging SiteMinder for authentication purposes, while adding OpenSSO Enterprise to the environment to protect the same application. OpenSSO Enterprise is also used to protect all applications subsequently added to the enterprise.

An OpenSSO Enterprise policy agent protects the Protected Resource, while OpenSSO Enterprise itself is protected by a SiteMinder Web Agent. In this use case, an access request goes to OpenSSO Enterprise for policy evaluation or for single sign-on purposes. But the SiteMinder Web Agent, installed on the same container as OpenSSO Enterprise, redirects the user to the SiteMinder login page for authentication. The OpenSSO Enterprise custom authentication module validates the SiteMinder session depending upon whether or not the user has previously logged in to OpenSSO Enterprise. After successful login, the OpenSSO Enterprise custom authentication module uses the SiteMinder session to generate an OpenSSO Enterprise session. OpenSSO Enterprise then honors the user session obtained by the SiteMinder Policy Server.

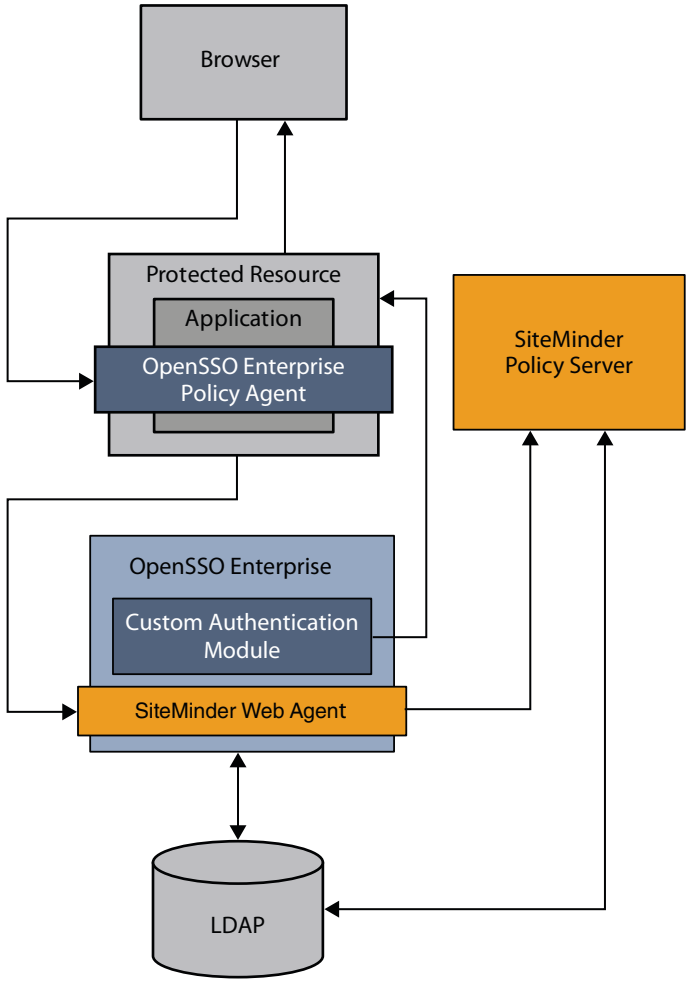


FIGURE 2–1 Single Sign-On Architecture

In this use case, both OpenSSO Enterprise server and SiteMinder policy server share the same user repository for user profile verification. OpenSSO Enterprise could also be configured to ignore the profile option if it relies on SiteMinder session for attributes.

The following figure illustrates the process flow for single sign-on using both SiteMinder and OpenSSO Enterprise.

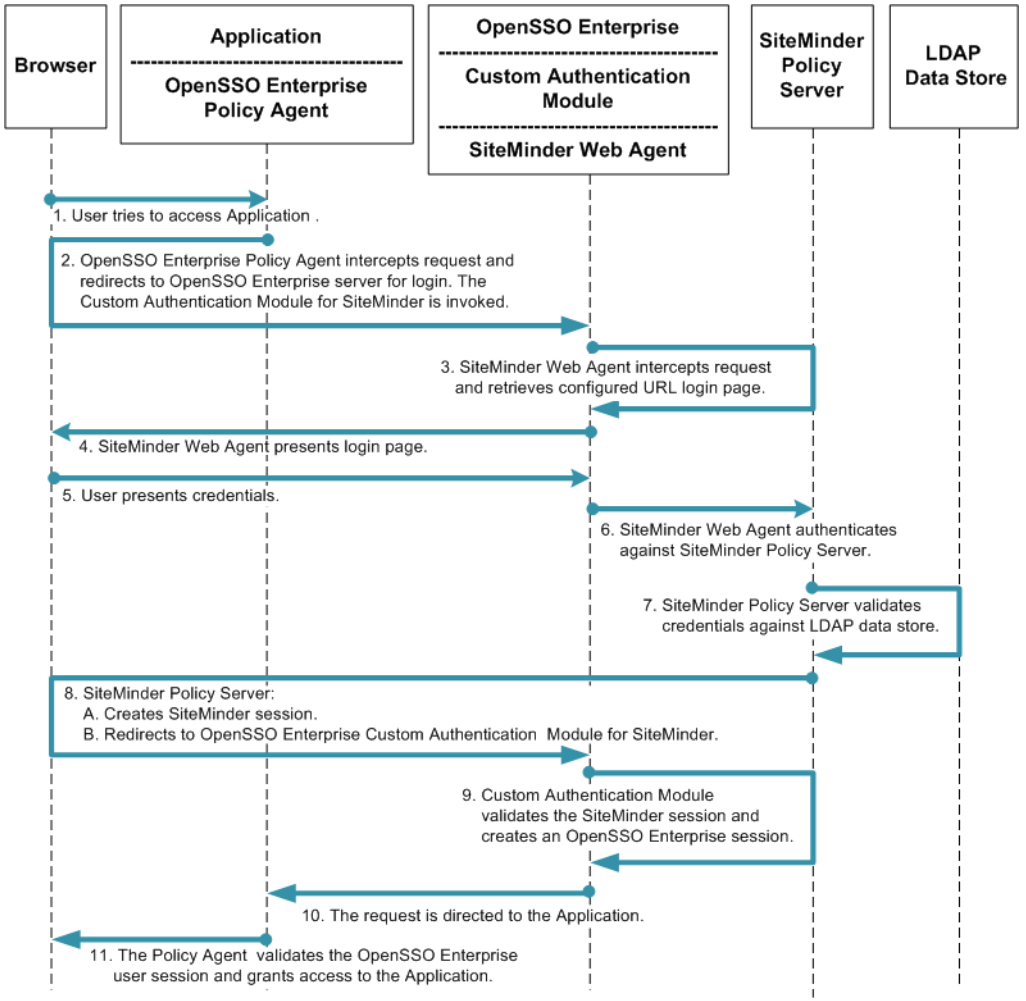


FIGURE 2–2 Single Sign-On Process Flow

Federated Single Sign-On Use Cases

The SAML, ID-FF, and WS-Federation protocols provide cross-domain single sign-on among multiple trusted business entities. These protocols are also used in Identity Federation. Identity Federation involves an Identity Provider, also known as an authentication provider, and a Service Provider where the user authentication session at the Identity provider is consumed. The following are common use cases in which SiteMinder is enabled for federation protocols:

- Enabling SiteMinder for federation protocols in a Service Provider environment
- Enabling SiteMinder for federation protocols in an Identity Provider environment

The deployment examples in this chapter are built upon simple single sign-on integration. You must set up single sign-on before enabling federation. For more information about setting up simple single sign-on, see the *Sun OpenSSO Enterprise Deployment Example: Single Sign-On*. After setting up simple single sign-on, you can enable SiteMinder for Federation in either the Identity Provider environment or in the Service Provider environment.

The federated single sign-on use cases are configured for transient federation. Transient federation assumes that the users exist only in the Identity Provider environment. The Service Provider honors user authentication at Identity Provider. The Service Provider then creates an anonymous session so that Service Provider applications, protected by single sign-on, can be accessed. During SAML interactions, user attribute information can be exchanged back to the Service Provider for authorization and other purposes.

Usually, bulk federation exists between Identity Provider and Service Provider. For more information about transient and bulk federation, see the OpenSSO Enterprise product documentation.

Federated Single Sign-On in an Identity Provider Environment

In this use case, the company uses SiteMinder in the Identity Provider environment to protect applications within the company intranet. As the company partners with external companies, the company deploys OpenSSO Enterprise in the Service Provider environment to leverage the SAMLv2 Federation protocols.

The following figure illustrates how SiteMinder can be enabled in an Identity Provider environment using OpenSSO Enterprise for federation protocols.

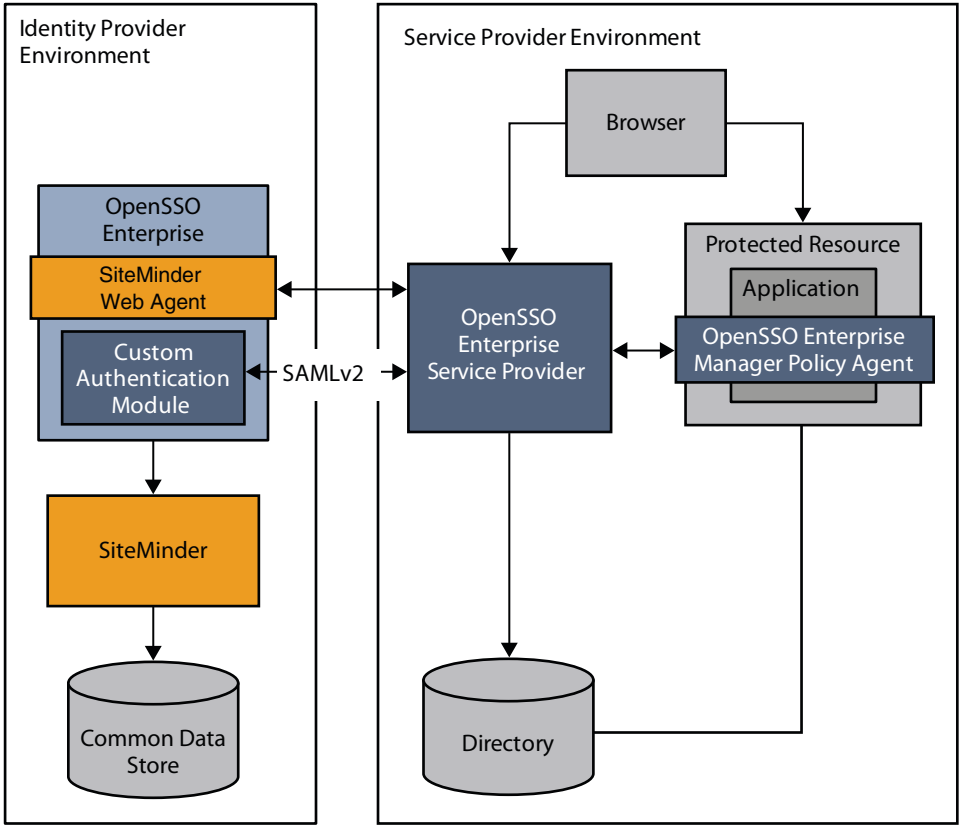


FIGURE 2–3 SiteMinder Federation in an Identity Provider Environment

In this deployment, OpenSSO Enterprise provides federated single sign-on among enterprise applications in partner environments, while SiteMinder continues to provide authentication. The following two figures illustrates a typical transaction flow.

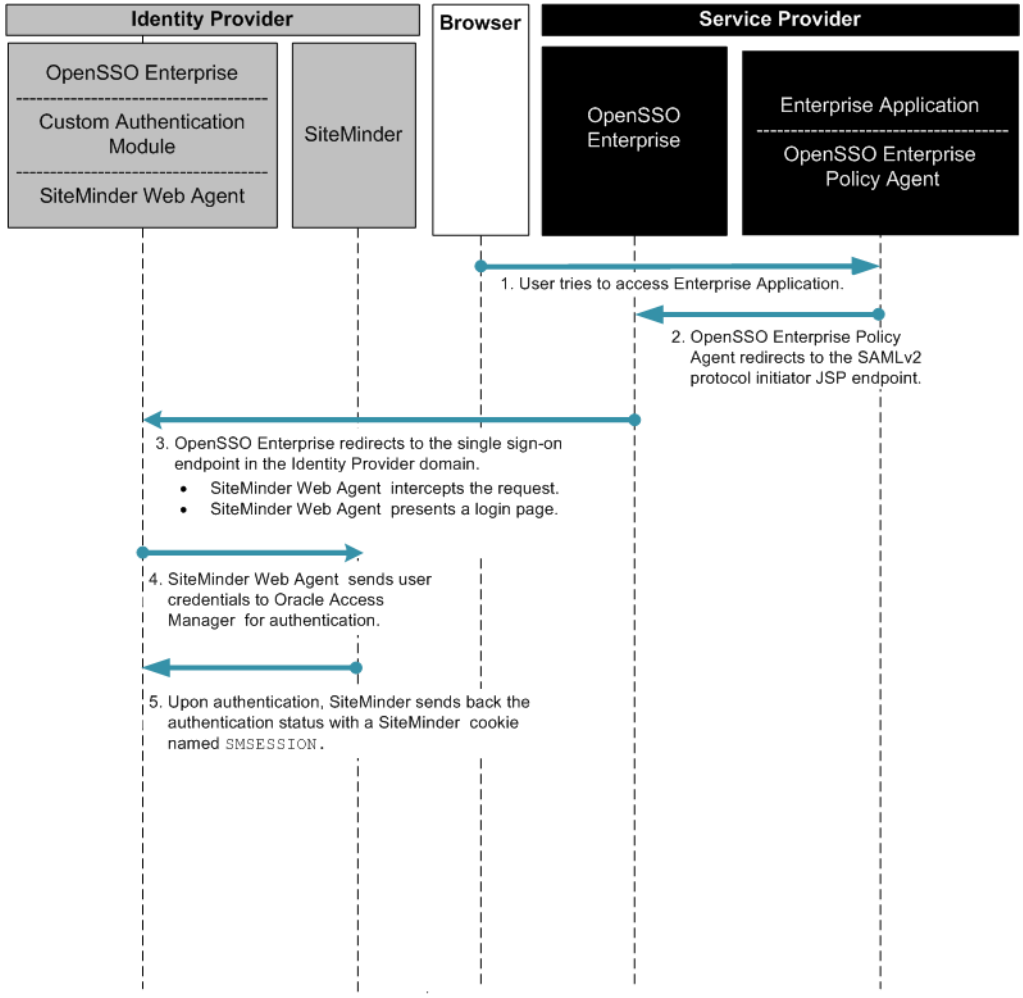


FIGURE 2–4 Process Flow for SiteMinder Federation in the Identity Provider Environment

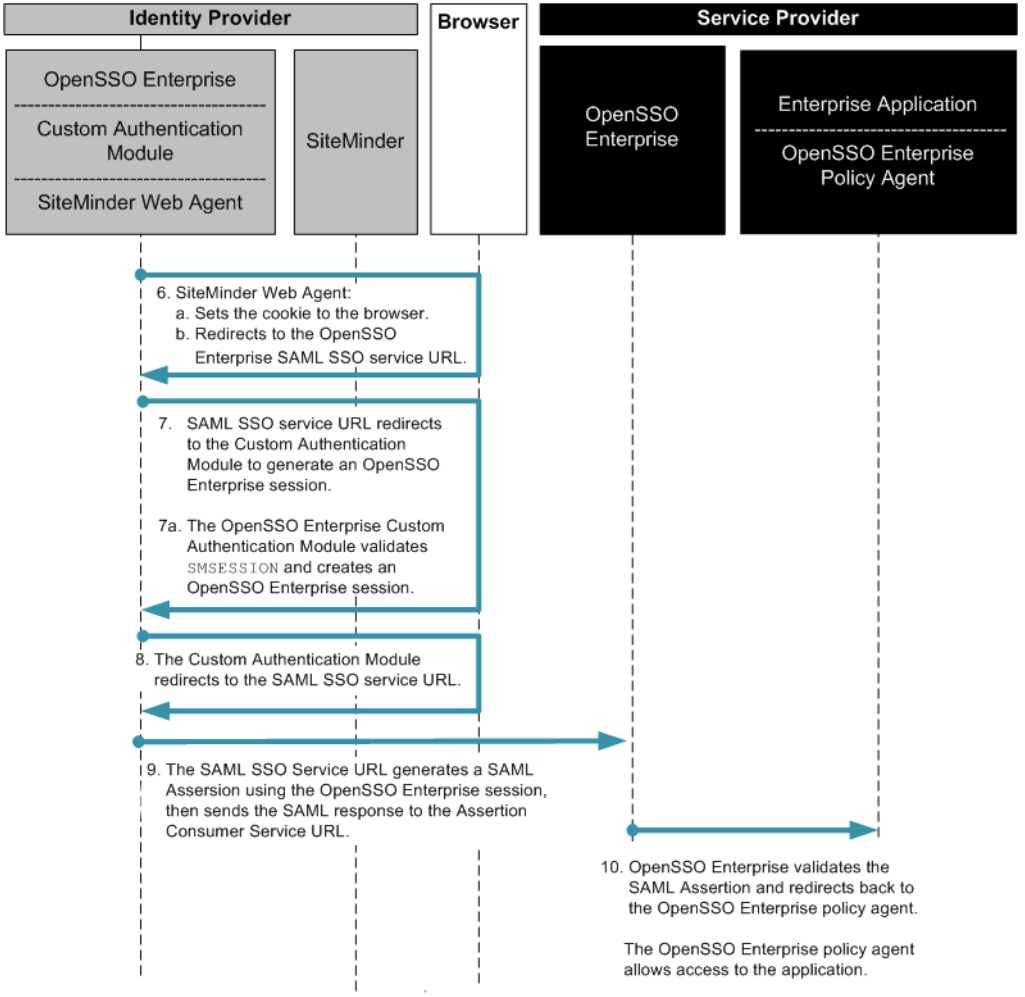


FIGURE 2–5 Process Flow for SiteMinder Federation in the Identity Provider Environment (continued)

Federated Single Sign-On Use Case in the Service Provider Environment

In this use case, the company uses SiteMinder in the Service Provider environment to protect legacy applications. OpenSSO Enterprise is installed to invoke Federation protocols. The OpenSSO Enterprise server includes a customized authentication module for handling SiteMinder sessions. A SiteMinder Web Agent is installed on the same OpenSSO Enterprise instance to protect OpenSSO Enterprise.

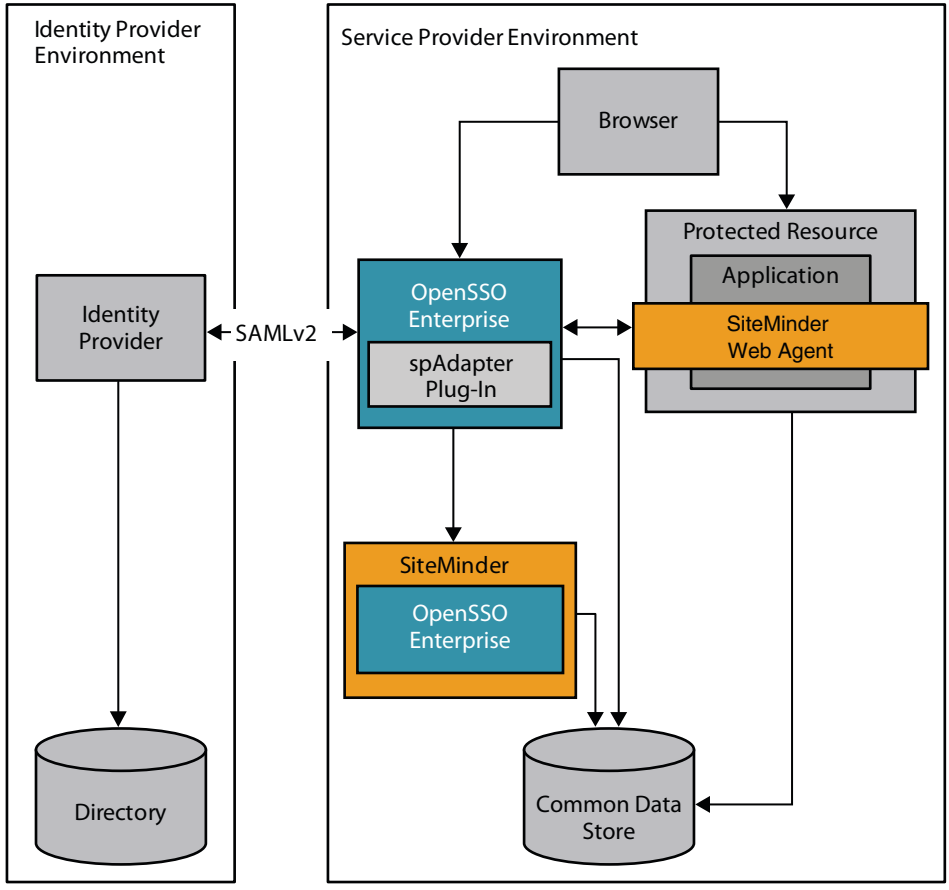


FIGURE 2–6 SiteMinder Federation in a Service Provider Environment

This use case includes two additional, lightweight components:

Custom Authentication Module (`spAdapter`)
: This is an OpenSSO Enterprise SAMLv2 plug-in that processes operations after federated single sign-on login is completed and before the target URL is displayed. After the OpenSSO Enterprise session is established, the `spAdapter` plug-in uses the OpenSSO Enterprise session to communicate with the SiteMinder Custom Authentication Scheme.

Custom Authentication Scheme
: This is a SiteMinder SAMLv2 plug-in. It uses the OpenSSO Enterprise configuration defined in the SAMLv2 metadata and the SAMLv2 session to generate a SiteMinder session.

