Oracle® OpenSSO STS

Administrator's Guide

Release 11*g*R1. Version 11.1.1.3.0

E17844-01

August 2010

Provides instructions for configuring and managing an Oracle OpenSSO Security Token Service server.

ORACLE®

Oracle OpenSSO STS Administrator's Guide, Release 11*g*R1. Version 11.1.1.3.0

E17844-01

Primary Author: Gina Cariaga

Contents

Part II Basic Server Administration

2 Overview of OpenSSO Security Token Service

3 Getting Started Using the OpenSSO STS Console

4 Managing the Security Token Service

5 Configuring OpenSSO STS System Properties

6 Managing the OpenSSO STS Authentication Service

7 Using the Logging Service

List of Tables

Preface