When an access request comes from a partner application, the SiteMinder login page is displayed. If the user has already been authenticated, the OpenSSO Enterprise custom authentication module creates a session for the user. The custom authentication module

consumes the SiteMinder session, and then generates a SAML assertion. The following two figures illustrate the steps in the single sign-on flow:

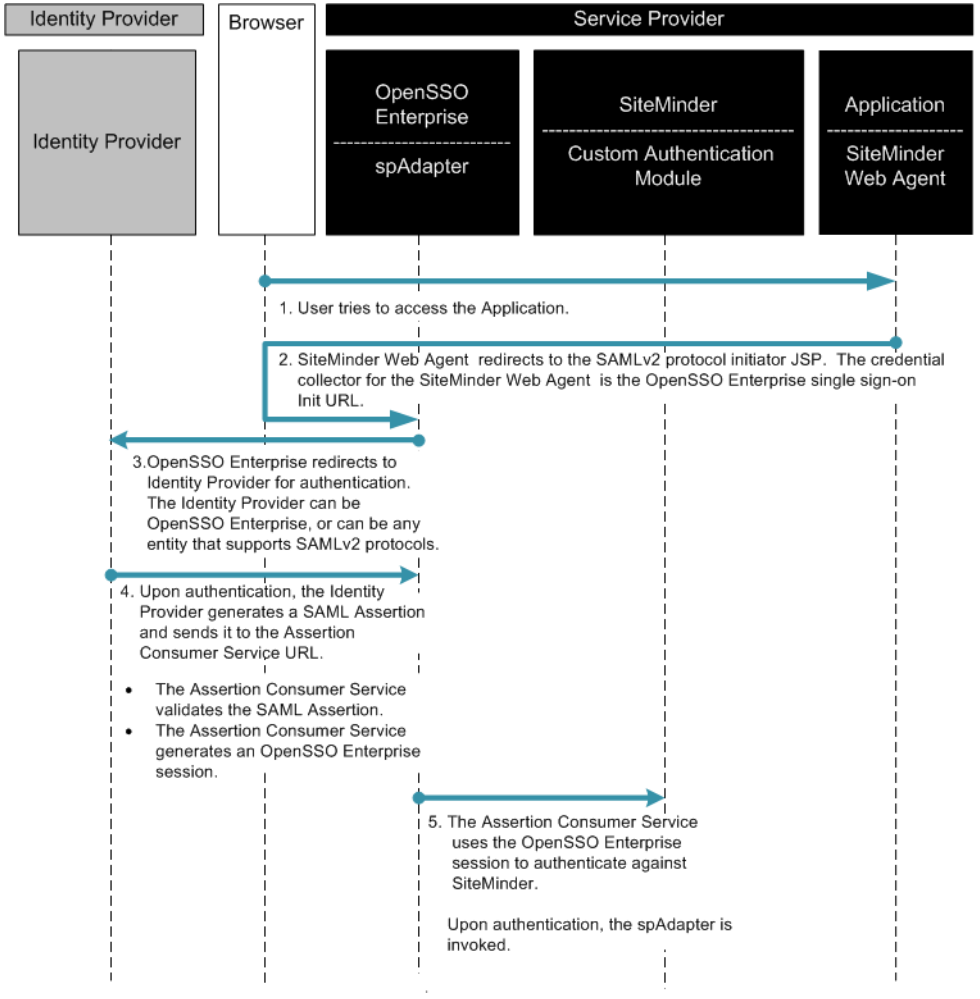


FIGURE 2–7 Process Flow for SiteMinder Federation in the Service Provider Environment

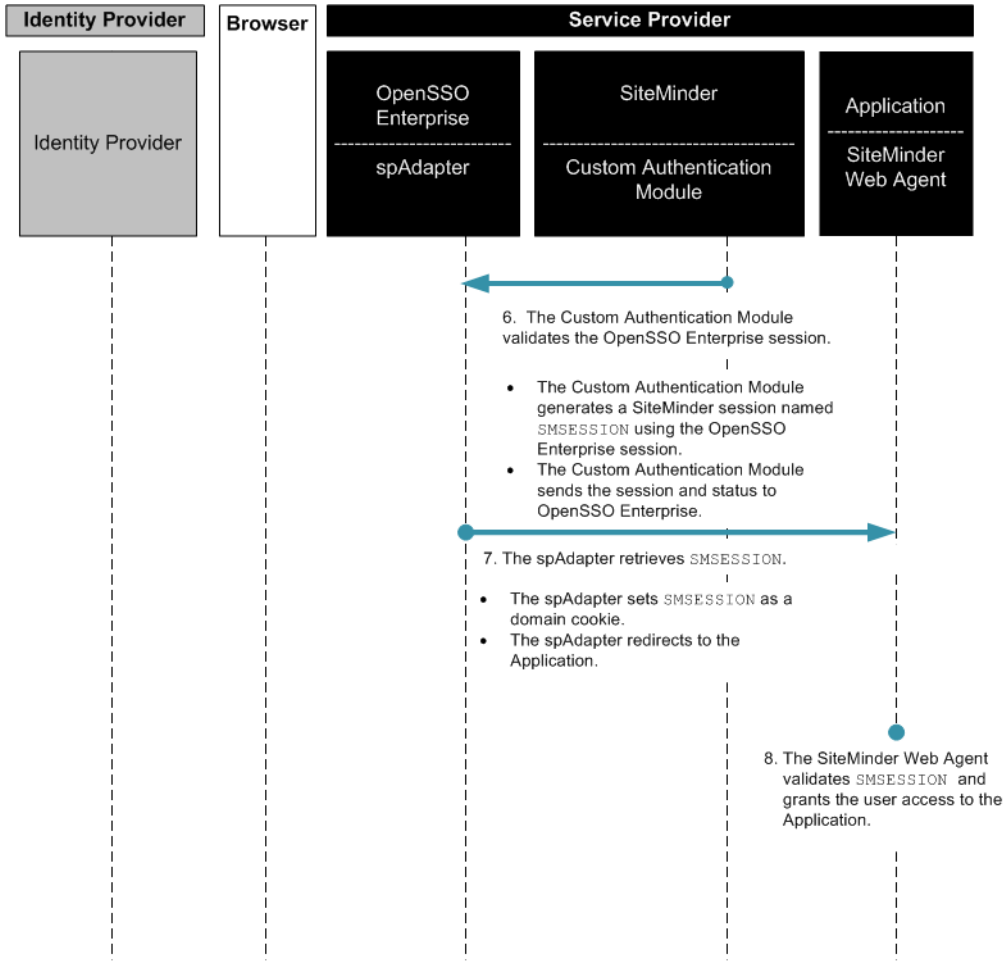


FIGURE 2–8 Process Flow for SiteMinder Federation in the Service Provider Environment (continued)

Installing SiteMinder

The use cases in this chapter describe Sun OpenSSO Enterprise 8.0 integrated with CA SiteMinder Server 6.0 Service Provider 5. Both products are installed on the Solaris operating system. Sun Web Server 6.1 SP5 is installed to serve the SiteMinder HTTP administrative interface. Sun Directory Server 6.3 is installed for its user data store and its configuration data store. Use these general instructions to install SiteMinder in any of the use cases discussed later in the chapter.

1. Install CA SiteMinder Access Manager.

You must have a licensed copy of CA SiteMinder to access its product documentation. See the product web page at `http://www.ca.com/us/products/product.aspx?id=5262`.

2. Install Sun Web Server 6.1 SP5.

   See the product documentation at *Sun Java System Web Server 6.1 2005Q4 SP5 Release Notes*.

3. Install Directory Server 6.3.

   See the product documentation at *Sun Java System Directory Server Enterprise Edition 6.3 Release Notes*.

Configuring SiteMinder After Installation

Use these general instructions after installing SiteMinder in any of the use cases discussed later in the chapter. To configure SiteMinder, follow these steps:

1. Log inn to SiteMinder.
2. Create a Sample User.
3. Create a SiteMinder Web Agent Configuration.
4. Create and Configure the User Directory.
5. Create and Configure a Form-Based Authentication Scheme.
6. Create a Policy.

▼ To Log In to SiteMinder

1 Go to the following URL:

`http://`*hostname*`:`*portnumber*`/SiteMinder`

2 On the SiteMinder Policy Server administration console, click "Administer Policy Server."

3 In the Policy Server login page, log in using the following credentials:

User Name: SiteMinder

Password: password

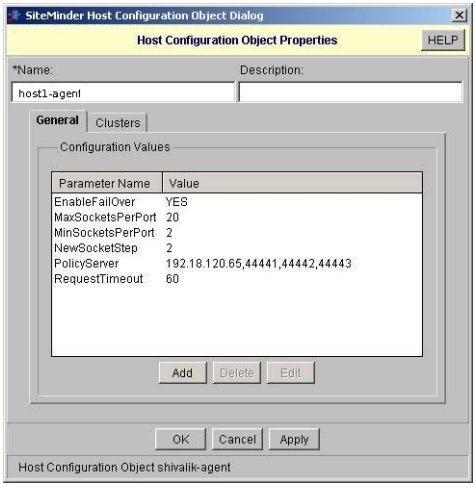
Creating a Sample User

Create a sample user in the SiteMinder Directory Server. In this use case, the new user is named `test`. You can base the name of this user on the attributes you use with SiteMinder. By default, Directory Server uses the `uid` naming attribute for the user.

▼ To Create a SiteMinder Web Agent Configuration

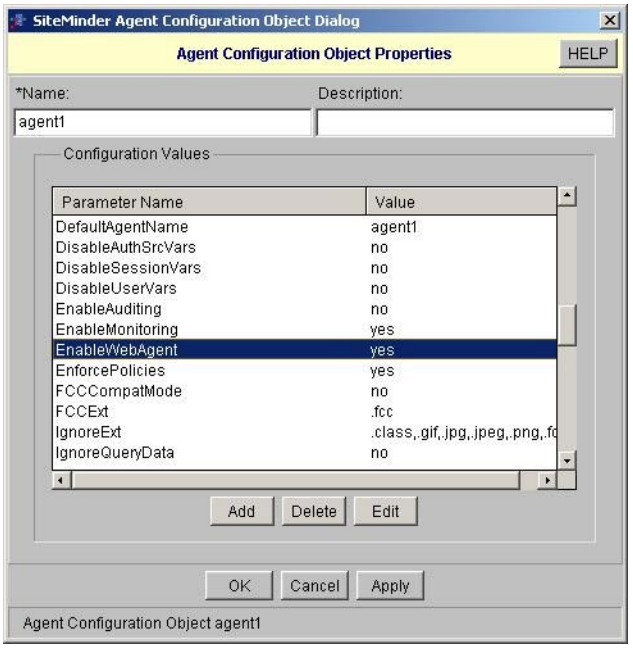
1 **In SiteMinder, create a host configuration object.**

In this example, the host configuration object is named `host1-agent`.



2 Create a web policy agent, and then create an appropriate Agent Configuration Object for the policy agent.

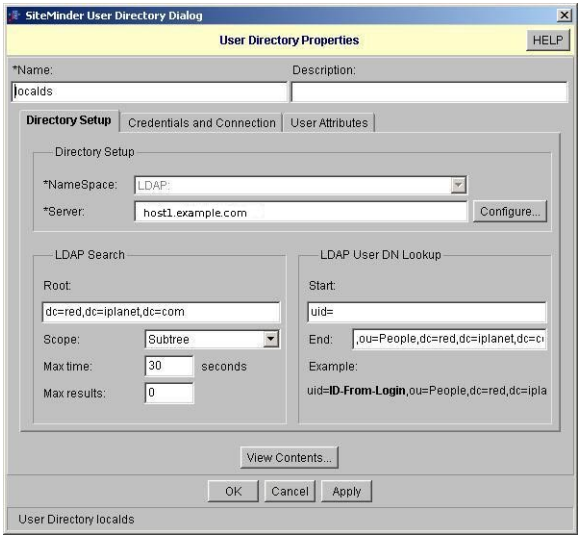
In this example, the Agent Configuration Object is named `agent1`.



▼ To Create and Configure the User Directory

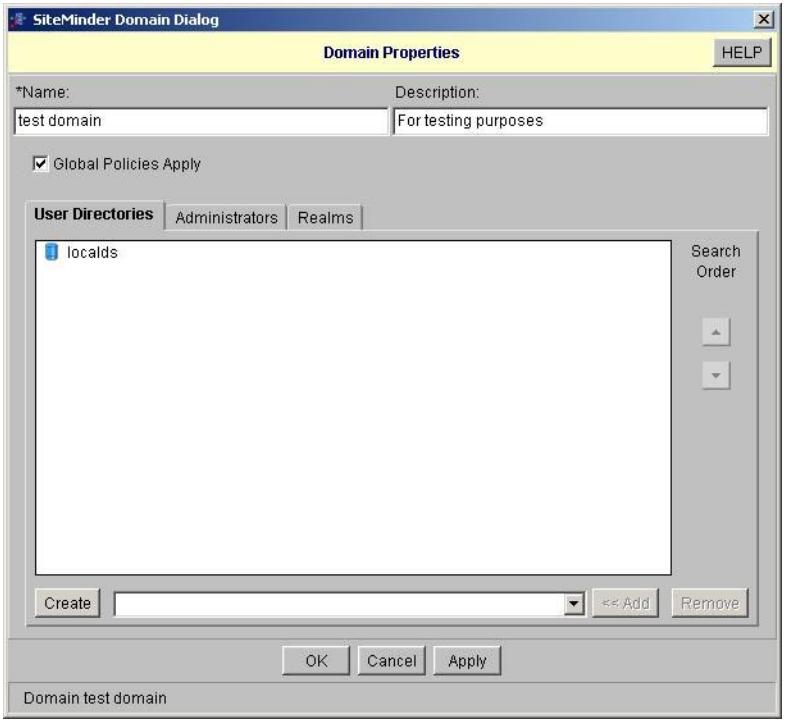
1 Create a user directory.

In this example, the user directory is named `localds`.



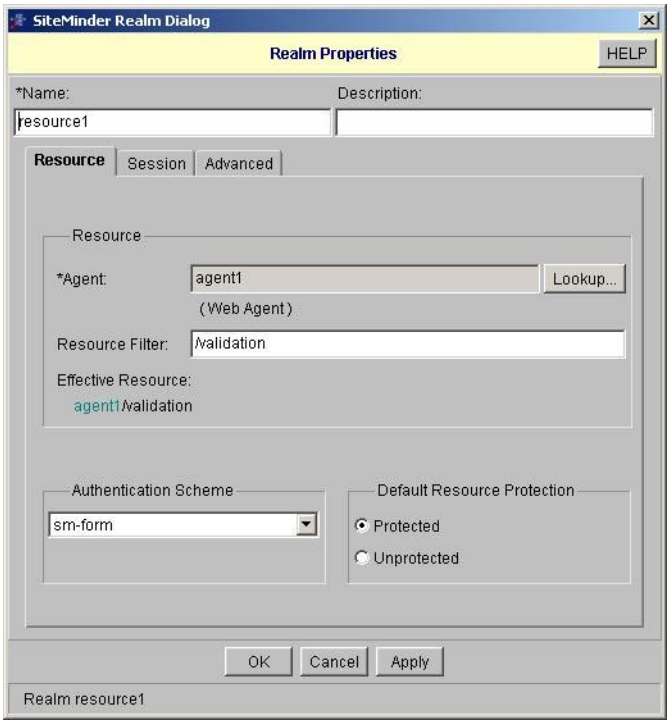
2 Create a domain.

In this example, the domain is named `test domain`. Under User Directories, specify `localds`.



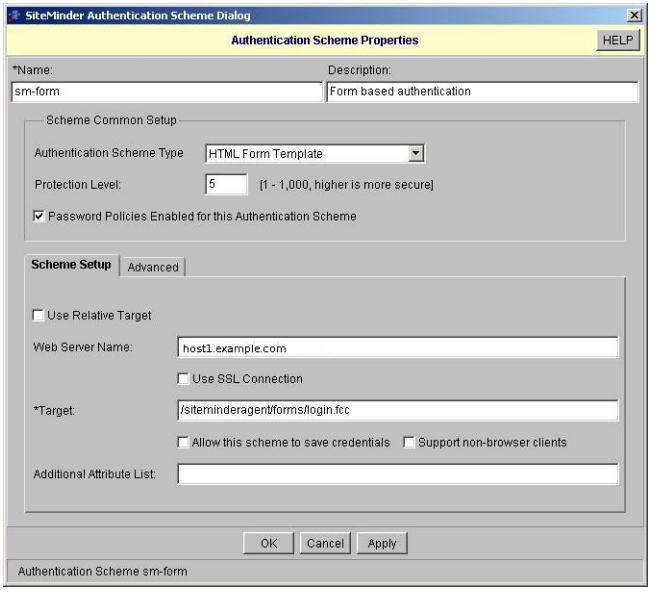
3 Create a realm.

In this example, the new realm is named `resource1`.



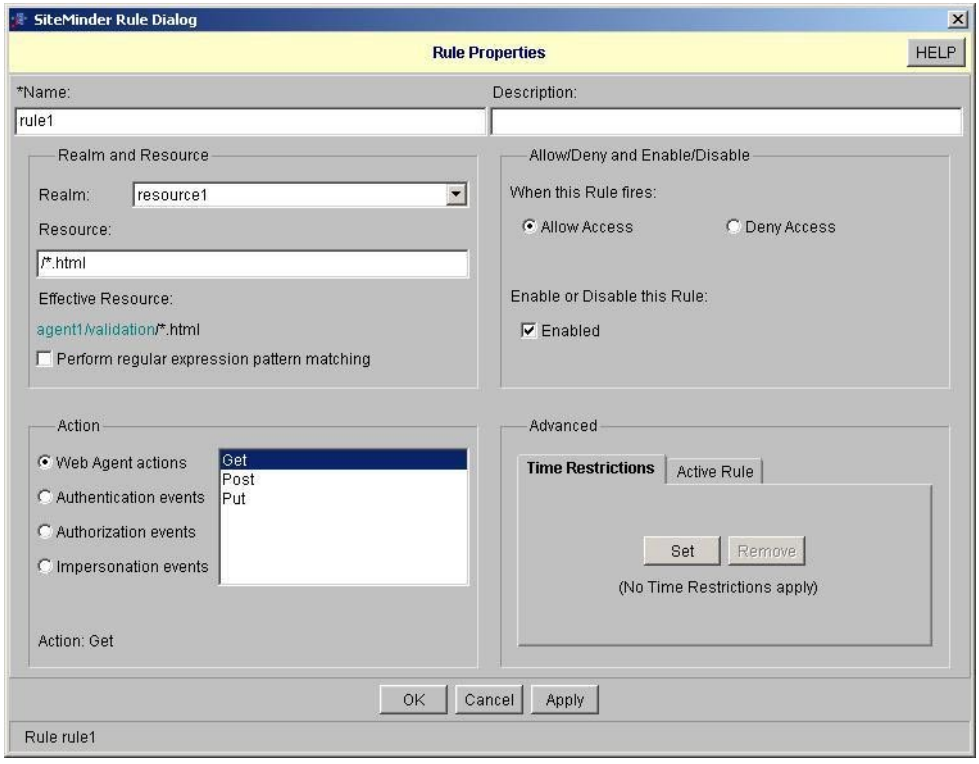
Creating and Configuring a Form-Based Authentication Scheme

In SiteMinder, create a form authentication scheme, and then create a configuration for the authentication scheme.



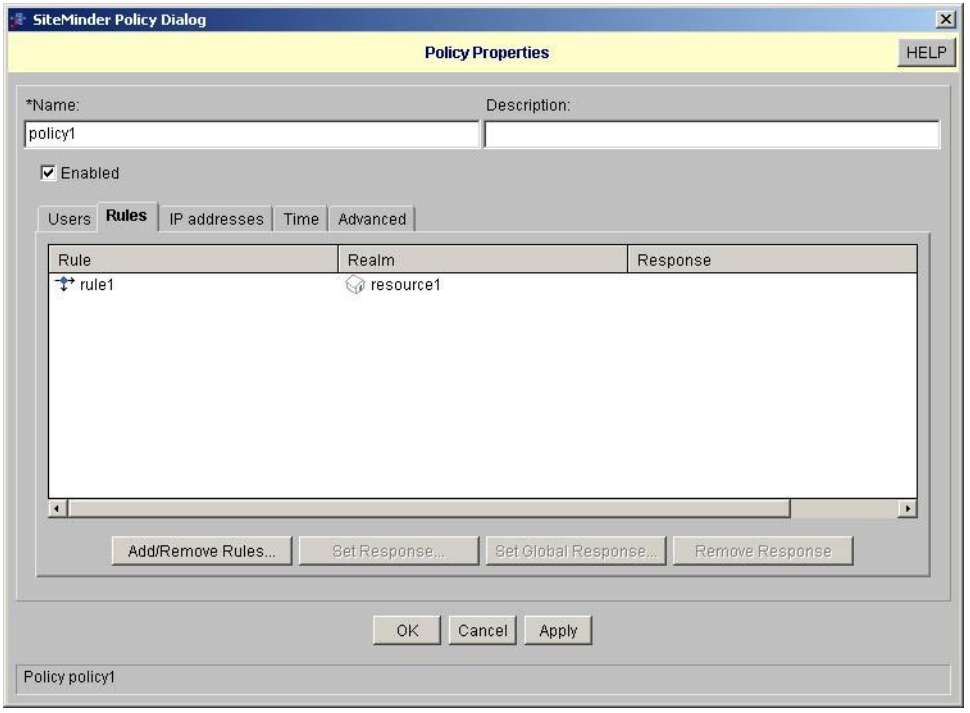
▼ To Create a Policy

1 **Create a rule under the `resource1` realm, and then configure the rule URLs.**
In this example, the new rule is named `rule1`.



2 Create a policy.

In this example, the new policy is named `policy1`.



3 Assign the users and add the rules to the policy.

Using OpenSSO Enterprise to Enable SiteMinder Federation in an Identity Provider Environment

The following is a high-level overview of the sequence you must follow to enable SiteMinder with OpenSSO Enterprise in an Identity Provider Environment:

1. Install the Principal Components.
2. Configure the Identity Provider OpenSSO Enterprise to Use SAMLv2 Identity Provider Protocols.
3. Configure the SiteMinder Agent to Protect OpenSSO Enterprise URLs.
4. Install the OpenSSO Enterprise Policy Agent in the Service Provider.
5. Verify that Single Sign-On is Working Properly.
6. Review Sample Identity Provider Interactions.

▼ To Install the Principal Components

The following are the principal components in this use case:

- OpenSSO Enterprise in the Identity Provider container
- SiteMinder Web Agent
- SiteMinder custom authentication module
- OpenSSO Enterprise in the Service Provider container

Before You Begin

- The Identity Provider and Service Provider should be in installed in different domains. If this is not possible, they should minimally use different cookie names or cookie domains.
- You can defer the installation of OpenSSO Enterprise policy agent for protecting the OpenSSO Enterprise Service Provider until the end of the installation procedures. This gives you the opportunity to verify that the SAML2 setup is working before you proceed.
- Before proceeding, be sure to read the general instructions in "Installing SiteMinder" on page 165 and in "Configuring SiteMinder After Installation" on page 166. The following steps provide additional installation information specific only to this use case.

1 Install and configure OpenSSO Enterprise in the same container in which the Identity Provider is installed.

For detailed installation instructions, see the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.

- Be sure that the Identity Provider container supports SiteMinder Web Agent installation.
- Configure OpenSSO Enterprise to use the same user repository as the SiteMinder user repository. This enables both OpenSSO Enterprise and SiteMinder to provide a single session for the same user.

2 Install and configure the SiteMinder Web Agent on the OpenSSO Enterprise container.

For now, configure the SiteMinder Web Agent to protect an arbitrary URL on the container. In this example, the protected URL is `/validation/index.html`.

- As in the previous section, create a context root `/validation`, or create a directory named `validation` under the `docroot`.
- Be sure that the SiteMinder form authentication scheme is working for the protected URL.

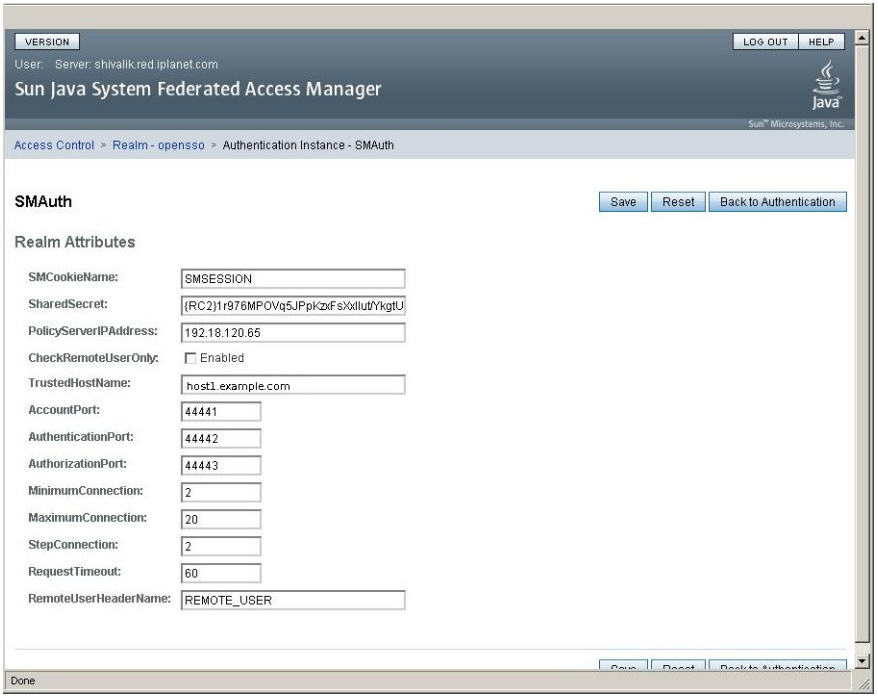
3 Install the SiteMinder custom authentication module in OpenSSO Enterprise.

After you unzip the OpenSSO Enterprise binary, the SiteMinder custom authentication module is located under the directory *unzip-directory*`/integrations/siteminder/`. The `README.html` provides steps for building a custom authentication module. The following parameters must be set to enable the SiteMinder SDK to connect to the SiteMinder Policy Server:

SMCookieName:	SiteMinder cookie name. The default name is `SMSESSION`.
SharedSecret:	Unique policy agent configuration obtained from SiteMinder, and used by OpenSSO Enterprise to point to the SiteMinder SDK .
PolicyServerIPAddress:	Indicates where the SiteMinder Policy Server is located.
CheckRemoteUserOnly:	This attribute should be enabled when the SiteMinder Web Agent is installed on the same host as OpenSSO Enterprise. The SiteMinder Web Agent performs session validation. When this attribute is enabled, the rest of the configuration is not needed.
TrustedHostName:	Name of the SiteMinder SDK host name.
AccountPort	One of 3 TCP ports used by the SiteMinder Server to connect to the SiteMinder SDK.
AuthenticationPort:	One of 3 TCP ports used by the SiteMinder Server to connect to the SiteMinder SDK.
AuthorizationPort:	One of 3 TCP ports used by the SiteMinder Server to connect to the SiteMinder SDK.
MinimumConnection:	In a connection pool implementation, the maximum number of concurrent connections that a can be opened.
MaximumConnection:	In a connection pool implementation, the minimum number of concurrent connections that a can be opened.
StepConnection:	In a connection pool implementation, the number of concurrent connections that can be opened.
RequestTimeout:	Maximum time that the SiteMinder SDKwaits before it connects to SiteMinder Policy Server.

RemoteUserHeaderName: When configured, the SiteMinder Web Agent sets a header name for the remote user after successful authentication. This parameter is used only when the `checkRemoteHeaderOnly` flag is set. The `SMAuth` module uses this parameter to create an OpenSSO Enterprise session.

The following diagram shows an example of SiteMinder custom authentication module configuration.



4 Install and configure OpenSSO Enterprise in the container in which the Service Provider is installed.

For detailed installation instructions, see the OpenSSO Enterprise Installation and Configuration Guide.

5 Install the SiteMinder Web Agent in the OpenSSO Enterprise container.

See the SiteMinder product documentation.

▼ To Configure the Identity Provider OpenSSO Enterprise to Use SAMLv2 Identity Provider Protocols

Before you can enable the SAMLv2 Identity Provider protocols, you must generate, customize, and load each of the following:

- Identity Provider metadata
- Identity Provider extended metadata
- Service Provider metadata
- Service Provider extended metadata.

Before You Begin

- Read through the following instructions for the changes that you must make to the default metadata. The SAML2 samples contain instructions on how to setup SAML2.
- You must import Identity Provider metadata and Identity Provider extended metadata as hosted metadata. You must import Service Provider metadata and Service Provider extended metadata as remote entity metadata. To change a configuration from the default `hosted` to `remote`, modify the extended metadata XML element `<EntityConfig>`. Change the default attribute `hosted=true` to `hosted=false`.
- See the OpenSSO Enterprise product documentation for commands and syntax.

1 Generate the metadata templates in both Identity Provider and Service Provider environments.

Use the `famadm` command. You can also use the browser-based interface at the following URL:

`http:`*host:port*`/opensso/famadm.jsp`

- At Identity Provider :

```
famadm create-metadata-templ -y idp_entity_id -u amadmin
                    -f admin_password_file_name -m idp_standard_metadata
                    -x idp_extended_metadata -i idp_meta_alias
```

where `idp_meta_alias` is `/idp`

- At Service Provider:

```
famadm create-metadata-templ -y sp_entity_id -u amadmin
                   -f admin_password_file_name -m sp_standard_metadata
                   -x sp_extended_metadata -s sp_meta_alias
```

where `sp_meta_alias` is `/sp`

2 Customize Identity Provider and Service Provider extended metadata.

The Identity Provider extended metadata should be added as an attribute named `AuthUrl`. This URL attribute is used by the SAML protocols to redirect for authentication purposes. In the following example, `AuthUrl`redirects to the SiteMinder authentication module.

```
<Attribute name="AuthUrl">
            <Value>http://host:port/opensso/UI/Login?module=SMAuth</Value>
       </Attribute>
```

Another option is to make the SiteMinder custom authentication module the default login module in OpenSSO Enterprise. The cost of using this option is that you must specify an LDAP login module for logging in as an administrator.

The Service Provider extended metadata uses the attribute named `transientUser`. Set this value to your anonymous user:

```
 <Attribute name="transientUser">
            <Value>anonymous</Value>
        </Attribute>
```

3 Load the Identity Provider and Service Provider metadata.

First create a Circle of Trust as mentioned in the URL. The Circle of Trust should also be added in the extended metadata.

In your extended template files, you will see a sample Circle of Trust. Modify the sample to create your Circle of Trust.

```
<Attribute name="cotlist">
            <Value>samplesaml2cot</Value>
       </Attribute>
```

Load the hosted metadata in both the Identity Provider and the Service Provider using the `famadm` command or through OpenSSO Enterprise administration console.

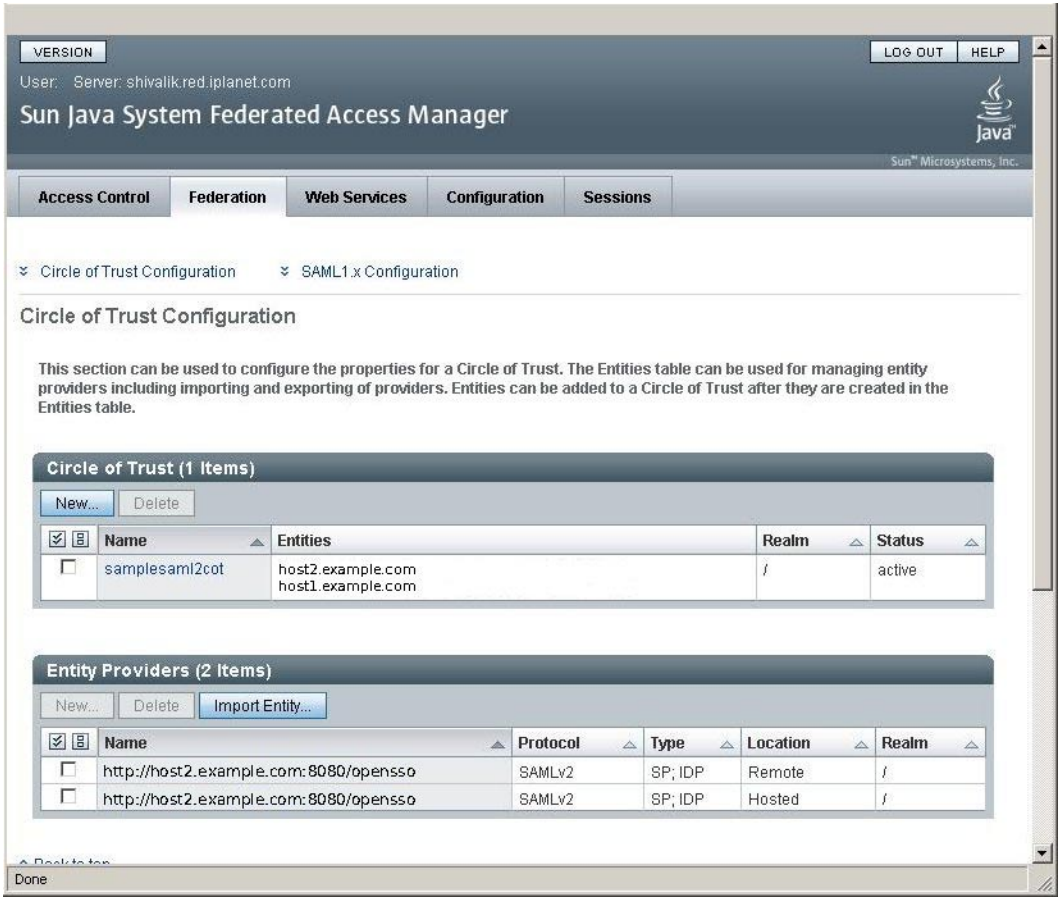
4 Exchange the metadata Service Provider with the Identity Provider metadata.

5 Exchange the Identity Provider metadata with the Service Provider.

6 Load all metadata.

7 After successful metadata exchange, verify through the OpenSSO Enterprise administration console that SAMLv2 is working properly.

The following shows a sample UI for SAMLv2 configuration.

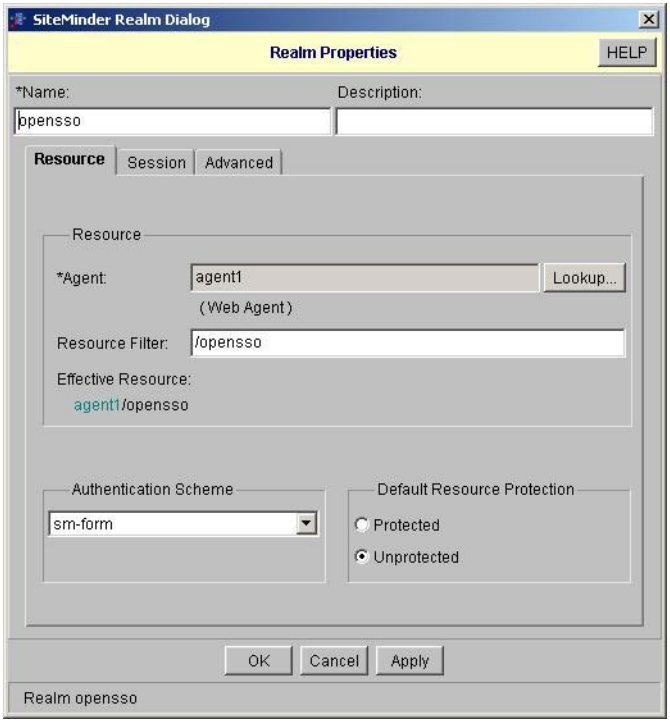


▼ To Configure the SiteMinder Agent to Protect OpenSSO Enterprise URLs

This configuration protects the SAML Single Sign-On Service URL so that the SiteMinder session must be established before the SAML assertion is generated.

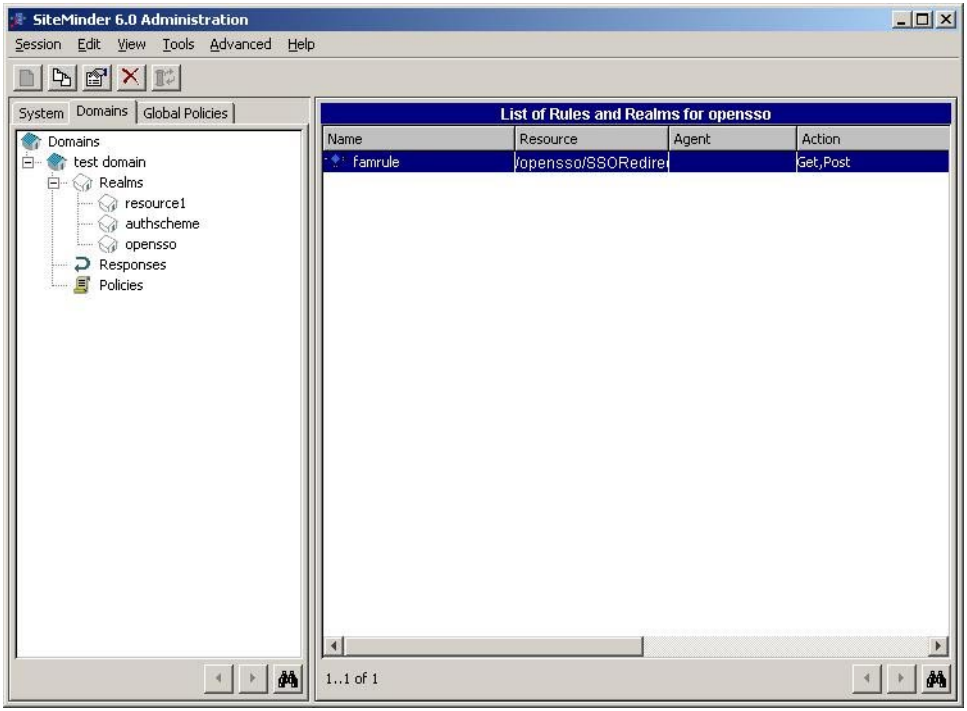
1 **In the SiteMinder administration console, create a new realm in unprotected mode.**

In this example, the realm is named `opensso`.



2 **Create a rule that protects only the SAML2 SSO URL.**

Other URLs are unprotected for now.



Installing the OpenSSO Enterprise Policy Agent

The policy agent must be supported on the container where the enterprise application is deployed. For detailed installation information, see the policy agent documentation.

Change the policy agent login URL to the OpenSSO Enterprise SAML2 Service Provider-initiated Single Sign-on Service URL. Example:

```
http://<sphost>:<spport>/opensso/saml2/jsp/spSSOInit.jsp?metaAlias=<Service
Provider MetaAlias> &idpEntityID=<Identity Provider Entity
ID>&NameIDFormat=transient
```

▼ To Verify that Single Sign-On is Working Properly

1 **Authenticate at the SiteMinder login page using user name and password.**

2 **Access the enterprise application in the Service Provider environment.**

The enterprise application is protected by OpenSSO Enterprise Service Provider Agent. The agent should allow access to the application.

Sample Identity Provider Interactions

1. Access the SM Agent protected application

```
http://HostName.example.com:9898/validation/index.html

GET /validation/index.html HTTP/1.1
Host: HostName.example.com:9898
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,
text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive

HTTP/1.x 302 Moved Temporarily
Server: Netscape-Enterprise/6.0
Date: Fri, 01 Feb 2008 23:46:12 GMT
Cache-Control: no-cache
Location: http://HostName.example.com:9898/SiteMinderagent/forms/
login.fcc?TYPE=33554433&REALMOID=06-1716e557-15f3-100f-b9a4-835cc8200cb3&GUID=
&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=$SM$sHjbzl4f9R%2bcSa0%2fEgnu6oUQQPMQnUg
kU6Zvx5zWZpQ%3d&TARGET=$SM$http%3a%2f%2fshivalik%2ered%2eiplanet%2ecom%3a9898%
2fvalidation%2findex%2ehtml
Connection: close
```

2. SiteMinder authentication

```
http://HostName.example.com:9898/SiteMinderagent/forms/login.fcc?TYPE=
33554433&REALMOID=06-1716e557-15f3-100f-b9a4-835cc8200cb3&GUID=&SMAUTHREASON=
0&METHOD=GET&SMAGENTNAME=$SM$sHjbzl4f9R%2bcSa0%2fEgnu6oUQQPMQnUgkU6Zvx5zWZpQ%
3d&TARGET=$SM$http%3a%2f%2fshivalik%2ered%2eiplanet%2ecom%3a9898%2fvalidation%
2findex%2ehtml

GET /SiteMinderagent/forms/login.fcc?TYPE=33554433&REALMOID=06-1716e557-15f3-
100f-b9a4-835cc8200cb3&GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=$SM$sHjbzl4
f9R%2bcSa0%2fEgnu6oUQQPMQnUgkU6Zvx5zWZpQ%3d&TARGET=$SM$http%3a%2f%2fshivalik%2
ered%2eiplanet%2ecom%3a9898%2fvalidation%2findex%2ehtml HTTP/1.1
Host: HostName.example.com:9898
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,
text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive

HTTP/1.x 200 OK
Server: Netscape-Enterprise/6.0
Date: Fri, 01 Feb 2008 23:46:12 GMT
Content-Type: text/html; charset=ISO-8859-1
Connection: close
------------------------------------------------------------
http://HostName.example.com:9898/SiteMinderagent/forms/login.fcc?TYPE=
33554433&REALMOID=06-1716e557-15f3-100f-b9a4-835cc8200cb3&GUID=&SMAUTHREASON=
0&METHOD=GET&SMAGENTNAME=$SM$sHjbzl4f9R%2bcSa0%2fEgnu6oUQQPMQnUgkU6Zvx5zWZpQ%
3d&TARGET=$SM$http%3a%2f%2fshivalik%2ered%2eiplanet%2ecom%3a9898%2fvalidation%
2findex%2ehtml

POST /SiteMinderagent/forms/login.fcc?TYPE=33554433&REALMOID=06-1716e557-15f3-
100f-b9a4-835cc8200cb3&GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=$SM$sHjbzl4
f9R%2bcSa0%2fEgnu6oUQQPMQnUgkU6Zvx5zWZpQ%3d&TARGET=$SM$http%3a%2f%2fshivalik%
2ered%2eiplanet%2ecom%3a9898%2fvalidation%2findex%2ehtml HTTP/1.1
Host: HostName.example.com:9898
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,
text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
```

```
Connection: keep-alive
Referer: http://HostName.example.com:9898/SiteMinderagent/forms/
login.fcc?TYPE=33554433&REALMOID=06-1716e557-15f3-100f-b9a4-835cc8200cb3&
GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=$SM$sHjbzl4f9R%2bcSa0%
2fEgnu6oUQQPMQnUgkU6Zvx5zWZpQ%3d&TARGET=$SM$http%3a%2f%2fshivalik%2ered%
2eiplanet%2ecom%3a9898%2fvalidation%2findex%2ehtml
Content-Type: application/x-www-form-urlencoded
Content-Length: 233
SMENC=ISO-8859-1&SMLOCALE=US-EN&USER=test&PASSWORD=test&target=http%
3A%2F%2FHostName.example.com%3A9898%2Fvalidation%
2Findex.html&smauthreason=0&smagentname=sHjbzl4f9R%2BcSa0%
2FEgnu6oUQQPMQnUgkU6Zvx5zWZpQ%3D&postpreservationdata=
HTTP/1.x 302 Moved Temporarily
Server: Netscape-Enterprise/6.0
Date: Fri, 01 Feb 2008 23:46:18 GMT
Content-Type: magnus-internal/fcc
Set-Cookie: SMSESSION=2xm2Iw6fTMBcjA6rlK/YUY1CRBudYxwOCkfpCo95YKAp2b4ZzLOPT
qi2S14CQ7nRja+fUq53Aj0pmTxDvPKTMcKD1Ql1hGx0gPK7xx2eqMP3IyTAK3qNahRgt7mQRTIB
BDEE0rOJcpgrMRtsteC90yMdiJrrEeqfC38utU6mxO9BejwjRuGN2rmf9WM4Odl+4TE0iUOiP/k
iCR6sn2r03GBsbBjOi12oSlh/4JAyfOwxsgBJCwDiZVlFXNiKNaKdY1UQr8OcKeO33eNn3w9RW9
ZrjRibQTQcxxmiR+gsvAuM8etEzP6GCFKjc1s8I3DNuSBbDqfyt81YUSYdEYa9UKfvvOJplZOIT
BkQajcAEPOq+vTYxQ4BH2RmjdPMVcIxRm2bibM9QtuQD83C9QubTk1lq4j+ywPsvutiYEoGHV+7
6VXws5NsvhK2gH4ZTC0xsd76X2/1no8xMv9c3W4DcSp9cQQ74/7+a7gzT+hxQSpyQFf4mDTnq/D
XS5V7tcLS0EyFcf8RwSbvDPnICiebR3vtZgHRL1kEZheEh9ToHmwqIO9cCqz9rJXR7/NL+o/AQr
7M4o+LyA7KxozAueUj0pg8GINteUGVxMLWmR7Xm/Lp0pI9DjM5mfbmP8Ka+w0T6H9LHNlQGaYZA
PCkeABAXqLb8q8yJUzPdI0BVlp1awNCx579DereoCIzCZdQ99rVDSQUS77KCQATnYXrHqTxqbXxW
beDf6gk9ZCf29XTzO8hBLdScqGOBX1OvDvzdghcjHnupQf1fYltt/3MrZ/Jrxonbpgxg4C5zVgSU
PrNqb66RYWQOelZXooh7lTPoFHsMFodVnecsOZmEMXNI8DB08pyo5KhRZJk2Mr4o3rPNtiHPpnXc
d+imapuosG3FwF5Sv6flh8jbiE9/MZdIQ06hgWEIiCnUEYdboli4TWgy0/QpCbdJ7OviU275VZiC
W6hMTRyrxnEvoQ=; path=/; domain=.red.example.com
Cache-Control: no-cache
Location: http://HostName.example.com:9898/validation/index.html
Connection: close
-------------------------------------------------------------
http://HostName.example.com:9898/validation/index.html
GET /validation/index.html HTTP/1.1
Host: HostName.example.com:9898
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,
text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://HostName.example.com:9898/SiteMinderagent/forms/
```

```
login.fcc?TYPE=33554433&REALMOID=06-1716e557-15f3-100f-b9a4-835cc8200cb3&GUID=
&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=$SM$sHjbzl4f9R%2bcSa0%2fEgnu6oUQQPMQnUg
kU6Zvx5zWZpQ%3d&TARGET=$SM$http%3a%2f%2fshivalik%2ered%2eiplanet%2ecom%3a9898%
2fvalidation%2findex%2ehtml
Cookie: SMSESSION=2xm2Iw6fTMBcjA6rlK/YUY1CRBudYxwOCkfpCo95YKAp2b4ZzLOPTqi2S14
CQ7nRja+fUq53Aj0pmTxDvPKTMcKD1Ql1hGx0gPK7xx2eqMP3IyTAK3qNahRgt7mQRTIBBDEE0rOJ
cpgrMRtsteC90yMdiJrrEeqfC38utU6mxO9BejwjRuGN2rmf9WM4Odl+4TE0iUOiP/kiCR6sn2r03
GBsbBjOi12oSlh/4JAyfOwxsgBJCwDiZVlFXNiKNaKdY1UQr8OcKeO33eNn3w9RW9ZrjRibQTQcxx
miR+gsvAuM8etEzP6GCFKjc1s8I3DNuSBbDqfyt81YUSYdEYa9UKfvvOJplZOITBkQajcAEPOq+vT
YxQ4BH2RmjdPMVcIxRm2bibM9QtuQD83C9QubTk1lq4j+ywPsvutiYEoGHV+76VXws5NsvhK2gH4Z
TC0xsd76X2/1no8xMv9c3W4DcSp9cQQ74/7+a7gzT+hxQSpyQFf4mDTnq/DXS5V7tcLS0EyFcf8Rw
SbvDPnICiebR3vtZgHRL1kEZheEh9ToHmwqIO9cCqz9rJXR7/NL+o/AQr7M4o+LyA7KxozAueUj0p
g8GINteUGVxMLWmR7Xm/Lp0pI9DjM5mfbmP8Ka+w0T6H9LHNlQGaYZAPCkeABAXqLb8q8yJUzPdI0
BVlp1awNCx579DereoCIzCZdQ99rVDSQUS77KCQATnYXrHqTxqbXxWbeDf6gk9ZCf29XTzO8hBLdS
cqGOBX1OvDvzdghcjHnupQf1fYltt/3MrZ/Jrxonbpgxg4C5zVgSUPrNqb66RYWQOelZXooh7lTPo
FHsMFodVnecsOZmEMXNI8DB08pyo5KhRZJk2Mr4o3rPNtiHPpnXcd+imapuosG3FwF5Sv6flh8jbi
E9/MZdIQ06hgWEIiCnUEYdboli4TWgy0/QpCbdJ7OviU275VZiCW6hMTRyrxnEvoQ=

HTTP/1.x 200 OK
Server: Netscape-Enterprise/6.0
Date: Fri, 01 Feb 2008 23:46:18 GMT
Set-Cookie: SMSESSION=jlO0TgMQfglpU+GHQCJqbnoE2Pevax6fdzPGU7ZAgJuPb/fxTjCbWX1
B1RO6QaLJn6VoVGNK8Sy6IeILAyv+LciS/OMK1E0tSXnL5Uvit3XIuWuiSMuklyDMIlOQ6n3ZSGGr
9sKBUch5YVfGcfGjHQFcBIlzegQxBRrgH/l2rc8aTEHdCrprvBiRHwQlxJbrcWMqfJw7h+HUEtiz9
bQCUkwMbpEW4eBfNyRlZTGov3K5hg4HK4tuoyvOeKdZaewlTB4Lm+QeGWo2qv2mPDP+eVtBiVtRVH
HTHGfSthTJYQOOc4rPV2dnl8axpWppGByeUmfmeService Provider9x5hVxDi91iyobTybKpDz0
bltkvnHbqwbLfehUPtJFxS3Z54y9dmiuoQ+B5Kdrs7DNuvrnAI1ZQdDKQEVA4Pt+vA9KO18ah9V1I
7BZ9D/x60uWxfaA3Ty8lRgWhMYqdBulFMD1B29sxboNHWdJ2FaxQJGjMpSEZ5iHB50ovF4YFXRyPP
5Tl7eJxIebLKX02LFrG/osNZ9UKHrMY1MRK5WWHJlYB040ADVcTNrFkc39vcYIA1eGDYhC/NaOd41
2HP5S0UX0/59ADMLBsX/qBjcdODy3li+4eZnK1oHw/9yr3LCjewJ+H9w0k0/dQw99vgwEM2RPFgH5
Y7W6k6h1efp67VKXLBiJ1OZPJe2SCEDAOUla8qsC8fQ0VWTy/TfVhVtqJOaSLZrACX7uhPzbZE1EA
Pd8x7UeJquFll3WpdnZYObd0DQLeoWZcF2rPIcfBn+8X8oig5KzvAgQ9R8MR+h7OkYfhmwwBDaQkb
KPpIxjpeLNxKpkEVWJ9HoHOpZ/txCQUAHqPV41YjZ6CQfBfUqdOHbfje9O+0pJ1aHMntI4VYZOqdx
sA+n9cgKjNQ8ruHOqSKhAQfEgipwcM2fMU3Uqmtr+0/+5bi7Cbs=; path=/;
domain=.red.example.com
Content-Type: text/html
Etag: "dcea10a4-1-0-88"
Last-Modified: Thu, 10 Jan 2008 01:42:07 GMT
Content-Length: 136
Accept-Ranges: bytes
-----------------------------------------------------------
```

3. SAML Service Provider SSO initiation

```
http://ide-13.red.example.com:8080/opensso/saml2/jsp/spSSOInit.jsp?
metaAlias=/sp&idpEntityID=
http://HostName.example.com:8080/opensso&NameIDFormat=transient
```

```
GET /opensso/saml2/jsp/spSSOInit.jsp?metaAlias=/sp&idpEntityID=
http://HostName.example.com:8080/opensso&NameIDFormat=transient HTTP/1.1
Host: ide-13.red.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,
text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: SMSESSION=jlO0TgMQfglpU+GHQCJqbnoE2Pevax6fdzPGU7ZAgJuPb/fxTj
CbWX1B1RO6QaLJn6VoVGNK8Sy6IeILAyv+LciS/OMK1E0tSXnL5Uvit3XIuWuiSMukly
DMIlOQ6n3ZSGGr9sKBUch5YVfGcfGjHQFcBIlzegQxBRrgH/l2rc8aTEHdCrprvBiRHw
QlxJbrcWMqfJw7h+HUEtiz9bQCUkwMbpEW4eBfNyRlZTGov3K5hg4HK4tuoyvOeKdZae
wlTB4Lm+QeGWo2qv2mPDP+eVtBiVtRVHHTHGfSthTJYQOOc4rPV2dnl8axpWppGByeUm
fmeService Provider9x5hVxDi91iyobTybKpDz0bltkvnHbqwbLfehUPtJFxS3Z54y9
dmiuoQ+B5Kdrs7DNuvrnAI1ZQdDKQEVA4Pt+vA9KO18ah9V1I7BZ9D/x60uWxfaA3Ty8l
RgWhMYqdBulFMD1B29sxboNHWdJ2FaxQJGjMpSEZ5iHB50ovF4YFXRyPP5Tl7eJxIebLK
X02LFrG/osNZ9UKHrMY1MRK5WWHJlYB040ADVcTNrFkc39vcYIA1eGDYhC/NaOd412HP5
S0UX0/59ADMLBsX/qBjcdODy3li+4eZnK1oHw/9yr3LCjewJ+H9w0k0/dQw99vgwEM2RP
FgH5Y7W6k6h1efp67VKXLBiJ1OZPJe2SCEDAOUla8qsC8fQ0VWTy/TfVhVtqJOaSLZrAC
X7uhPzbZE1EAPd8x7UeJquFll3WpdnZYObd0DQLeoWZcF2rPIcfBn+8X8oig5KzvAgQ9R8
MR+h7OkYfhmwwBDaQkbKPpIxjpeLNxKpkEVWJ9HoHOpZ/txCQUAHqPV41YjZ6CQfBfUqd
OHbfje9O+0pJ1aHMntI4VYZOqdxsA+n9cgKjNQ8ruHOqSKhAQfEgipwcM2fMU3Uqmtr+0
/+5bi7Cbs=

HTTP/1.x 302 Moved Temporarily
X-Powered-By: JService Provider/2.1
Server: Sun Java System Application Server 9.1
Set-Cookie: JSESSIONID=765d4c266461607b4b55811d34ca; Path=/opensso
Location: http://HostName.example.com:8080/opensso/SSORedirect/
metaAlias/idp?SAMLRequest=nVTNjtowEL7vU0S%2BQ5ywXcACJAqqirRtKbA99Gac
SbHq2KlnwtK3rx1YRNUV2nIdj2e%2BP3uEsjK1mDa0syv41QBSkhwqY1G0J2PWeCucRI
3CygpQkBLr6adHkXe5qL0jp5xhd4v5mGHeH%2BYPSvF8ONxu4aHkWZaDLPpl2Rts%2B8
PBu16ewZDfK5Z8A4%2Fa2TELY1iyQGxgYZGkpVDifNDheYdnm7wn7vuix7%2BzZB6gaS
upvbUjqkWa4k7vpdE%2Fux6Krq6NtEBd5Sox4AOeuhosokvX6y8rKLQHRWkFJKdGS0x1
UbPkg%2FMKWvJjVkqDELEsJaLew7myPLF8r22h7Y%2FrkmyPTSg%2BbjbLztSTLqUilk
wRwUfwM2exqcCvwe%2B1gqfV45mOLqCT9a6Tebl%2BwQRrNrkbRbtEK6S%2FMPA6WPkC
ik3eDmGUXqw6La7F5zB8MV86o9XvWxIUragkXe%2BOFV10yrZVkJcWNdig7noZ939tQh
hKDf5%2FFA3WGOOeZx4kBc%2FJN8AmR45%2FszpTPb0TKNrgBEMIDnQL5Zmrauk1xkTD
IcZkcrTxcvDMBJdWUN5i6tU2JVQcHcox8M%2FOFzHo4ZFAsYnK1s7TyevX8Jw1elWOIF
b679cy%2BQM%3D
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 0
```

```
Date: Fri, 01 Feb 2008 23:47:30 GMT
------------------------------------------------------------
http://HostName.example.com:8080/opensso/SSORedirect/metaAlias/
idp?SAMLRequest=nVTNjtowEL7vU0S%2BQ5ywXcACJAqqirRtKbA99GacSbHq2Klnw
tK3rx1YRNUV2nIdj2e%2BP3uEsjK1mDa0syv41QBSkhwqY1G0J2PWeCucRI3CygpQkB
Lr6adHkXe5qL0jp5xhd4v5mGHeH%2BYPSvF8ONxu4aHkWZaDLPpl2Rts%2B8PBu16ew
ZDfK5Z8A4%2Fa2TELY1iyQGxgYZGkpVDifNDheYdnm7wn7vuix7%2BzZB6gaSupvbUj
qkWa4k7vpdE%2Fux6Krq6NtEBd5Sox4AOeuhosokvX6y8rKLQHRWkFJKdGS0x1UbPkg
%2FMKWvJjVkqDELEsJaLew7myPLF8r22h7Y%2FrkmyPTSg%2BbjbLztSTLqUilkwRwU
fwM2exqcCvwe%2B1gqfV45mOLqCT9a6Tebl%2BwQRrNrkbRbtEK6S%2FMPA6WPkCik3
eDmGUXqw6La7F5zB8MV86o9XvWxIUragkXe%2BOFV10yrZVkJcWNdig7noZ939tQhhK
Df5%2FFA3WGOOeZx4kBc%2FJN8AmR45%2FszpTPb0TKNrgBEMIDnQL5Zmrauk1xkTDI
cZkcrTxcvDMBJdWUN5i6tU2JVQcHcox8M%2FOFzHo4ZFAsYnK1s7TyevX8Jw1elWOIF
b679cy%2BQM%3D

GET /opensso/SSORedirect/metaAlias/idp?SAMLRequest=nVTNjtowEL7vU0S%2
BQ5ywXcACJAqqirRtKbA99GacSbHq2KlnwtK3rx1YRNUV2nIdj2e%2BP3uEsjK1mDa0s
yv41QBSkhwqY1G0J2PWeCucRI3CygpQkBLr6adHkXe5qL0jp5xhd4v5mGHeH%2BYPSvF
8ONxu4aHkWZaDLPpl2Rts%2B8PBu16ewZDfK5Z8A4%2Fa2TELY1iyQGxgYZGkpVDifND
heYdnm7wn7vuix7%2BzZB6gaSupvbUjqkWa4k7vpdE%2Fux6Krq6NtEBd5Sox4AOeuho
sokvX6y8rKLQHRWkFJKdGS0x1UbPkg%2FMKWvJjVkqDELEsJaLew7myPLF8r22h7Y%2F
rkmyPTSg%2BbjbLztSTLqUilkwRwUfwM2exqcCvwe%2B1gqfV45mOLqCT9a6Tebl%2Bw
QRrNrkbRbtEK6S%2FMPA6WPkCik3eDmGUXqw6La7F5zB8MV86o9XvWxIUragkXe%2BOF
V10yrZVkJcWNdig7noZ939tQhhKDf5%2FFA3WGOOeZx4kBc%2FJN8AmR45%2FszpTPb0
TKNrgBEMIDnQL5Zmrauk1xkTDIcZkcrTxcvDMBJdWUN5i6tU2JVQcHcox8M%2FOFzHo4
ZFAsYnK1s7TyevX8Jw1elWOIFb679cy%2BQM%3D HTTP/1.1
Host: HostName.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: SMSESSION=jlO0TgMQfglpU+GHQCJqbnoE2Pevax6fdzPGU7ZAgJuPb/fxTjC
bWX1B1RO6QaLJn6VoVGNK8Sy6IeILAyv+LciS/OMK1E0tSXnL5Uvit3XIuWuiSMuklyDM
IlOQ6n3ZSGGr9sKBUch5YVfGcfGjHQFcBIlzegQxBRrgH/l2rc8aTEHdCrprvBiRHwQlx
JbrcWMqfJw7h+HUEtiz9bQCUkwMbpEW4eBfNyRlZTGov3K5hg4HK4tuoyvOeKdZaewlTB
4Lm+QeGWo2qv2mPDP+eVtBiVtRVHHTHGfSthTJYQOOc4rPV2dnl8axpWppGByeUmfme
Service Provider9x5hVxDi91iyobTybKpDz0bltkvnHbqwbLfehUPtJFxS3Z54y9dm
iuoQ+B5Kdrs7DNuvrnAI1ZQdDKQEVA4Pt+vA9KO18ah9V1I7BZ9D/x60uWxfaA3Ty8lRg
WhMYqdBulFMD1B29sxboNHWdJ2FaxQJGjMpSEZ5iHB50ovF4YFXRyPP5Tl7eJxIebLKX02
LFrG/osNZ9UKHrMY1MRK5WWHJlYB040ADVcTNrFkc39vcYIA1eGDYhC/NaOd412HP5S0UX
0/59ADMLBsX/qBjcdODy3li+4eZnK1oHw/9yr3LCjewJ+H9w0k0/dQw99vgwEM2RPFgH5Y
7W6k6h1efp67VKXLBiJ1OZPJe2SCEDAOUla8qsC8fQ0VWTy/TfVhVtqJOaSLZrACX7uhPz
```

```
bZE1EAPd8x7UeJquFll3WpdnZYObd0DQLeoWZcF2rPIcfBn+8X8oig5KzvAgQ9R8MR+h7O
kYfhmwwBDaQkbKPpIxjpeLNxKpkEVWJ9HoHOpZ/txCQUAHqPV41YjZ6CQfBfUqdOHbfje9
O+0pJ1aHMntI4VYZOqdxsA+n9cgKjNQ8ruHOqSKhAQfEgipwcM2fMU3Uqmtr+0/+5bi7Cbs=

HTTP/1.x 302 Moved Temporarily
X-Powered-By: JService Provider/2.1
Server: Sun Java System Application Server 9.1
Set-Cookie: JSESSIONID=766be1d1028d55badd1ed0fe34ac; Path=/opensso
Location: http://HostName.example.com:8080/opensso/UI/Login?module=
SMAuth&goto=http%3A%2F%2FHostName.example.com%3A8080%2Fopensso%
2FSSORedirect%2FmetaAlias%2Fidp%3FReqID%3Ds27926cc0299bbe6f0112ead7
ff38b7985321e904c
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 0
Date: Fri, 01 Feb 2008 23:48:30 GMT
------------------------------------------------------------
```

4. Redirection to SiteMinder authentication module in OpenSSO Enterprise

```
http://HostName.example.com:8080/opensso/UI/Login?module=SMAuth&goto=
http%3A%2F%2FHostName.example.com%3A8080%2Fopensso%2FSSORedirect%
2FmetaAlias%2Fidp%3FReqID%3Ds27926cc0299bbe6f0112ead7ff38b7985321e904c

GET /opensso/UI/Login?module=SMAuth&goto=http%3A%2F%2FHostName.example.com%
3A8080%2Fopensso%2FSSORedirect%2FmetaAlias%2Fidp%3FReqID%3Ds27926cc0299bbe6f0112
ead7ff38b7985321e904c HTTP/1.1
Host: HostName.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: JSESSIONID=766be1d1028d55badd1ed0fe34ac; SMSESSION=jlO0TgMQfglpU+GHQ
CJqbnoE2Pevax6fdzPGU7ZAgJuPb/fxTjCbWX1B1RO6QaLJn6VoVGNK8Sy6IeILAyv+LciS/OMK1
E0tSXnL5Uvit3XIuWuiSMuklyDMIlOQ6n3ZSGGr9sKBUch5YVfGcfGjHQFcBIlzegQxBRrgH/l2r
c8aTEHdCrprvBiRHwQlxJbrcWMqfJw7h+HUEtiz9bQCUkwMbpEW4eBfNyRlZTGov3K5hg4HK4tuo
yvOeKdZaewlTB4Lm+QeGWo2qv2mPDP+eVtBiVtRVHHTHGfSthTJYQOOc4rPV2dnl8axpWppGByeU
mfmeService Provider9x5hVxDi91iyobTybKpDz0bltkvnHbqwbLfehUPtJFxS3Z54y9dmiuoQ+
B5Kdrs7DNuvrnAI1ZQdDKQEVA4Pt+vA9KO18ah9V1I7BZ9D/x60uWxfaA3Ty8lRgWhMYqdBulFMD
1B29sxboNHWdJ2FaxQJGjMpSEZ5iHB50ovF4YFXRyPP5Tl7eJxIebLKX02LFrG/osNZ9UKHrMY1M
RK5WWHJlYB040ADVcTNrFkc39vcYIA1eGDYhC/NaOd412HP5S0UX0/59ADMLBsX/qBjcdODy3li+
4eZnK1oHw/9yr3LCjewJ+H9w0k0/dQw99vgwEM2RPFgH5Y7W6k6h1efp67VKXLBiJ1OZPJe2SCED
```

```
AOUla8qsC8fQ0VWTy/TfVhVtqJOaSLZrACX7uhPzbZE1EAPd8x7UeJquFll3WpdnZYObd0DQLeoW
ZcF2rPIcfBn+8X8oig5KzvAgQ9R8MR+h7OkYfhmwwBDaQkbKPpIxjpeLNxKpkEVWJ9HoHOpZ/txC
QUAHqPV41YjZ6CQfBfUqdOHbfje9O+0pJ1aHMntI4VYZOqdxsA+n9cgKjNQ8ruHOqSKhAQfEgipw
cM2fMU3Uqmtr+0/+5bi7Cbs=

HTTP/1.x 302 Moved Temporarily
X-Powered-By: Servlet/2.5
Server: Sun Java System Application Server 9.1
Cache-Control: private
Pragma: no-cache
Expires: 0
X-DSAMEVersion: 8.0 (2007-November-29 01:17)
AM_CLIENT_TYPE: genericHTML
Set-Cookie: AMAuthCookie=AQIC5wM2LY4SfczvfJJpn1IfT3pStks2VjzPMebgYVAxtyE=
@AAJTSQACMDE=#; Domain=HostName.example.com; Path=/
Set-Cookie: amlbcookie=01; Domain=HostName.example.com; Path=/
Set-Cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfczvfJJpn1IfT3pStks2VjzPMebgYVAxtyE=
@AAJTSQACMDE=#; Domain=HostName.example.com; Path=/
Set-Cookie: AMAuthCookie=LOGOUT; Domain=HostName.example.com;
Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/
X-AuthErrorCode: 0
Location: http://HostName.example.com:8080/opensso/SSORedirect/metaAlias/
idp?ReqID=s27926cc0299bbe6f0112ead7ff38b7985321e904c&iPlanetDirectoryPro=
AQIC5wM2LY4SfczvfJJpn1IfT3pStks2VjzPMebgYVAxtyE%3D%40AAJTSQACMDE%3D%23
Content-Type: text/html; charset=iso-8859-1
Content-Length: 0
Date: Fri, 01 Feb 2008 23:48:30 GMT
------------------------------------------------------------
http://HostName.example.com:8080/opensso/SSORedirect/metaAlias/idp?ReqID=
s27926cc0299bbe6f0112ead7ff38b7985321e904c&iPlanetDirectoryPro=
AQIC5wM2LY4SfczvfJJpn1IfT3pStks2VjzPMebgYVAxtyE%3D%40AAJTSQACMDE%3D%23

GET /opensso/SSORedirect/metaAlias/idp?ReqID=s27926cc0299bbe6f0112ead7ff38b79
85321e904c&iPlanetDirectoryPro=AQIC5wM2LY4SfczvfJJpn1IfT3pStks2VjzPMebgYVAxtyE%
3D%40AAJTSQACMDE%3D%23 HTTP/1.1
Host: HostName.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: JSESSIONID=766be1d1028d55badd1ed0fe34ac; SMSESSION=jlO0TgMQfglpU+
GHQCJqbnoE2Pevax6fdzPGU7ZAgJuPb/fxTjCbWX1B1RO6QaLJn6VoVGNK8Sy6IeILAyv+Lci
```

```
S/OMK1E0tSXnL5Uvit3XIuWuiSMuklyDMIlOQ6n3ZSGGr9sKBUch5YVfGcfGjHQFcBIlzegQx
BRrgH/l2rc8aTEHdCrprvBiRHwQlxJbrcWMqfJw7h+HUEtiz9bQCUkwMbpEW4eBfNyRlZTGov
3K5hg4HK4tuoyvOeKdZaewlTB4Lm+QeGWo2qv2mPDP+eVtBiVtRVHHTHGfSthTJYQOOc4rPV2
dnl8axpWppGByeUmfmeService Provider9x5hVxDi91iyobTybKpDz0bltkvnHbqwbLfehUP
tJFxS3Z54y9dmiuoQ+B5Kdrs7DNuvrnAI1ZQdDKQEVA4Pt+vA9KO18ah9V1I7BZ9D/x60uWxfa
A3Ty8lRgWhMYqdBulFMD1B29sxboNHWdJ2FaxQJGjMpSEZ5iHB50ovF4YFXRyPP5Tl7eJxIebL
KX02LFrG/osNZ9UKHrMY1MRK5WWHJlYB040ADVcTNrFkc39vcYIA1eGDYhC/NaOd412HP5S0UX
0/59ADMLBsX/qBjcdODy3li+4eZnK1oHw/9yr3LCjewJ+H9w0k0/dQw99vgwEM2RPFgH5Y7W6
k6h1efp67VKXLBiJ1OZPJe2SCEDAOUla8qsC8fQ0VWTy/TfVhVtqJOaSLZrACX7uhPzbZE1EAP
d8x7UeJquFll3WpdnZYObd0DQLeoWZcF2rPIcfBn+8X8oig5KzvAgQ9R8MR+h7OkYfhmwwBDaQ
kbKPpIxjpeLNxKpkEVWJ9HoHOpZ/txCQUAHqPV41YjZ6CQfBfUqdOHbfje9O+0pJ1aHMntI4VY
ZOqdxsA+n9cgKjNQ8ruHOqSKhAQfEgipwcM2fMU3Uqmtr+0/+5bi7Cbs=; amlbcookie=01;
iPlanetDirectoryPro=AQIC5wM2LY4SfczvfJJpn1IfT3pStks2VjzPMebgYVAxtyE=@AAJTS
QACMDE=#

HTTP/1.x 302 Moved Temporarily
X-Powered-By: JService Provider/2.1
Server: Sun Java System Application Server 9.1
Location: http://ide-13.red.example.com:8080/opensso/Consumer/metaAlias/
sp?SAMLart=AAQAAE6JQxQxFQ72nsd5qDmVUTW5T3ieNSAqIADayEcXVxKAZQSjzCxJMDE%3D
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 0
Date: Fri, 01 Feb 2008 23:48:30 GMT
------------------------------------------------------------
```

5. Finish SAML SSO

```
http://ide-13.red.example.com:8080/opensso/Consumer/metaAlias/sp?SAMLart=
AAQAAE6JQxQxFQ72nsd5qDmVUTW5T3ieNSAqIADayEcXVxKAZQSjzCxJMDE%3D

GET /opensso/Consumer/metaAlias/sp?SAMLart=AAQAAE6JQxQxFQ72nsd5qDmVUTW5T
3ieNSAqIADayEcXVxKAZQSjzCxJMDE%3D HTTP/1.1
Host: ide-13.red.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: JSESSIONID=765d4c266461607b4b55811d34ca; SMSESSION=jlO0TgMQfglpU+
GHQCJqbnoE2Pevax6fdzPGU7ZAgJuPb/fxTjCbWX1B1RO6QaLJn6VoVGNK8Sy6IeILAyv+Lci
S/OMK1E0tSXnL5Uvit3XIuWuiSMuklyDMIlOQ6n3ZSGGr9sKBUch5YVfGcfGjHQFcBIlzegQx
BRrgH/l2rc8aTEHdCrprvBiRHwQlxJbrcWMqfJw7h+HUEtiz9bQCUkwMbpEW4eBfNyRlZTGov
3K5hg4HK4tuoyvOeKdZaewlTB4Lm+QeGWo2qv2mPDP+eVtBiVtRVHHTHGfSthTJYQOOc4rPV2
```

```
dnl8axpWppGByeUmfmeService Provider9x5hVxDi91iyobTybKpDz0bltkvnHbqwbLfehU
PtJFxS3Z54y9dmiuoQ+B5Kdrs7DNuvrnAI1ZQdDKQEVA4Pt+vA9KO18ah9V1I7BZ9D/x60uWx
faA3Ty8lRgWhMYqdBulFMD1B29sxboNHWdJ2FaxQJGjMpSEZ5iHB50ovF4YFXRyPP5Tl7eJxI
ebLKX02LFrG/osNZ9UKHrMY1MRK5WWHJlYB040ADVcTNrFkc39vcYIA1eGDYhC/NaOd412HP5
S0UX0/59ADMLBsX/qBjcdODy3li+4eZnK1oHw/9yr3LCjewJ+H9w0k0/dQw99vgwEM2RPFgH5Y
7W6k6h1efp67VKXLBiJ1OZPJe2SCEDAOUla8qsC8fQ0VWTy/TfVhVtqJOaSLZrACX7uhPzbZE1
EAPd8x7UeJquFll3WpdnZYObd0DQLeoWZcF2rPIcfBn+8X8oig5KzvAgQ9R8MR+h7OkYfhmwwB
DaQkbKPpIxjpeLNxKpkEVWJ9HoHOpZ/txCQUAHqPV41YjZ6CQfBfUqdOHbfje9O+0pJ1aHMntI
4VYZOqdxsA+n9cgKjNQ8ruHOqSKhAQfEgipwcM2fMU3Uqmtr+0/+5bi7Cbs=

HTTP/1.x 200 OK
X-Powered-By: JService Provider/2.1
Server: Sun Java System Application Server 9.1
Set-Cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfcxHYS5DBuFiEDZVArdPot5Wt07zTqKO6+w=
@AAJTSQACMDE=#; Domain=ide-13.red.example.com; Path=/
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
Date: Fri, 01 Feb 2008 23:47:30 GMT
```

Using OpenSSO Enterprise to Enable SiteMinder Federation in a Service Provider Environment

The following is a high-level overview of the sequence you must follow to enable SiteMinder with OpenSSO Enterprise in a Service Provider Environment:

1. Install OpenSSO Enterprise Instances.
2. Install and Configure SiteMinder in Service Provider Domain.
3. Configure OpenSSO Enterprise Identity Provider and Service Provider for SAML2 protocols.
4. Review Sample Single Sign-On Interactions.

▼ To Install OpenSSO Enterprise Instances

1 Install OpenSSO Enterprise in the Identity Provider Environment.

OpenSSO Enterprise is not the only supported access control software that can be used in the Identity Provider. But for optimum protocol interoperability, choosing OpenSSO Enterprise is a good practice. For detailed installation and configuration information, see the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.

Ideally, Service Provider and Identity Provider are deployed in two different domains. At minimum, the cookie domains should be different to ensure cookie validation consistency.

2 **Install OpenSSO Enterprise in the Service Provider Environment.**

The OpenSSO Enterprise in the Service Provider environment is the SAML2 protocols initiator. The SiteMinder Web Agent can protect the enterprise application, but will still redirect to OpenSSO Enterprise for single sign-on purposes.

▼ To Install and Configure SiteMinder in the Service Provider Domain

Before You Begin Before proceeding, be sure to read the general instructions in "Installing SiteMinder" on page 165 and in "Configuring SiteMinder After Installation" on page 166. The following steps provide additional installation information specific only to this use case.

1 **Install SiteMinder.**

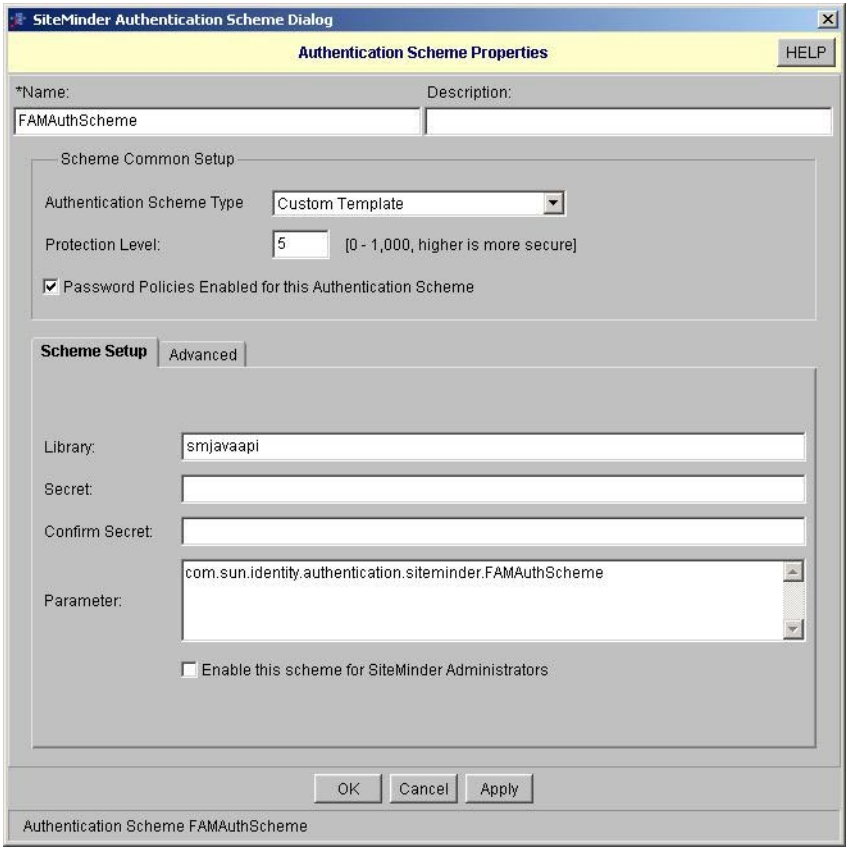
This is the domain that protects its enterprise applications using their SiteMinder agents. For the installation of SiteMinder and SiteMinder agents, see the CA SiteMinder product documentation.

2 **Create a custom authentication scheme.**

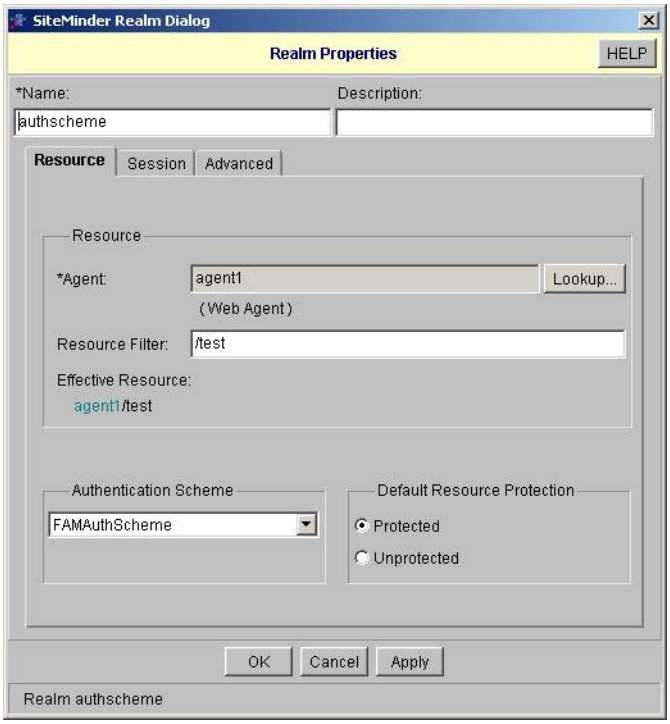
a. **Copy the compiled SiteMinder authentication scheme JAR files into the SiteMinder `lib` directory.**

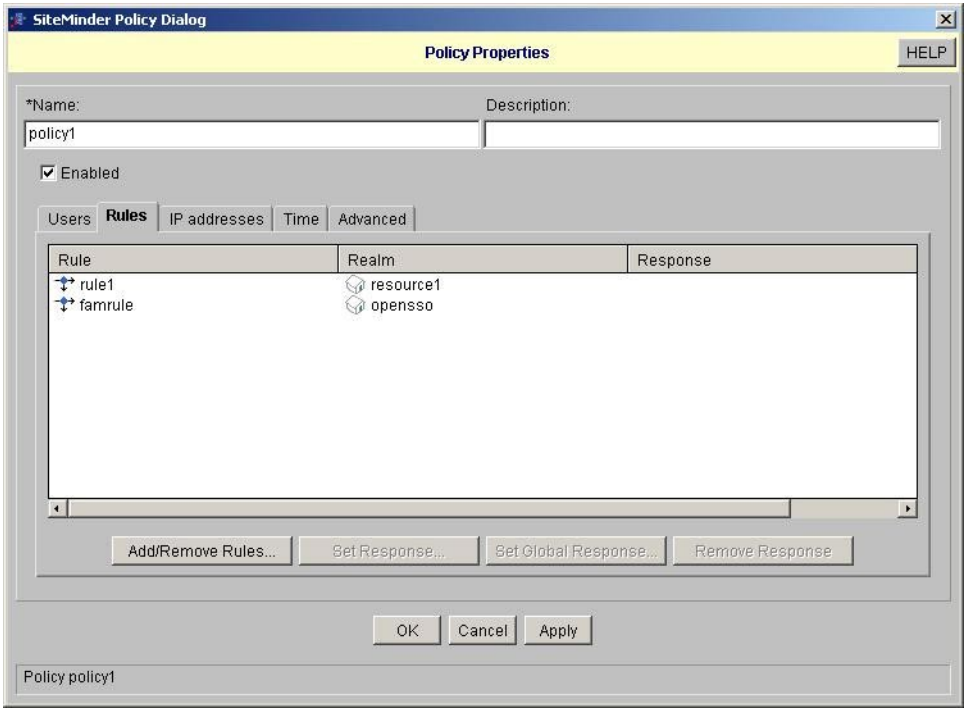
After you unzip the OpenSSO Enterprise binary, the SiteMinder custom authentication module is located under the directory *unzip-directory*`/integrations/siteminder/`. The `README.html` provides steps for building a custom authentication module. The OpenSSO Enterprise authentication module is a Java-based authentication scheme in SiteMinder. The `README.html` explains the steps for configuring the SiteMinder authentication scheme.

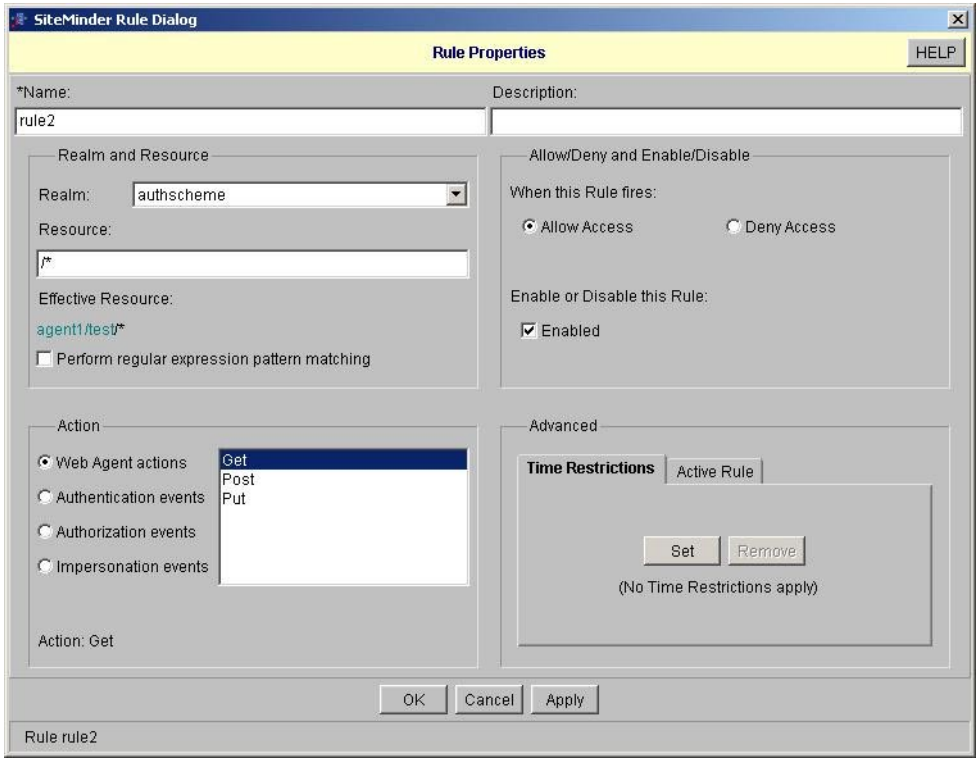
b. In the SiteMinder console, click Authentication Scheme, and then click "Create Custom Authentication Scheme."



3 Configure a resource and a policy to trigger the OpenSSO Enterprise authentication module.




SiteMinder Policy Dialog
Policy Properties
HELP
*Name:
Description:
policy1
Enabled
Users
Rules
IP addresses
Time
Advanced
Rule
Realm
Response
rule1
resource1
famrule
opensso
Add/Remove Rules...
Set Response...
Set Global Response...
Remove Response
OK
Cancel
Apply
Policy policy1



▼ To Configure the OpenSSO Enterprise Identity Provider and Service Provider for SAML2 protocols

For these configurations, you must have the following:

- Identity Provider metadata
- Identity Provider extended metadata
- Service Provider metadata
- Service Provider extended metadata

In Identity Provider, import Identity Provider metadata and Identity Provider extended metadata as hosted metadata. Import Service Provider metadata and Service Provider extended metadata as remote entity metadata.

Before You Begin Before loading metadata, read through the following steps for the changes that you must make to the metadata. See the SAML2 samples for detailed instructions on how to setup SAML2, See the OpenSSO website for commands and syntax.

1. **Edit the extended metadata XML element `<EntityConfig>`.**

   Change the `hosted` attribute from `true` to `false`.

2. **Generate the metadata templates in both Identity Provider and Service Provider environments.**

   You can generate the metadata templates in one of two ways:

   - Use the browser-based URL `http://`*host:port*`/opensso/famadm.jsp`
   - Use the `famadm` command.

     At the Identity Provider, where `idp_meta_alias` is `/idp`:

     ```
     famadm create-metadata-templ -y idp_entity_id -u amadmin
     -f admin_password_file_name -m idp_standard_metadata -x idp_extended_metadata
     -i idp_meta_alias
     ```

     At the Service Provider, where `sp_meta_alias` is `/sp`:

     ```
     famadm create-metadata-templ -y sp_entity_id -u amadmin
     -f admin_password_file_name -msp_standard_metadata
     -x sp_extended_metadata -s sp_meta_alias
     ```

3. **Customize the extended metadata at the Service Provider.**

   Add the Service Provider extended metadata as an attribute. This attribute is used by the SAML protocols to do any post-SSO Authentication process. In this example, the attribute is named `spAdapter`. In the architecture diagram, this is the SiteMinder Plug-In. The SiteMinder Plug-In uses the OpenSSO Enterprise session to authenticate against SiteMinder and to establish the SiteMinder session. The Service Provider metadata must have the following attributes:

   ```
   <Attribute name="spAdapter">
               <Value>com.sun.identity.saml2.plugins.SMAdapter</Value>
           </Attribute>
           <Attribute name="spAdapterEnv">
               <Value>AgentIP=192.18.120.65</Value>
               <Value>AgentID=agent1</Value>
               <Value>PolicyServerIPAddress=192.18.120.65</Value>
               <Value>AuthorizationPort=44443</Value>
               <Value>AuthenticationPort=44442</Value>
               <Value>AccountingPort=44441</Value>
               <Value>AgentHostName=HostName.example.com</Value>
               <Value>ConnectionMinimum=2</Value>
               <Value>ConnectionMaximum=20</Value>
               <Value>ConnectionStep=2</Value>
               <Value>RequestTimeout=60</Value>
               <Value>FAMCookieName=iPlanetDirectoryPro</Value>
               <Value>SMCookieName=SMSESSION</Value>
               <Value>CookieDomain=.red.example.com</Value>
               <Value>Resource=/test/index.html</Value>
               <Value>SharedSecret={RC2}1r976MPOVq5JPpKzxFsXxIlut/YkgtUeklaceAoONCN
   ```

```
                mFJKDY+W8CkVpGY0to+x6apsIQAMPWLsgrm6NcdvyXv7K9Vf0vEALeWOy5BqLAhw
                fgKp4TbFRQspgv4w24ZOWsk57rwJ0N4kUJdM9lsLRu5hGKXArRJNpF80vS/U53TZ
                vM/qE5I3DcCOWKY4lJBZh</Value>
        </Attribute>
```

4 **Set the Service Provider extended metadata attribute `transientUser` to your anonymous user.**

```
 <Attribute name="transientUser">
             <Value>anonymous</Value>
        </Attribute>
```

Also verify that the OpenSSO Enterprise Service Provider is enabled for Anonymous authentication. See the OpenSSO Enterprise product documentation for more information.

5 **Add the Circle of Trust through the OpenSSO Enterprise administration console.**

Before loading, verify that the `hosted` attribute in the extended metadata has been changed to `false`.

6 **Load the hosted metadata in both the Identity Provider and the Service Provider.**

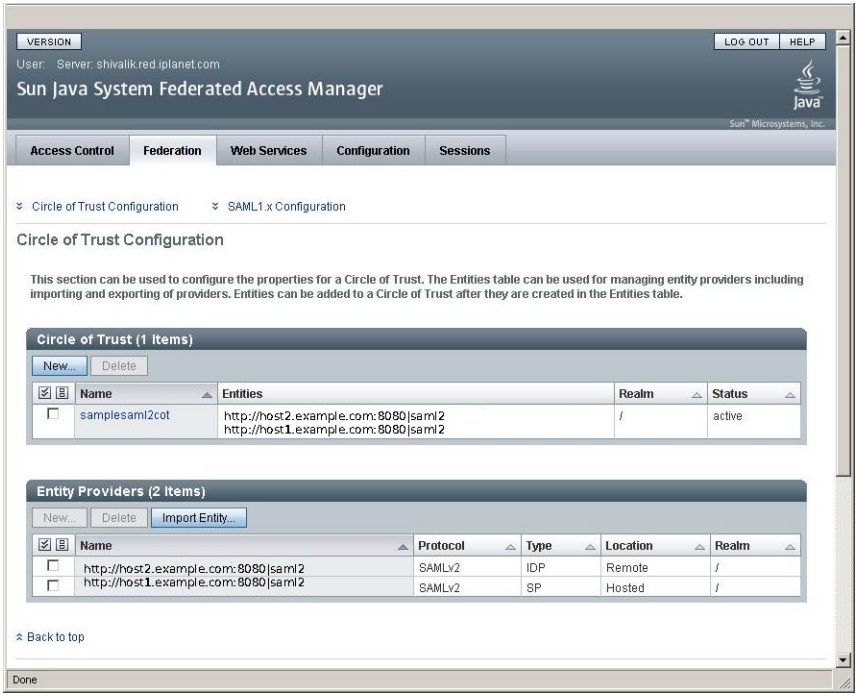
You can use the `famadm` command or the OpenSSO Enterprise administration console.

7 **Exchange the Service Provider metadata with the Identity Provider.**

8 **Exchange the Identity Provider metadata with the Service Provider metadata.**

9 **Load the metadata.**

10 **After successful metadata exchange, verify through OpenSSO Enterprise administration console that metadata is properly configured.**



11 Verify that Single Sign-On works properly.

Access the enterprise application protected by SiteMinder Service Provider Agent. This should redirect to the OpenSSO Enterprise for authentication where the SAML2 SSO is initiated.

Sample Service Provider Interactions

This section provides sample output from the following interactions:

1. “1. Invocation of SAML SSO request” on page 200
2. “2. Redirection to Identity Provider” on page 201
3. “3. Redirection to Login” on page 202
4. “4. Redirection to Service Provider Assertion Consumer Service ” on page 205
5. “5. Check the SMSESSION Creation” on page 205

1. Invocation of SAML SSO request

```
http://HostName.example.com:8080/opensso/saml2/jsp/
spSSOInit.jsp?metaAlias=/sp&idpEntityID=
http://ide-13.red.example.com:8080/opensso&NameIDFormat=transient

GET /opensso/saml2/jsp/spSSOInit.jsp?metaAlias=/sp&idpEntityID=
```

```
http://ide-13.red.example.com:8080/opensso&NameIDFormat=transient HTTP/1.1
Host: HostName.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: JSESSIONID=5fa8300161a1d5dc746ad8f9fb31
```

2. Redirection to Identity Provider

```
HTTP/1.x 302 Moved Temporarily
X-Powered-By: JService Provider/2.1
Server: Sun Java System Application Server 9.1
Location: http://ide-13.red.example.com:8080/opensso/SSORedirect/
metaAlias/idp?SAMLRequest=nVRfb9owEH%2Fvp4j8DnESaMECJAaahtRtDOge9
maSy7Dm2JnvQtm3rx0oYmqFVF7P57vfP3uEstK1mDa0Myv42wBSFB0qbVC0J2PWOCO
sRIXCyApQUC7W06%2BPIu1yUTtLNrea3S3mY4bpYJvxLNsW5Rb60O89DNNhWQ4l7yV
8cJ8kUAzT7J5nKYt%2BgkNlzZj5MSxaIDawMEjSkC9xPujwtMN7m2Qoej3Rf%2FjFo
rmHpoyk9taOqBZxrAroJFnXQdFVtZYGqJvbSgz4gMe2BoNo4%2FX6%2BwoK5SCnuAK
SU60k%2Bps1iz5bl0NLfcxKqRECkqVEVHs4V5Ynjp%2BUKZT5fV2Q7bEJxZfNZtmZO
lKlzIlFU0RwAfrMGmwqcGtwe5XD0%2BrxTAZ3ai%2B1%2BnOdzuuACy5Ys8ndKNglWi
HdhYHX4cpXWGzyERCj%2BGLZaXUtvvnxi%2FnSapX%2FuyVDwY5K0vXuUFFFp2xbBTl
pUIHxCq%2BXYf%2BPxoMvFbiPqeoN0to%2BzxxI8s6Ta4BNjiz%2F53Ume3orULTx8a
YQHOgW0jNb1dIpDKmGQwjL5Gjl5eCZ9k6toLzF2KttucjDaF8OsX%2B2rghx908Fik3
QtraOTm6%2Fh%2Bes0btyeLHit9%2FL5AU%3D
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 0
Date: Mon, 04 Feb 2008 19:44:57 GMT
------------------------------------------------------------
http://ide-13.red.example.com:8080/opensso/SSORedirect/metaAlias/
idp?SAMLRequest=nVRfb9owEH%2Fvp4j8DnESaMECJAaahtRtDOge9maSy7Dm2Jn
vQtm3rx0oYmqFVF7P57vfP3uEstK1mDa0Myv42wBSFB0qbVC0J2PWOCOsRIXCyApQ
UC7W06%2BPIu1yUTtLNrea3S3mY4bpYJvxLNsW5Rb60O89DNNhWQ4l7yV8cJ8kUAz
T7J5nKYt%2BgkNlzZj5MSxaIDawMEjSkC9xPujwtMN7m2Qoej3Rf%2FjFormHpoyk
9taOqBZxrAroJFnXQdFVtZYGqJvbSgz4gMe2BoNo4%2FX6%2BwoK5SCnuAKSU60k%
2Bps1iz5bl0NLfcxKqRECkqVEVHs4V5Ynjp%2BUKZT5fV2Q7bEJxZfNZtmZOlKlzI
lFU0RwAfrMGmwqcGtwe5XD0%2BrxTAZ3ai%2B1%2BnOdzuuACy5Ys8ndKNglWiHdh
YHX4cpXWGzyERCj%2BGLZaXUtvvnxi%2FnSapX%2FuyVDwY5K0vXuUFFFp2xbBTlp
UIHxCq%2BXYf%2BPxoMvFbiPqeoN0to%2BzxxI8s6Ta4BNjiz%2F53Ume3orULTx8
aYQHOgW0jNb1dIpDKmGQwjL5Gjl5eCZ9k6toLzF2KttucjDaF8OsX%2B2rghx908F
ik3QtraOTm6%2Fh%2Bes0btyeLHit9%2FL5AU%3D
```

```
GET /opensso/SSORedirect/metaAlias/idp?SAMLRequest=nVRfb9owEH%2Fvp
4j8DnESaMECJAaahtRtDOge9maSy7Dm2JnvQtm3rx0oYmqFVF7P57vfP3uEstK1mDa
0Myv42wBSFB0qbVC0J2PWOCOsRIXCyApQUC7W06%2BPIu1yUTtLNrea3S3mY4bpYJv
xLNsW5Rb60O89DNNhWQ4l7yV8cJ8kUAzT7J5nKYt%2BgkNlzZj5MSxaIDawMEjSkC9
xPujwtMN7m2Qoej3Rf%2FjFormHpoyk9taOqBZxrAroJFnXQdFVtZYGqJvbSgz4gMe
2BoNo4%2FX6%2BwoK5SCnuAKSU60k%2Bps1iz5bl0NLfcxKqRECkqVEVHs4V5Ynjp%
2BUKZT5fV2Q7bEJxZfNZtmZOlKlzIlFU0RwAfrMGmwqcGtwe5XD0%2BrxTAZ3ai%2B
1%2BnOdzuuACy5Ys8ndKNglWiHdhYHX4cpXWGzyERCj%2BGLZaXUtvvnxi%2FnSapX%
2FuyVDwY5K0vXuUFFFp2xbBTlpUIHxCq%2BXYf%2BPxoMvFbiPqeoN0to%2BzxxI8s6
Ta4BNjiz%2F53Ume3orULTx8aYQHOgW0jNb1dIpDKmGQwjL5Gjl5eCZ9k6toLzF2Ktt
ucjDaF8OsX%2B2rghx908Fik3QtraOTm6%2Fh%2Bes0btyeLHit9%2FL5AU%3D HTTP/1.1
Host: ide-13.red.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

3. Redirection to Login

```
HTTP/1.x 302 Moved Temporarily
X-Powered-By: JService Provider/2.1
Server: Sun Java System Application Server 9.1
Set-Cookie: JSESSIONID=5f9f32d1896460b979b16ac14fb3; Path=/opensso
Location: http://ide-13.red.example.com:8080/opensso/UI/Login?realm=
/&goto=http%3A%2F%2Fide-13.red.example.com%3A8080%2Fopensso%
2FSSORedirect%2FmetaAlias%2Fidp%3FReqID%3Ds28b3033bdfbe5e547929ff9a
04108611ed9236032
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 0
Date: Mon, 04 Feb 2008 19:43:58 GMT
------------------------------------------------------------
http://ide-13.red.example.com:8080/opensso/UI/Login?realm=/&goto=
http%3A%2F%2Fide-13.red.example.com%3A8080%2Fopensso%2FSSORedirect%
2FmetaAlias%2Fidp%3FReqID%3Ds28b3033bdfbe5e547929ff9a04108611ed9236032

GET /opensso/UI/Login?realm=/&goto=http%3A%2F%2Fide-13.red.example.com%
3A8080%2Fopensso%2FSSORedirect%2FmetaAlias%2Fidp%3FReqID%3Ds28b3033bdfb
e5e547929ff9a04108611ed9236032 HTTP/1.1
Host: ide-13.red.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
```

```
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: JSESSIONID=5f9f32d1896460b979b16ac14fb3

HTTP/1.x 200 OK
X-Powered-By: JService Provider/2.1
Server: Sun Java System Application Server 9.1
Cache-Control: private
Pragma: no-cache
Expires: 0
X-DSAMEVersion: 8.0 (2007-November-29 01:17)
AM_CLIENT_TYPE: genericHTML
Set-Cookie: AMAuthCookie=AQIC5wM2LY4SfczOj691d2eiNkQCzmceO14vekWbCSzRU/
E=@AAJTSQACMDE=#; Domain=ide-13.red.example.com; Path=/
Set-Cookie: amlbcookie=01; Domain=ide-13.red.example.com; Path=/
Content-Type: text/html;charset=UTF-8
Transfer-Encoding: chunked
Date: Mon, 04 Feb 2008 19:43:58 GMT
------------------------------------------------------------


http://ide-13.red.example.com:8080/opensso/UI/Login?AMAuthCookie=
AQIC5wM2LY4SfczOj691d2eiNkQCzmceO14vekWbCSzRU%2FE%3D%40AAJTSQACMDE%3D%23

POST /opensso/UI/Login?AMAuthCookie=AQIC5wM2LY4SfczOj691d2eiNkQCzmceO14v
ekWbCSzRU%2FE%3D%40AAJTSQACMDE%3D%23 HTTP/1.1
Host: ide-13.red.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://ide-13.red.example.com:8080/opensso/UI/Login?realm=/
&goto=http%3A%2F%2Fide-13.red.example.com%3A8080%2Fopensso%2FSSORedirect%
2FmetaAlias%2Fidp%3FReqID%3Ds28b3033bdfbe5e547929ff9a04108611ed9236032
Cookie: JSESSIONID=5f9f32d1896460b979b16ac14fb3; AMAuthCookie=AQIC5wM2LY4
SfczOj691d2eiNkQCzmceO14vekWbCSzRU/E=@AAJTSQACMDE=#; amlbcookie=01
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: 254
IDToken0=&IDToken1=amadmin&IDToken2=admin123&IDButton=Log+In&goto=
aHR0cDovL2lkZS0xMy5yZWQuaXBsYW5ldC5jb206ODA4MC9vcGVuc3NvL1NTT1JlZGlyZWN0L2
1ldGFBbGlhcy9pZHA%2FUmVxSUQ9czI4YjMwMzNiZGZiZTVlNTQ3OTI5ZmY5YTA0MTA4NjExZW
Q5MjM2MDMy&encoded=true&gx_charset=UTF-8
HTTP/1.x 302 Moved Temporarily
X-Powered-By: Servlet/2.5
Server: Sun Java System Application Server 9.1
Cache-Control: private
Pragma: no-cache
Expires: 0
X-DSAMEVersion: 8.0 (2007-November-29 01:17)
AM_CLIENT_TYPE: genericHTML
X-AuthErrorCode: 0
Set-Cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfczOj691d2eiNkQCzmceO14vekWbCSzRU/
E=@AAJTSQACMDE=#; Domain=ide-13.red.example.com; Path=/
Set-Cookie: AMAuthCookie=LOGOUT; Domain=ide-13.red.example.com;
Expires=Thu, 01-Jan-1970 00:00:10 GMT; Path=/
Location: http://ide-13.red.example.com:8080/opensso/SSORedirect/
metaAlias/idp?ReqID=s28b3033bdfbe5e547929ff9a04108611ed9236032
Content-Type: text/html; charset=iso-8859-1
Content-Length: 0
Date: Mon, 04 Feb 2008 19:44:05 GMT
------------------------------------------------------------
http://ide-13.red.example.com:8080/opensso/SSORedirect/metaAlias/
idp?ReqID=s28b3033bdfbe5e547929ff9a04108611ed9236032

GET /opensso/SSORedirect/metaAlias/idp?ReqID=s28b3033bdfbe5e54792
9ff9a04108611ed9236032 HTTP/1.1
Host: ide-13.red.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
 Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://ide-13.red.example.com:8080/opensso/UI/Login?realm=
/&goto=http%3A%2F%2Fide-13.red.example.com%3A8080%2Fopensso%2FSSORedirect%
2FmetaAlias%2Fidp%3FReqID%3Ds28b3033bdfbe5e547929ff9a04108611ed9236032
Cookie: JSESSIONID=5f9f32d1896460b979b16ac14fb3; amlbcookie=01;
iPlanetDirectoryPro=AQIC5wM2LY4SfczOj691d2eiNkQCzmceO14vekWbCSzRU/E=@AAJTSQACMDE=#

HTTP/1.x 302 Moved Temporarily
X-Powered-By: JService Provider/2.1
```

```
Server: Sun Java System Application Server 9.1
Location: http://HostName.example.com:8080/opensso/Consumer/metaAlias/
sp?SAMLart=AAQAAI4sWYpfoDDYJrHzsMnG%2BjyNM94p5ejn49a%2BnZ0s3ylY7knQ6tkLMDE%3D
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 0
Date: Mon, 04 Feb 2008 19:44:05 GMT
```

4. Redirection to Service Provider Assertion Consumer Service

```
http://HostName.example.com:8080/opensso/Consumer/metaAlias/sp?SAMLart=
AAQAAI4sWYpfoDDYJrHzsMnG%2BjyNM94p5ejn49a%2BnZ0s3ylY7knQ6tkLMDE%3D

GET /opensso/Consumer/metaAlias/sp?SAMLart=AAQAAI4sWYpfoDDYJrHzsMnG%
2BjyNM94p5ejn49a%2BnZ0s3ylY7knQ6tkLMDE%3D HTTP/1.1
Host: HostName.example.com:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.11)
Gecko/20071127 Firefox/2.0.0.11
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=
0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://ide-13.red.example.com:8080/opensso/UI/Login?realm=
/&goto=http%3A%2F%2Fide-13.red.example.com%3A8080%2Fopensso%2FSSORedirect%
2FmetaAlias%2Fidp%3FReqID%3Ds28b3033bdfbe5e547929ff9a04108611ed9236032
Cookie: JSESSIONID=5fa8300161a1d5dc746ad8f9fb31
```

5. Check the SMSESSION Creation

```
HTTP/1.x 200 OK
X-Powered-By: JService Provider/2.1
Server: Sun Java System Application Server 9.1
Set-Cookie: iPlanetDirectoryPro=AQIC5wM2LY4SfcwFsRqmpq0e6m+iL+tjmqYhTDsKeABb4Eg=
@AAJTSQACMDE=#; Domain=HostName.example.com;
Path=/Set-Cookie: SMSESSION=jnNJdOyhPMa6A7FKeD0tCgHyq3yt8Tsvtmj6G4NjbP05ftAMggw+
hqolfo32FJ8iOnggFoZ19qXVAJyqf0DvMqhM+X0oUVw3P3R83sBAT4uKtUaib7OxyTSi8W5pBI+hLexr
NczdpVWN9vCGDU97uBlJgpI8L9aeSNBgCSwo+gluvd1I72KGyFVgMLkIkfLMJhctpz+zKVt252yEf5Oh
QZlGhzT/DzNqBc+142eek5VwMzxABLhwuEQ1jI1VAGOYAeyQpSmikgNfWphDSV3X36L3+ZQqHZmzCwjb
8QKSrBZnMdGuZKCYc9U6N8VJ1Ft5zwi/lotOU198apSU2bI5nQzWnGjpz60Dxc6Ycy83bjOqby/ZYHrl
3OLv2wJ0RtEN8FPYFjbHLGg=; Domain=.red.example.com
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
Date: Mon, 04 Feb 2008 19:45:04 GMT
```

◆ ◆ ◆ CHAPTER 3

Integrating Oracle Access Manager

Oracle Access Manager (previously known as Oblix NetPoint and Oracle COREid) is an enterprise single sign-on product with many of the same features as Sun OpenSSO Enterprise and CA SiteMinder (previously known as Netegrity SiteMinder). Oracle Access Manager is usually used for both single sign-on and delegated administration. Many companies have an existing Oracle Access Manager deployed to protect both internal and external applications. This chapter describes options for integrating Oracle Access Manager with OpenSSO Enterprise. The chapter also provides instructions for configuring end-to-end Oracle Access Manager single sign-on using OpenSSO Enterprise.

The following topics are included in this chapter:

About Oracle Access Manager

Oracle has two solutions for web-based single sign-on. One solution is to use the legacy Oracle single sign-on product which is integrated in the Oracle Application Server. Another solution is to use the Oracle Access Manager product, previously known as Oblix Access, with Identity Server. The following major components comprise the Oracle Access System:

Oracle Identity Server	Provides user management and delegated administration functionality and workflows.
Oracle Policy Manager	Provides a web-based interface where administrators can create and manage access policies. The Policy Manager communicates with the

	directory server to write policy data, and communicates with the Access Server over the Oracle Access Protocol (OAP) to update the Access Server when certain policy modifications are made.
Oracle Access Server	Provides centralized authentication, authorization, and auditing to enable single sign-on and secure access control across enterprise resources.
Web Pass	An Oracle Access Manager web server plug-in (NSAPI filter). Web Pass passes information back and forth between a web server and the Identity Server. Depending upon its configuration, the Identity Server processes a request as either an XML or HTML file.
WebGate	A web server plug-in access client analogous to Sun OpenSSO Enterprise Policy Agent. WebGate intercepts HTTP requests for Web resources and forwards them to the Access Server for authentication and authorization.

Overview of a Typical Oracle Access Manager Session

The Access Server generates a session token with a URL that contains the `ObSSOCookie`. When the cookie is generated, part of the cookie is used as an encrypted session token. The encrypted session token contains the following:

- Distinguished name (DN) of the user
- Level of the authentication scheme
- IP address of the client to which the cookie was issued
- Time the cookie was originally issued
- Time the cookie was last updated

If the user has not been idle, the cookie is updated at a fixed interval to prevent the session from logout. The update interval is 1/4th of idle the session timeout parameter.

Unencrypted `ObSSOCookie` data includes the following:

- Cookie expiration time
- Domain in which the cookie is valid
- Optional flag that determines if the cookie can only be sent over SSL

The `ObSSOCookie` is a secure mechanism for user authentication. When the Access System generates the cookie, an MD-5 hash is taken of the session token. When `ObSSOCookie` is used to authenticate a user, the MD-5 hash is compared with the original cookie contents to be sure no one has tampered with the cookie. MD-5 is a one-way hash, so it cannot be unencrypted. The Access Server does the comparison by hashing the session token again and comparing the

output with the hash of the token already present in the Oracle Access Server cookie. If the two hashes do not match, the cookie is corrupt. The system relies on the fact that if someone tampers with the session token, the hashes will not match.

Understanding the Oracle Access Manager Use Cases

The following uses cases illustrate common Oracle Access Manager process flows:

- "Simple Single Sign-On Use Case" on page 209
- "Federated Single Sign-On Use Cases" on page 211

Single logout for any these of these use cases can be implemented in many ways. The logout for federation use cases must have a link in the partner portal for the following URL:

```
http:<sphost>:<spport>/opensso/saml2/jsp/spSingleLogoutInit.jsp?metaAlias=
<metaalias>&idpEntityID=<idp entityid>&RelayState=<integrated product logout url>
```

Single logout can also be achieved using Identity Provider-initiated single logout.

Simple Single Sign-On Use Case

Simple single sign-on integration is useful when an Oracle Access Manager instance is already deployed and configured to protect intranet enterprise applications. Additionally, OpenSSO Enterprise is deployed to protect the same intranet applications by honoring the user session obtained by Oracle Access Manager. In the following illustration, both OpenSSO Enterprise and Oracle Access Manager share the same user repository for user profile verification. OpenSSO Enterprise can also be configured to use the Ignore Profile option if it relies on the Oracle Access Manager session for attributes.

The following figure illustrates architecture in the simple single sign-on use case.

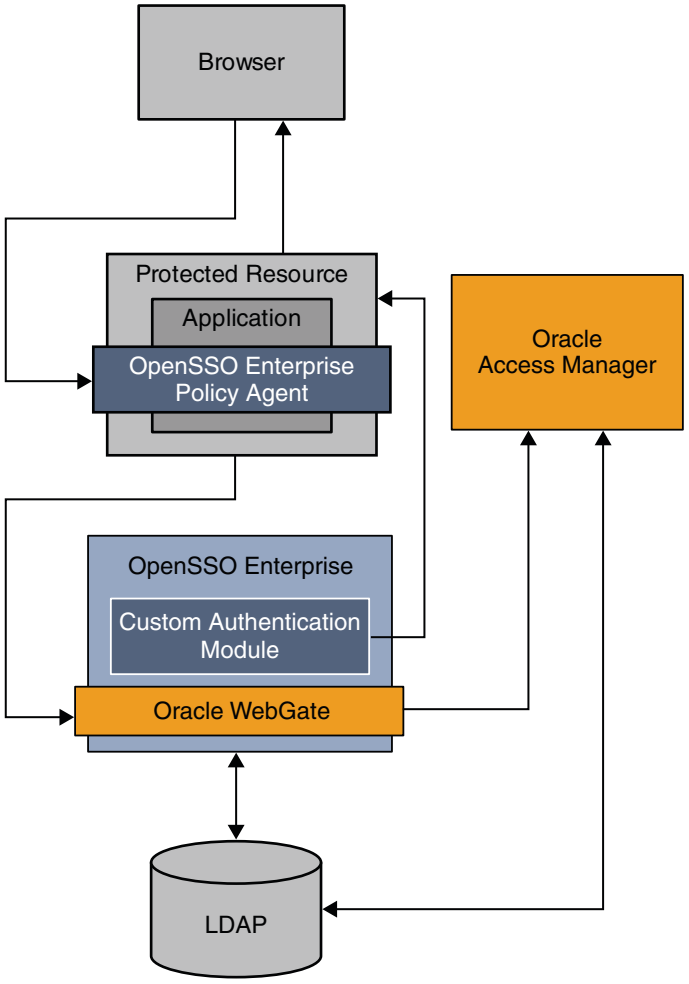


FIGURE 3–1 Simple Oracle Access Manager Single Sign-On

The following figure illustrates the process flow among components in the Identity Provider environment and Service Provider environment.

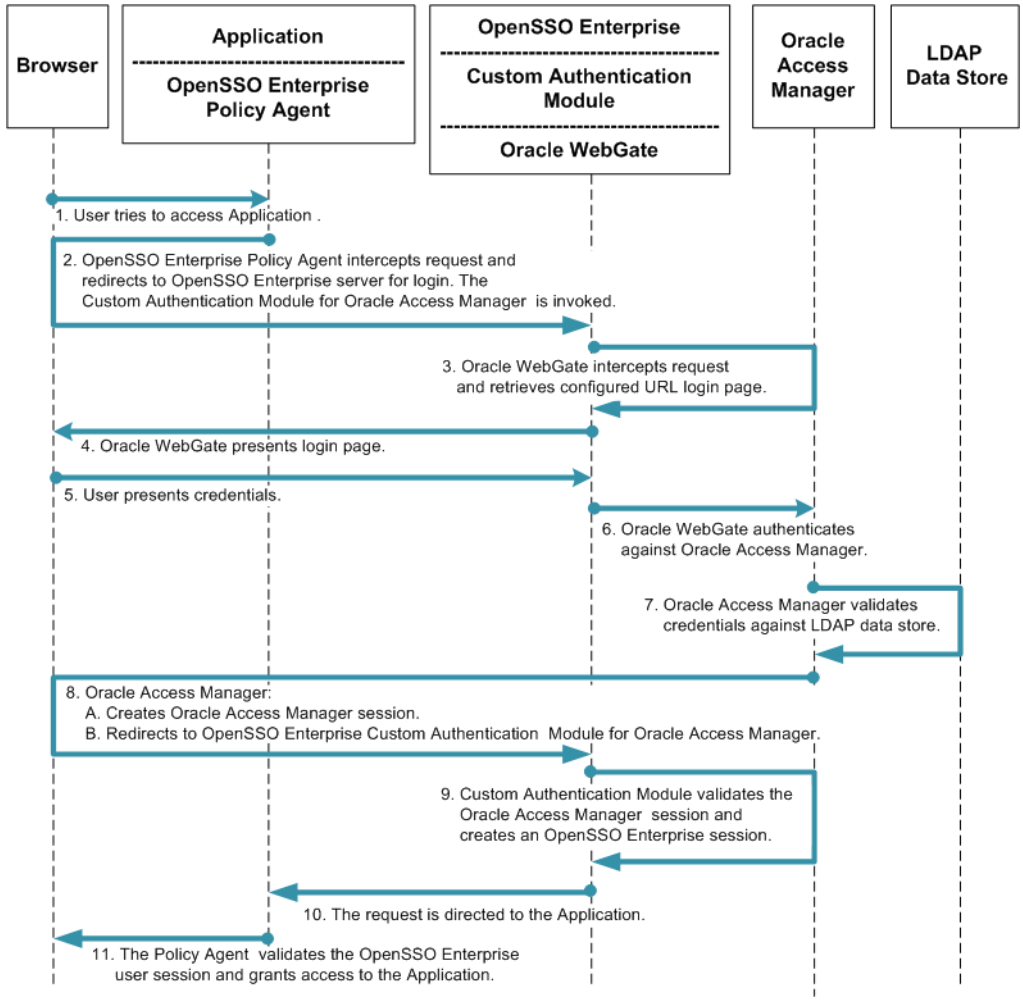


FIGURE 3–2 Process Flow for Simple Oracle Access Manager Single Sign-On

Federated Single Sign-On Use Cases

The SAML, ID-FF, and WS-Federation protocols provide cross-domain single sign-on among multiple trusted business entities. These protocols are also used in Identity Federation. Identity Federation involves an Identity Provider, also known as an authentication provider, and a Service Provider where the user authentication session at the Identity Provider is consumed. The following are common use cases in which Oracle Access Manager is enabled for federation protocols:

- Enabling Oracle Access Manager for federation protocols in a Service Provider environment

- Enabling Oracle Access Manager for federation protocols in an Identity Provider environment

The deployment examples in this chapter are built upon simple single sign-on integration. You must set up single sign-on before enabling federation. For more information about setting up simple single sign-on, see the *Deployment Example: Single Sign-On*. After setting up simple single sign-on, you can enable Oracle Access Manager for Federation in either the Identity Provider environment or in the Service Provider environment.

In the following examples, both Identity Provider and Service Provider are configured for transient federation. In most use cases, bulk federation is configured between the Identity Provider and Service Provider.

In transient federation, users exist only in the Identity Provider environment. The Service Provider honors the user authentication at the Identity Provider, and then creates an anonymous session. The anonymous session enables the Service Provider applications, protected by single sign-on, to be accessed. During SAML interactions, there is a possibility of exchanging user attribute information back to the Service Provider for authorization and other purposes. But that scenario is beyond the scope of this document.

Using OpenSSO Enterprise to Enable Oracle Federation in an Identity Provider Environment

In this example, Oracle Access Manager is the authentication provider in an Identity Provider environment and protects some of the intranet applications. OpenSSO Enterprise in this deployment resolves the single sign-on issues among enterprise applications in partner environments while Oracle Access Manager provides authentication.

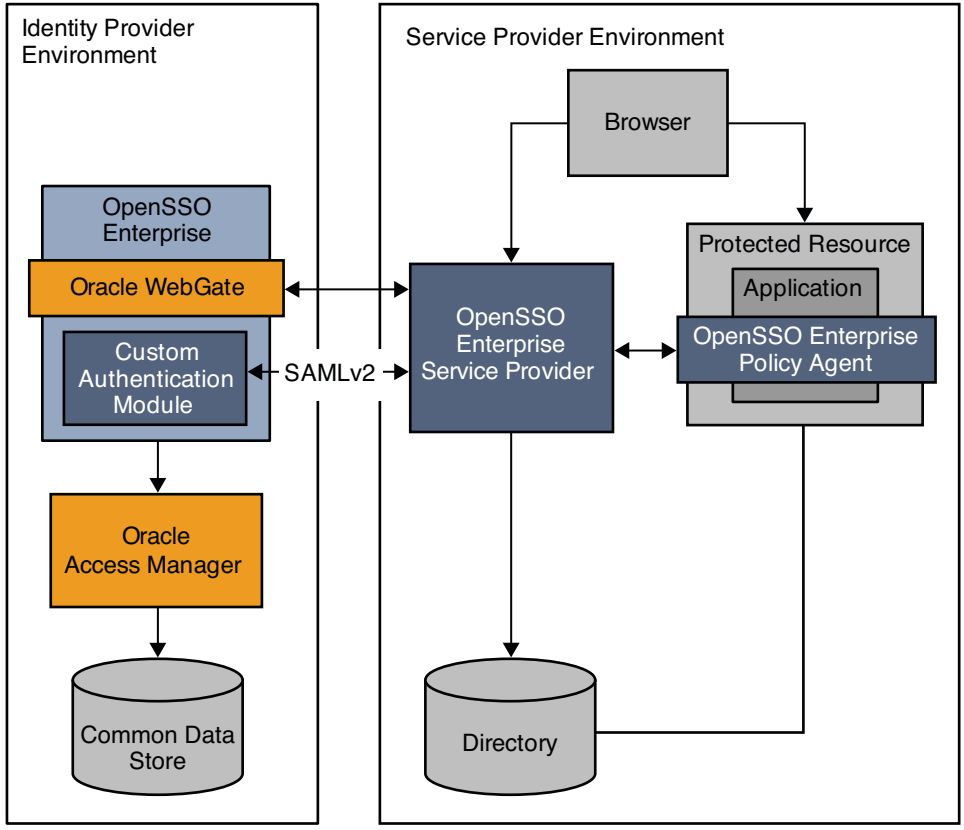


FIGURE 3–3 Oracle Access Manager Federation in an Identity Provider Environment

The following two figures illustrate the process flow among components in the Identity Provider environment and Service Provider environment.

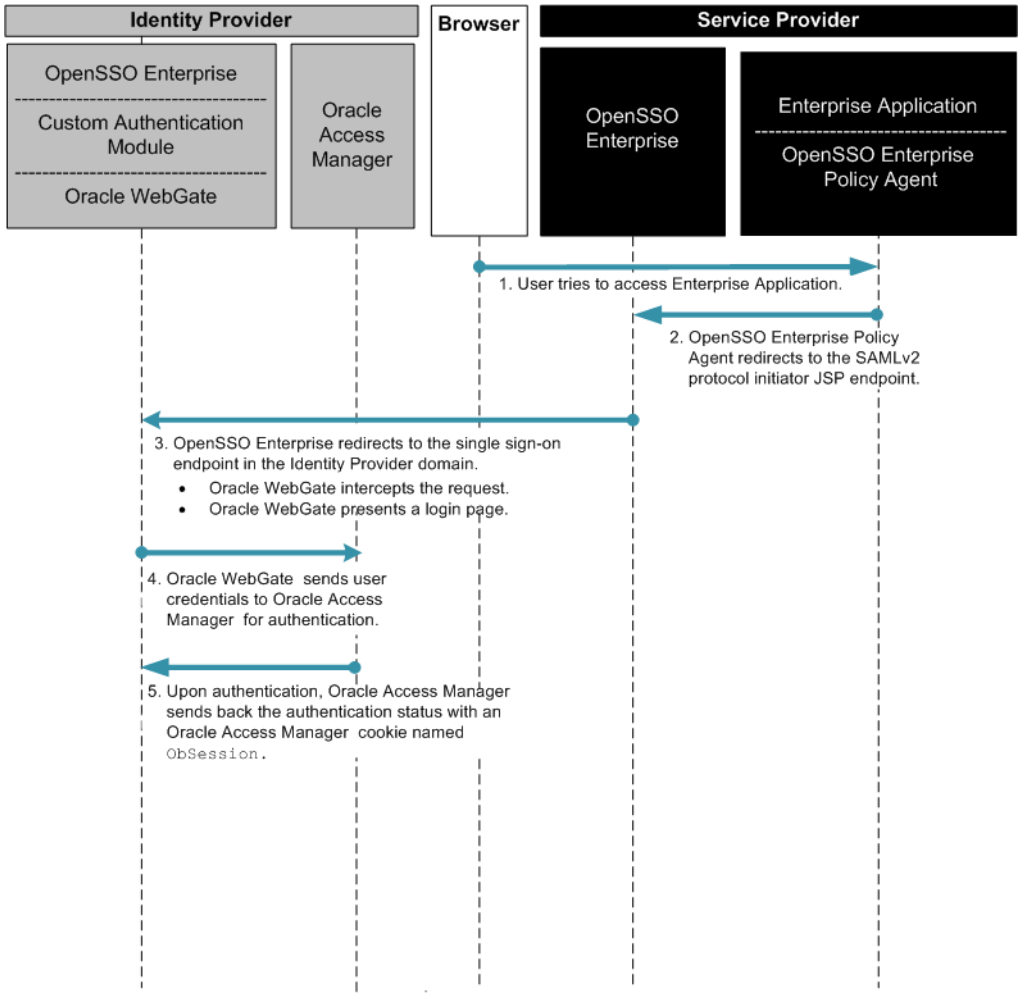


FIGURE 3–4 Process flow for Oracle Access Manager Federation in an Identity Provider Environment

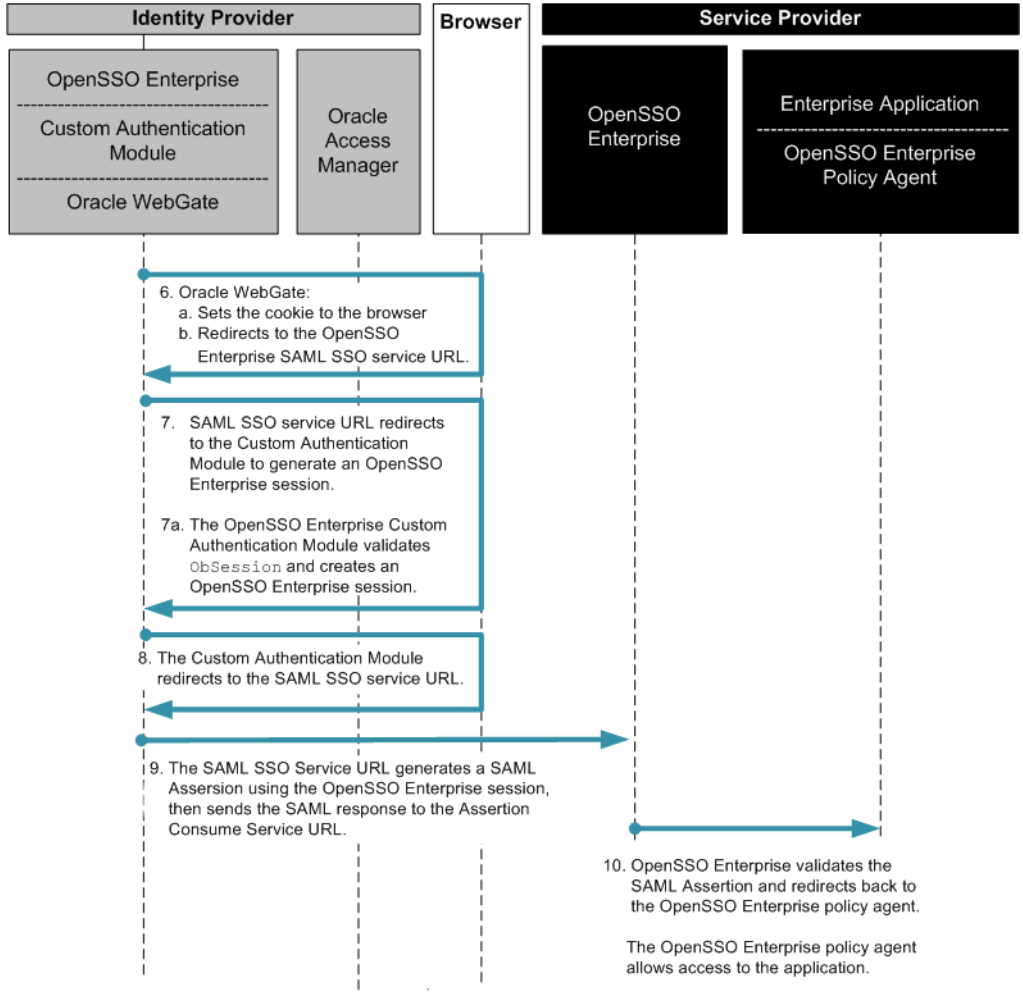


FIGURE 3–5 Process flow for Oracle Access Manager Federation in an Identity Provider Environment (continued)

Using OpenSSO Enterprise to Enable Oracle Federation in a Service Provider Environment

In this deployment, Oracle Access Manager is installed and configured in Service Provider Environment to protect legacy applications.

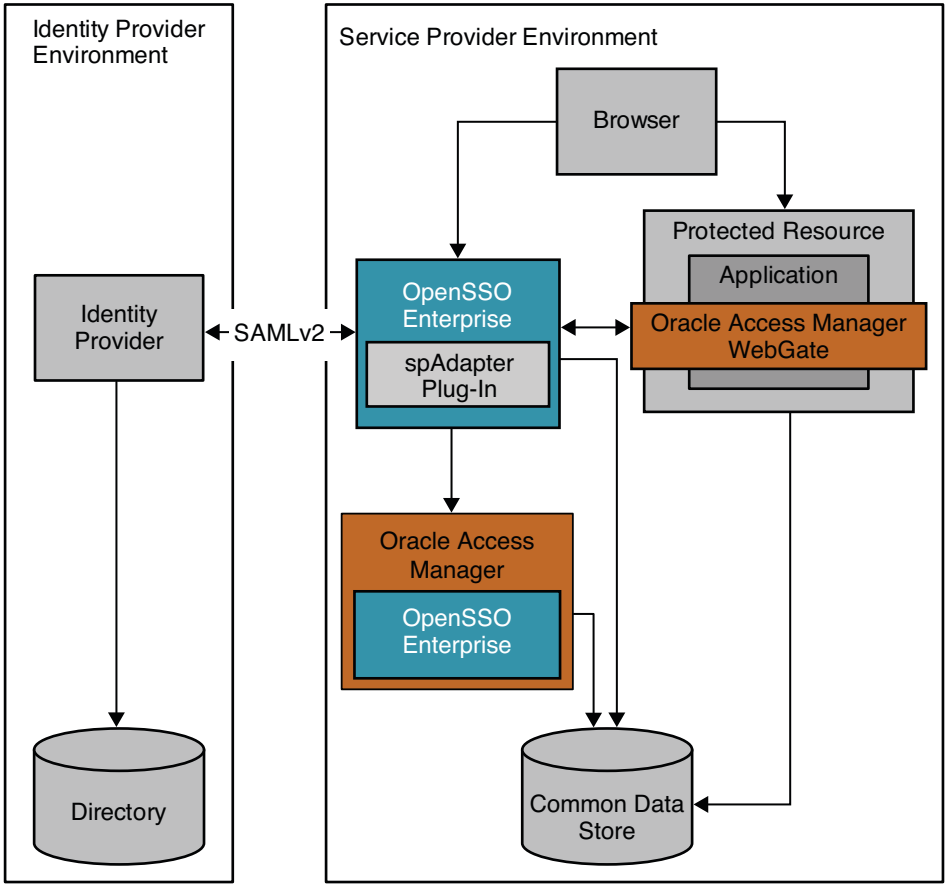


FIGURE 3–6 Oracle Access Manager Federation in a Service Provider Environment

The following two figures illustrate the process flow among components in the Identity Provider environment and Service Provider environment.

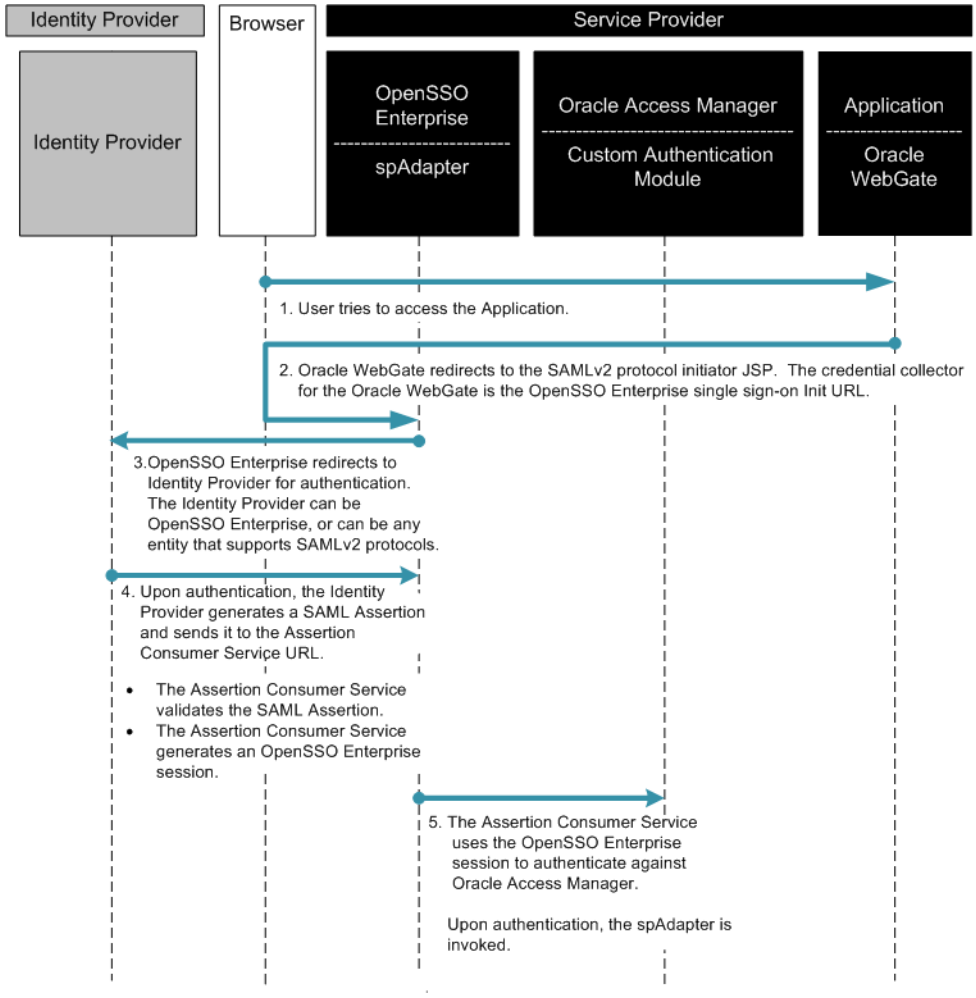


FIGURE 3–7 Process Flow for Oracle Access Manager Federation in a Service Provider Environment

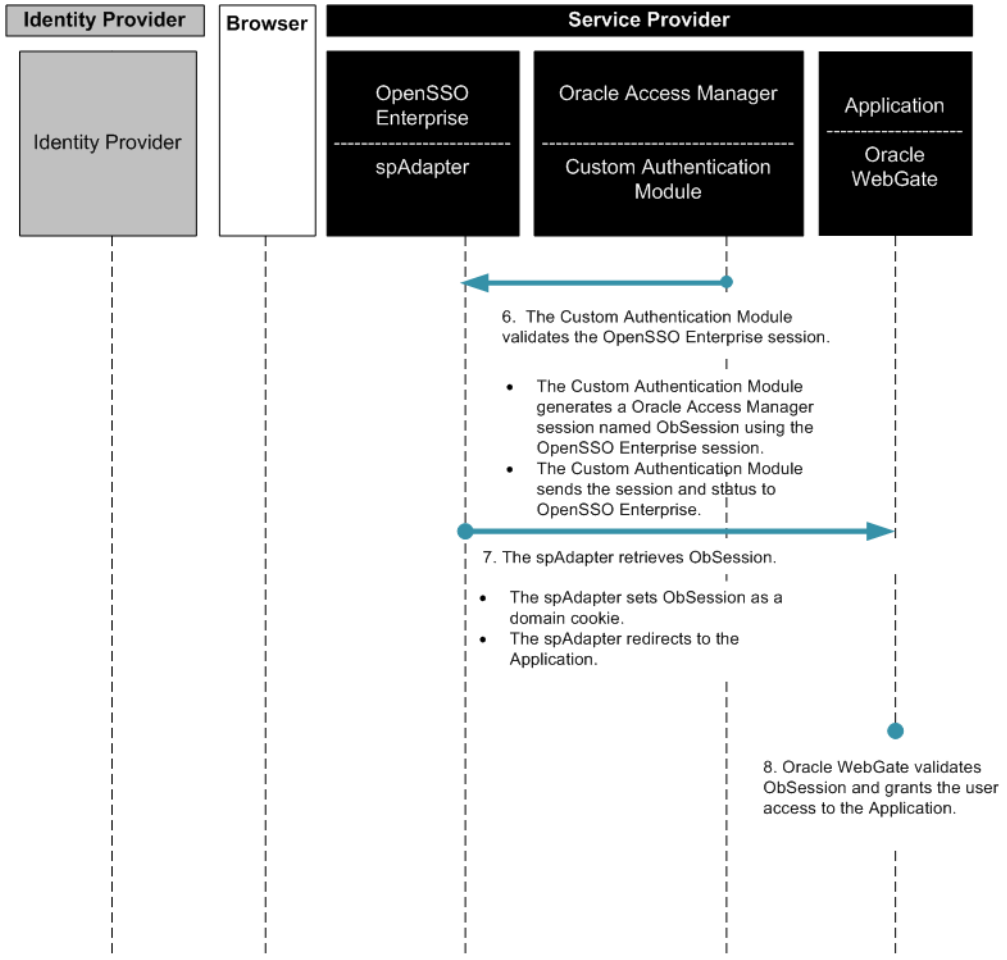


FIGURE 3–8 Process Flow for Oracle Access Manager Federation in a Service Provider Environment (continued)

Installing and Configuring Oracle Access Manager

To enable the legacy Oracle Access Manager single sign-on applications for SAML 2 federation protocols using OpenSSO Enterprise 8.0, follow these steps:

1. "To Install Oracle Access Manager and Oracle Access Manager Web Policy Agent" on page 219
2. "To Configure Oracle Access Manager" on page 220

▼ To Install Oracle Access Manager and Oracle Access Manager Web Policy Agent

1 **Install Oracle Identity Server, and then install the Oracle Access Server component.**

Obtain all required Oracle Access Manager components before you begin installation procedures. See *Oracle Access Manager Installation Guide*for detailed installation instructions.

For the examples in this document, Solaris-based installation was conducted. The system was tested with Sun Web Server 6.1 SP5 as the Oracle Administration plug-in interface, and Sun Directory Server 6.3 as the user data and configuration repository.

2 **After the successful installation, access the administration console.**

Go to the URL `http://`*host*`:`*port*`/access/oblix` and log in using the following credentials:

User Name: oadmin

Password: password

The administrative interface for managing core access server components, policy manager, and identity console is displayed.

3 **Install Oracle WebGate.**

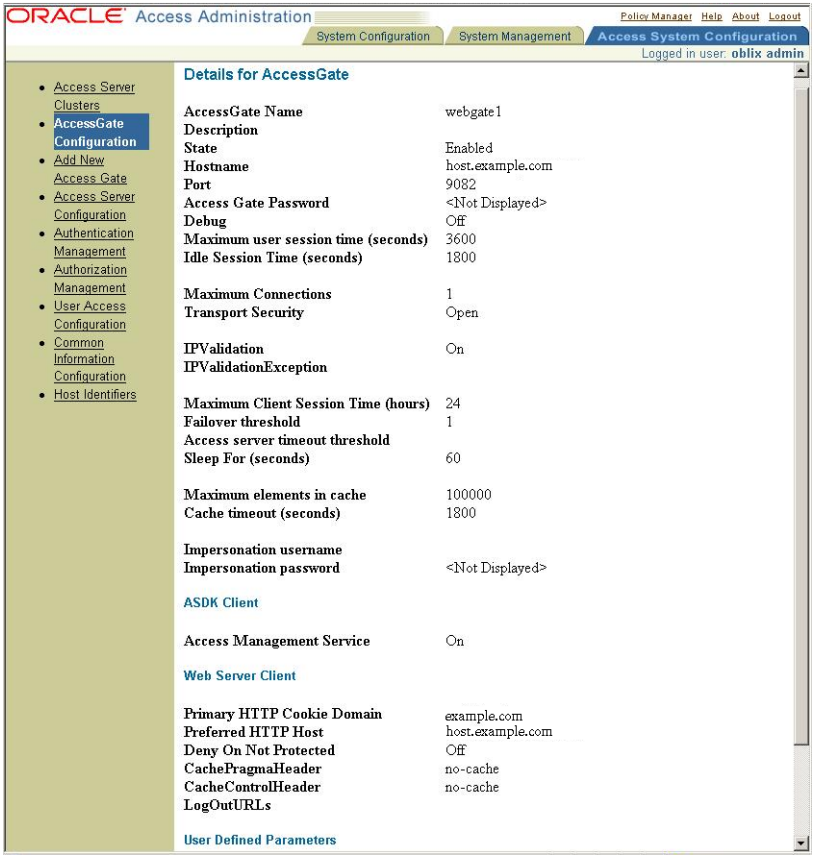
See the sectionInstalling the WebGate in the *Oracle Access Manager Installation Guide*.

▼ To Configure Oracle Access Manager

See the *Oracle Access Manager Installation Guide* for detailed configuration instructions. For the examples in this document, the Oracle Access and Policy Servers are tested using the configurations described below.

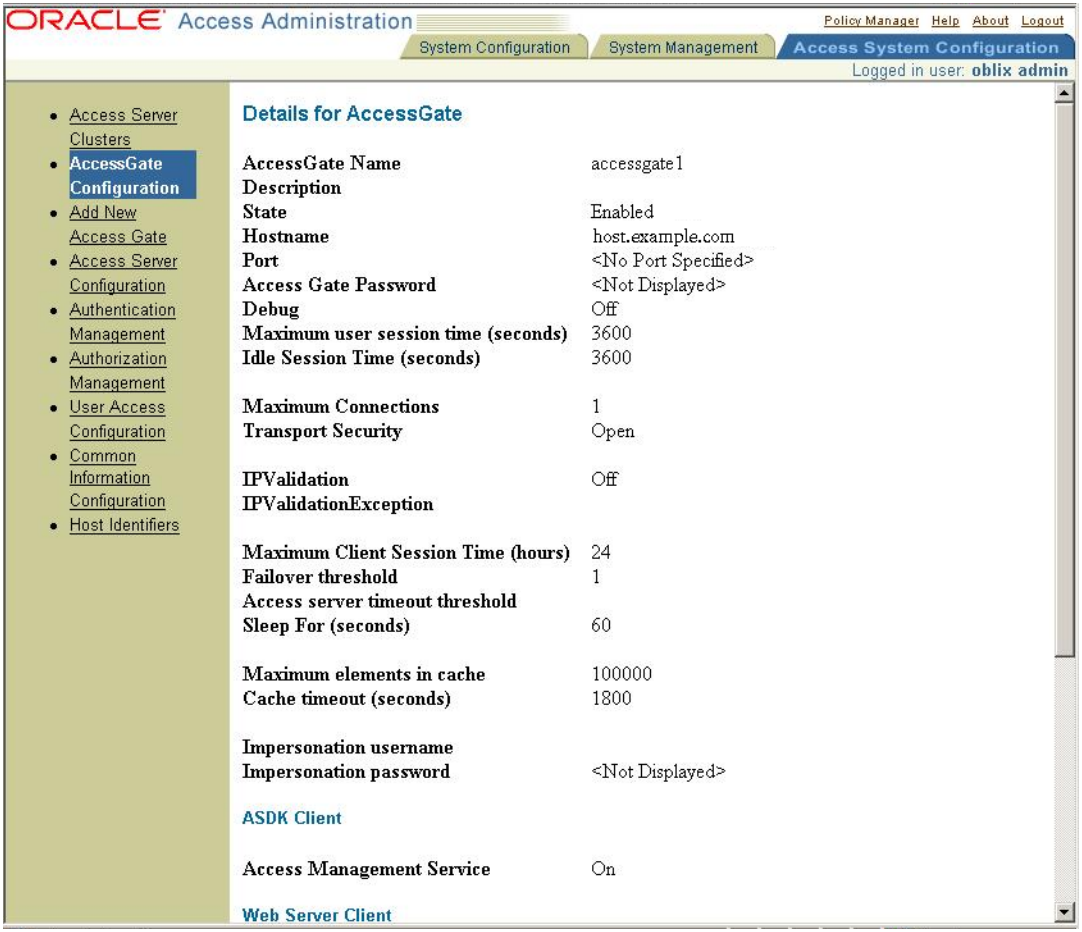
1 Create an Access Server Configuration named `access1`.

2 Create access gate configuration named `webgate1`.



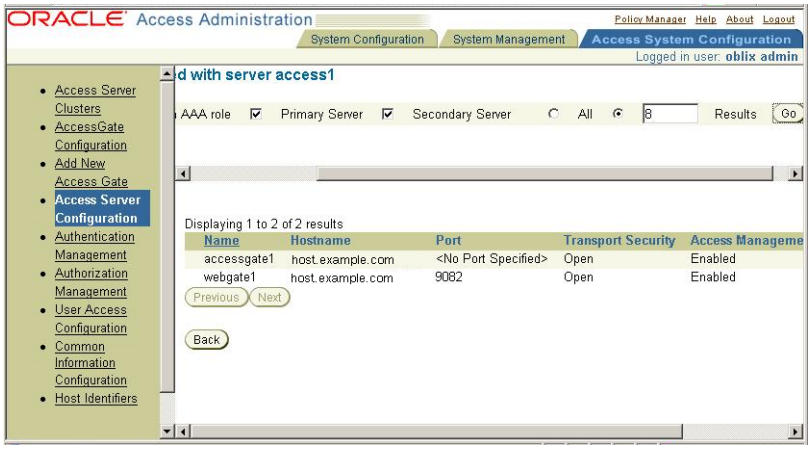
3 Create an access gate configuration for the SDK.

The SDK configuration is used for custom authentication modules and for other remote APIs.



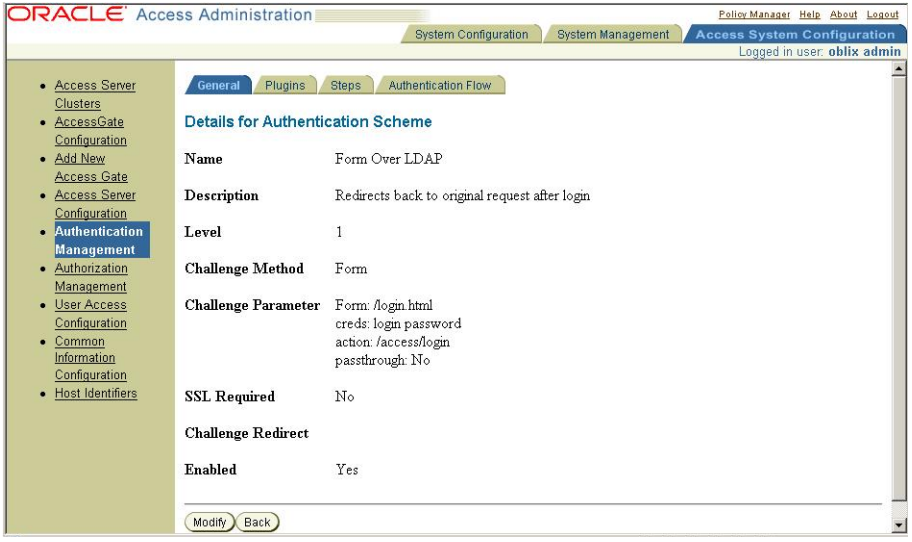
4 Associate the web gates with Oracle Access Server.

This establishes a trust relationship.



5 Create a form-based authentication scheme

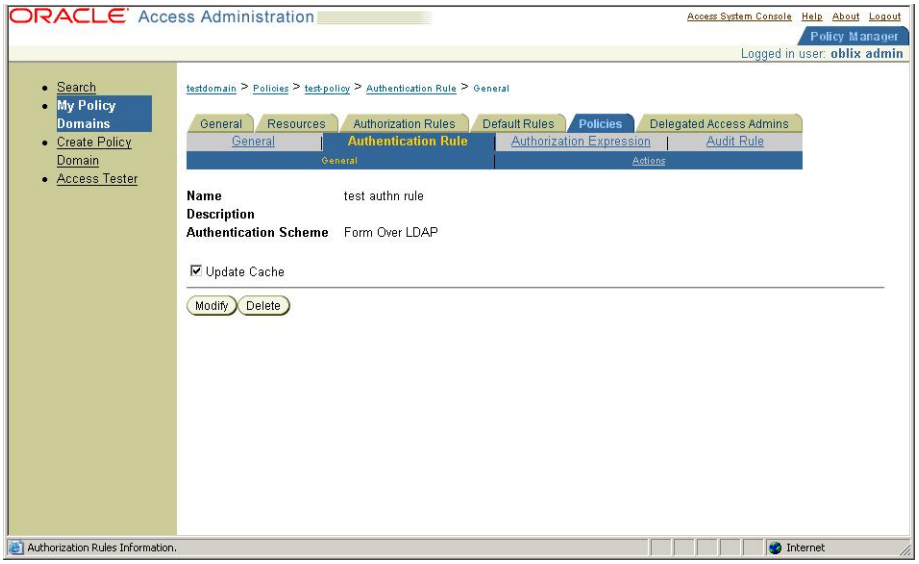
By default, Oracle Access Manager provides a credential collector form you can use it. You can also customize the form. For the examples in this document, the following properties are used.



6 Configure the plug-ins.

7 Access the Policy Manager console (top-right link) and create a policy for your protected resource.

Protect the resource with the form-based authentication.



Accessing your protected application should redirect to the form login page. Upon successful authentication, the protected application will redirect to the protected resource with a valid Oracle Access Manager session.

Using OpenSSO Enterprise to Enable Oracle Federation in the Identity Provider Environment

To enable Oracle Access Manager for federation in the Identity Provider environment, follow these steps:

1. Install and configure OpenSSO Enterprise in the Identity Provider container.
2. Install and configure Oracle Web Gate.
3. Install the Custom Authentication Module.
4. Install and configure OpenSSO Enterprise in the Service Provider container.
5. Set up SAML2.
6. Configure OpenSSO Enterprise for SAMLv2 Identity Provider protocols.
7. Configure Oracle Access Manager Agent to protect OpenSSO Enterprise URLs.
8. Configure the Service Provider.
9. Verify that Single Sign-On is working properly.

Installing and Configuring OpenSSO Enterprise in the Identity Provider Container

Follow the installation instructions in the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*. Make sure that the Identity Provider container is one of the supported Oracle Web Gate containers. Also make sure that the user repository is same as the Oracle Access Manager so that both OpenSSO Enterprise and Oracle Access Manager provide a session for the same user.

Installing and Configuring the Oracle WebGate

Follow the instructions in the section Installing the WebGate in the *Oracle Access Manager Installation Guide*.

Make sure that Oracle single sign-on is working for the protected URLs. Do not protect the OpenSSO Enterprise URLs yet because you must first configure OpenSSO Enterprise for authentication modules. For now, protect a temporary URL to ensure that Oracle WebGate is working properly. A temporary policy in Oracle Access Manager could be configured as in the following figure:

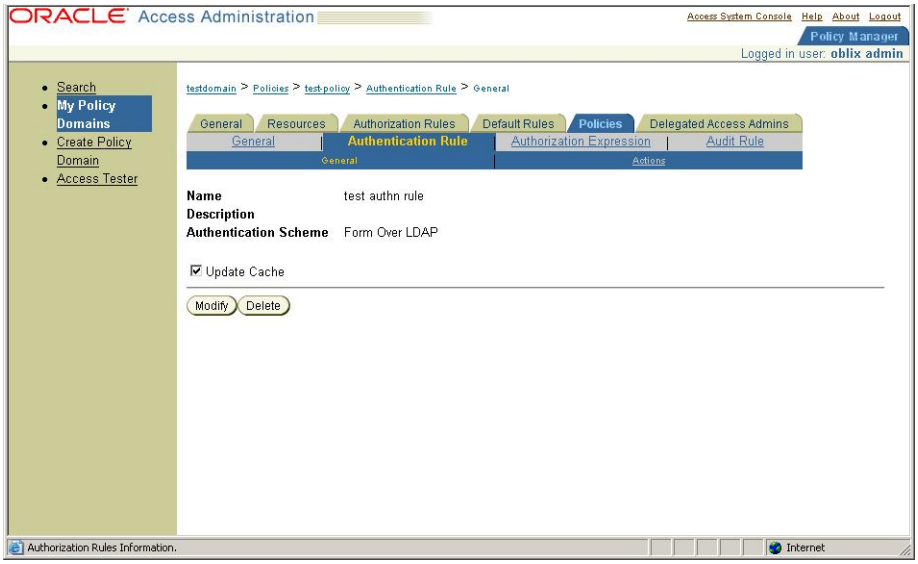


FIGURE 3–9 Configuring a temporary policy

Installing the Custom Oracle Authentication Module

In OpenSSO Enterprise, install a custom authentication module for Oracle Access Manager. After exploding the OpenSSO Enterprise WAR file, the custom authentication module is located under the directory *base-dir*`/samples/thirdparty/oblix`. Follow the instructions in the `README.txt` file for building and configuring a custom authentication module. Make sure that the custom authentication module is working before proceeding.

The custom authentication module implements the OpenSSO Enterprise pluggable interface `AMLoginModule`. The `AMLoginModule` interface validates the Oracle Access Manager session using the Oracle Access Manager SDK , and then creates an OpenSSO Enterprise session.

To configure the custom Oracle Authentication Module, provide the following information:

OblixCookieName:	Name of the Oblix session cookie
OblixSDKInstallDir:	Directory where the Oblix SDK is installed and configured. Make sure that `ObAccessClient.xml` is configured.
RemoteUserHeaderName:	The name of the header for an authenticated user after successful authentication.
CheckRemoteUserOnly:	If true, the authentication modules looks only for the remote user header name.

Installing and Configuring OpenSSO Enterprise in the Service Provider Container

Follow the installation instructions in the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*. A good practice is to install the Identity Provider and Server Provider in different domains. If that is not possible, make sure they use different cookie names or cookie domains. You do not have to install the OpenSSO Enterprise web policy agent to protect the OpenSSO Enterprise URLs at this time. First make sure that SAML2 is set up and working properly.

Setting Up SAML2

Before loading metadata, read the following sections and be sure you understand the changes that must be made to the metadata. The SAML2 samples contains instructions on how to setup SAML2.

In all, you must have Identity Provider metadata and extended metadata, as well as Service Provider metadata and extended metadata. In the Identity Provider, import Identity Provider metadata and extended metadata as hosted metadata. Import Service Provider metadata and extended metadata as remote entity metadata. To change the hosted or remote attributes, locate the extended metadata XML element `<EntityConfig>` which contains the following attribute with default value:

```
hosted=true
```

Change the value to `false`.

▼ To Configure the Identity Provider OpenSSO Enterprise for SAMLv2 Identity Provider Protocols

1 Generate the metadata templates on both Identity Provider and Service Provider environments.

You can use one of the following methods:

- Use the `famadm` command.
- Use a browser:

  `http:`*host*`:`*port*`/opensso/famadm.jsp`

a. At the Identity Provider, run the following command:

```
famadm create-metadata-templ -y idp_entity_id -u amadmin
-f admin_password_file_name -m idp_standard_metadata
```

```
-x idp_extended_metadata -i idp_meta_alias
```

where `idp_meta_alias` is "`/idp`".

b. At the Service Provider, run the following command:

```
famadm create-metadata-templ -y sp_entity_id -u amadmin
-f admin_password_file_name -m sp_standard_metadata
-x sp_extended_metadata -s sp_meta_alias
```

where `sp_meta_alias` is "`/sp`".

2 Customize extended metadata.

Use one of the following options:

- **To the Identity Provider extended metadata, add an attribute named `AuthUrl`.**

  This URL attribute is used by the SAML protocols to redirect to an OpenSSO Enterprise authentication module. In this use case, the redirect is to the custom Oracle Authentication Module. Example:

  ```
  <Attribute name="AuthUrl">
  <Value>http://host:port/opensso/UI/Login?module=OAMAuth</Value>
  </Attribute>
  ```

- **Make the custom Oracle authentication module as the default login module in OpenSSO Enterprise.**

  A consequence of using this option is that you have to specify an LDAP login module for logging in as `adminstrator`. The Service Provider extended metadata has an attribute named as `transientUser`. Set this value to your anonymous user. Example:

  ```
  <Attribute name="transientUser">
     <Value>anonymous</Value>
  </Attribute>
  ```

3 Change the `hosted` attribute in the Identity Provider and Service Provider extended metadata when loading remote metadata.

For a remote Identity Provider or Service Provider, set the value to "`false`" or "`0`".

4 Load the metadata.

a. Create circle of trust.

Add the circle of trust to the extended metadata. In the extended template files, you will see a sample circle of trust. Edit the following to correspond to your circle of trust.

```
<Attribute name="cotlist">
<Value>samplesaml2cot</Value>
</Attribute>
```

b. Load the hosted metadata in both the Identity Provider and Service Provider.

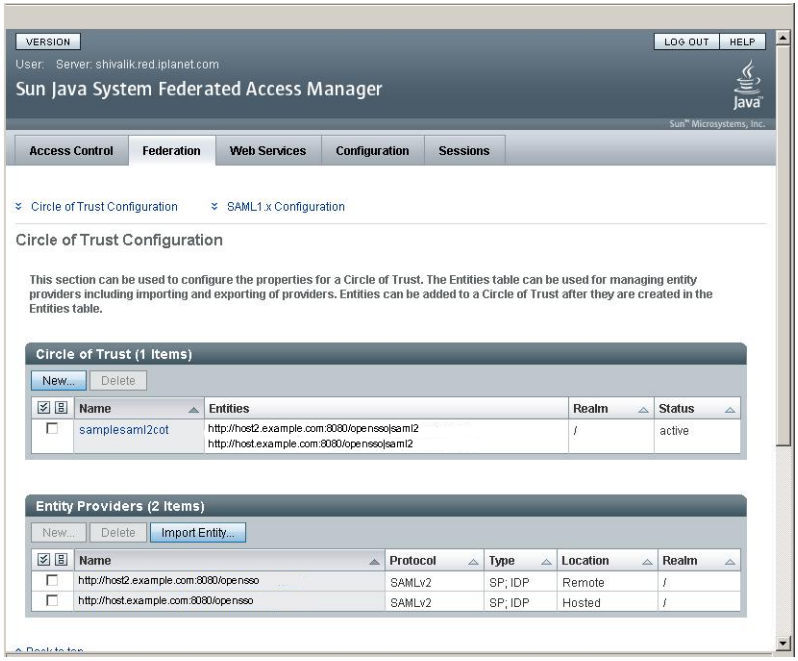
You can use either the `famadm` command or the OpenSSO Enterprise console.

c. Exchange the metadata .

Import the Service Provider metadata into the Identity Provider, and import the Identity Provider metadata into the Service Provider.

d. Load the metadata.

5 After successfully exchanging the metadata, verify through the OpenSSO Enterprise administration console that the metadata has been configured correctly.



To Configure Oracle Access Manager Agent to protect OpenSSO Enterprise URLs

There are many different ways to configure Oracle Access Manager Policy to protect OpenSSO Enterprise URLs. At minimum, you must configure a policy to protect the SAML Single Sign-On Service URL. The real-time policy can be different based on other deployment requirements. The Oracle Access Manager session must be established before the SAML Assertion is generated.

In Oracle Access Server Policy Console, create a policy domain named `fampolicy` to protect only the OpenSSO Enterprise Single Sign-On Service URL.

FIGURE 3–10 Creating a policy domain

▼ To Configure the Service Provider

1 **Install the OpenSSO Enterprise web policy agent in the Service Provider environment to protect OpenSSO Enterprise Service Provider.**

Follow the instructions in the *Sun Java System Access Manager Policy Agent 2.2 Release Notes*.

There is no restriction on the type of policy agent you use. However, be sure use an agent that is supported on the container where the application to be protected is deployed.

2 **Change the policy agent login URL.**

After verifying that simple single sign-on with the OpenSSO Enterprise works properly, change the policy agent login URL to the OpenSSO Enterprise SAML2 SP initiated Single Sign-on Service URL. Example:

```
http://<sphost>:<spport>/opensso/saml2/jsp/spSSOInit.jsp?metaAlias
=<SP MetaAlias>
&idpEntityID=<IDP Entity ID>&NameIDFormat=transient
```

▼ To Test the Single Sign-On

1 **Authenticate at Oracle Access Manager with username and password.**

2 **Access the enterprise application protected by the OpenSSO Enterprise Service Provider policy agent in the Service Provider environment.**

You should automatically be granted access to the protected application.

Using OpenSSO Enterprise to Enable Oracle Federation in a Service Provider Environment

To enable Oracle Access Manager for Federation in the Service Provider environment, follow these steps:

1. Install OpenSSO Enterprise in the Identity Provider environment.
2. Install OpenSSO Enterprise in the Service Provider environment.
3. Install Oracle Access Manager in the Service Provider domain.
4. Configure Oracle Access Manager for the OpenSSO Enterprise scheme.
5. Configure a resource.
6. Set Up SAMLv2.
7. Configure the OpenSSO Enterprise Identity Provider and Service Provider for SAML2 protocols.
8. Verify that single sign-on is working properly.

Installing OpenSSO Enterprise in the Identity Provider Environment

The Identity Provider does not have to be an OpenSSO Enterprise deployment. But for optimum protocol interoperability, use OpenSSO Enterprise. See the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*for detailed installation and configuration steps.

A good practice is to install the Identity Provider and Server Provider in different domains. If that is not possible, make sure they use different cookie names or cookie domains. This eliminates cookie validation inconsistency.

Installing OpenSSO Enterprise in the Service Provider Environment

See the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*for detailed installation and configuration steps.

The OpenSSO Enterprise in the Service Provider environment initiates the SAML2 protocols. The Oracle Access Manager Agent can protect the enterprise application by redirecting to OpenSSO Enterprise for single sign-on purposes.

Installing Oracle Access Manager

Install Oracle Access Manager in the Service Provider domain where enterprise applications are protected by Oracle WebGate agents. See the *Oracle Access Manager Installation Guide* for detailed installation instructions.

The plug-in name must be same as the name of the shared library.

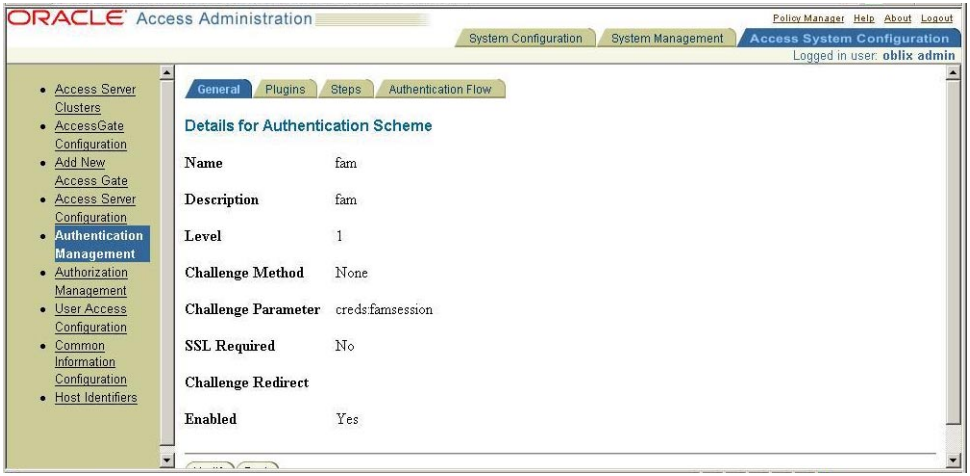


FIGURE 3–11 The plug-in name must be same as the name of the shared library

Configuring Oracle Access Manager for an OpenSSO Enterprise Scheme

The custom authentication scheme for Oracle Access Manager is a C-based implementation, and the custom authentication scheme should be built like a shared library. The custom authentication scheme in this chapter is a Solaris-based shared library and can be ported onto other platforms with similar semantics. This custom authentication module also uses the OpenSSO Enterprise C-SDK for validating the OpenSSO Enterprise session. When an OpenSSO Enterprise WAR file is exploded, the custom authentication module is located under the *base-dir*`/samples/thirdparty/oblix` directory. The `README.txt` contains instructions for configuring the Oracle Access Manager authentication scheme. The following figure provides some details for configuring OpenSSO Enterprise `AuthScheme`in Oracle Access Manager.

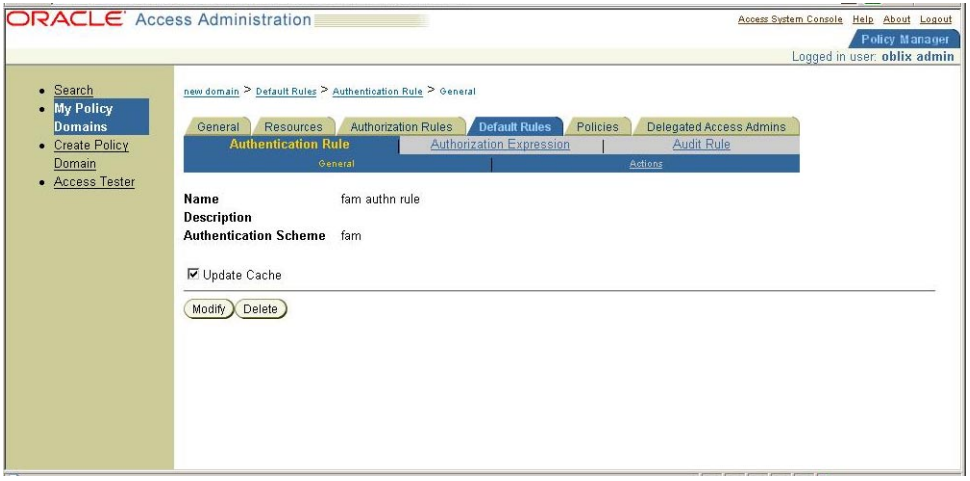


FIGURE 3–12 Configuring OpenSSO Enterprise `AuthScheme`in Oracle Access Manager

Configuring a Resource

For detailed instructions, see the *Oracle Containers for J2EE Security Guide* (`http://download-west.oracle.com/docs/cd/B31017_01/web.1013/b28957/coreid.htm#BJEDGDGFB`)

Later you will use the resource name you configure in Oracle Access Manager in the SAML Adapter configuration. The policy will trigger the OpenSSO Enterprise authentication module.

Setting Up SAMLv2

In all, you must have Identity Provider metadata and extended metadata, as well as Service Provider metadata and extended metadata. In the Identity Provider, import Identity Provider metadata and extended metadata as hosted metadata. Import Service Provider metadata and extended metadata as remote entity metadata. To change the hosted or remote attributes, locate the extended metadata XML element `<EntityConfig>` which contains the following attribute with default value:

```
hosted=true
```

Change the value to `false`.

▼ To Configure the OpenSSO Enterprise Identity and Service Providers for SAML2 Protocols

1 **Generate the metadata templates on both Identity Provider and Service Provider environments.**

Use the `famadm` command, or used use a browser to go to the following URL:

```
http:<host>:<port>/opensso/famadm.jsp
```

- At the Identity Provider:

```
famadm create-metadata-templ -y idp_entity_id
-u amadmin -f admin_password_file_name -m idp_standard_metadata
-x idp_extended_metadata -i idp_meta_alias
```

where `idp_meta_alias` is `/idp`

- At the Service Provider:

```
famadm create-metadata-templ -y sp_entity_id
-u amadmin -f admin_password_file_name -m sp_standard_metadata
-x sp_extended_metadata -s sp_meta_alias
```

where `sp_meta_alias` is `/sp`

2 Customize the Service Provider extended metadata.

Add the Service Provider extended metadata as an attribute named as `spAdapter`. This attribute is used by the SAML protocols to do any post single sign-on authentication processes. In the architecture diagram, this the Oracle Access Manager Plug-in. The `OAMPlugin` uses the OpenSSO Enterprise session to authenticate against Oracle Access Manager and establish `ObSSOCookie`. The Service Provider metadata must have the following attributes:

```
<Attribute name="spAdapter">
<Value>com.sun.identity.saml2.plugins.SMAdapter</Value>
</Attribute>

<Attribute name="spAdapterEnv">
<Value>FAMCookieName=iPlanetDirectoryPro</Value>
<Value>OAMCookieName=ObSSOCookie</Value>
<Value>CookieDomain=.red.example.com</Value>
<Value>Resource=/test/index.html</Value>
<Value>ObSDKInstallDir=/export/oam/AccessServerSDK</Value>
</Attribute>
```

3 Set the value for `transientUser` to the anonymous user.

The Service Provider extended metadata has an attribute named as `transientUser`. Make sure that the OpenSSO Enterprise Service Provider is enabled for Anonymous authentication.

```
<Attribute name="transientUser">
<Value>anonymous</Value>
</Attribute>
```

4 Create a circle of trust.

The circle of trust should also be added in your extended metadata.

5 Load the metadata.

6 Edit the following attribute to one of your circle of trust.

The extended template files contains a sample circle of trust.

```
<Attribute name="cotlist">
<Value>samplesaml2cot</Value>
</Attribute>
```

You can also add the circle of trust through the OpenSSO Enterprise administration console.

7 **Load the hosted metadata in both the Identity Provider and Service Provider.**
You can use the `famadm` command or the OpenSSO Enterprise administration console.

8 **Exchange the metadata between Identity Provider and Service Provider.**
and load the metadata.

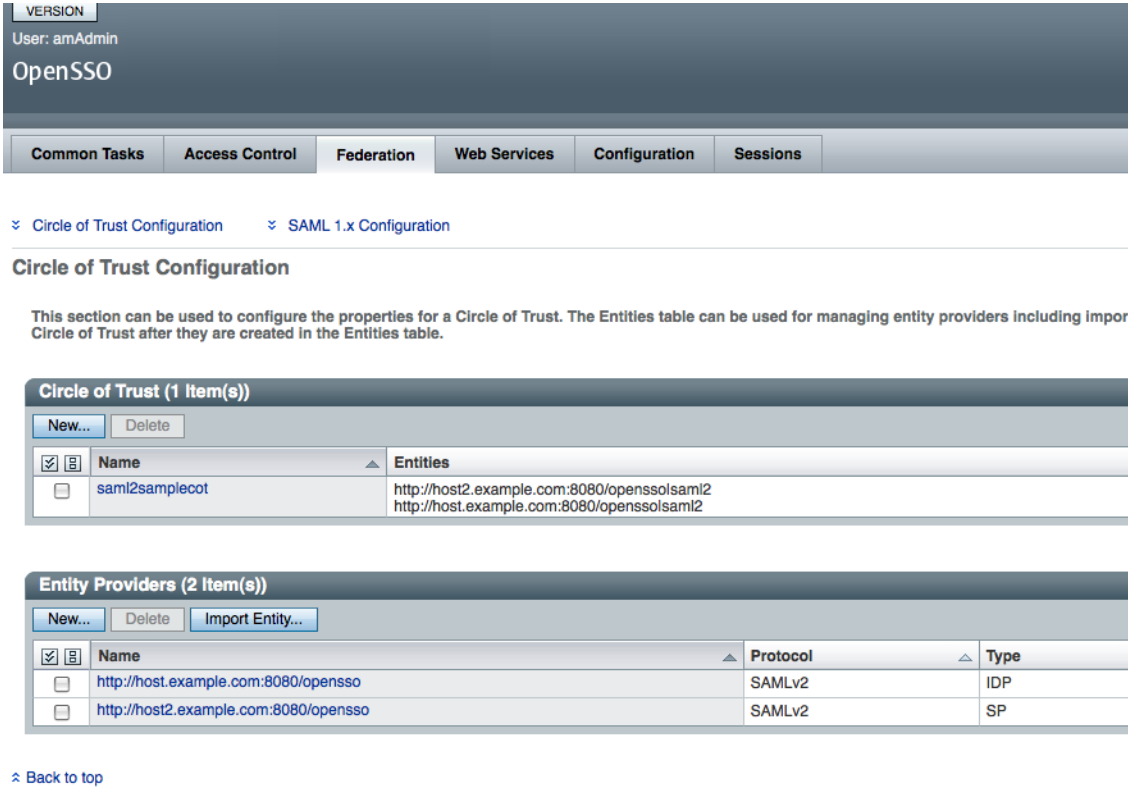
   a. **Import the Identity Provider metadata into the Service Provider metadata.**

   b. **Import the Service Provider metadata into the Identity Provider metadata.**

   c. **Change the `hosted` attribute value in the extended metadata to `false`.**

   d. **Load all metadata.**

9 **Verify through OpenSSO Enterprise administration console that the metadata is configured properly.**



Verifying that Single Sign-On Works Properly

Access the enterprise application protected by Oracle WebGate. Oracle WebGate redirects to OpenSSO Enterprise for authentication where the SAML2 single sign-on is initiated.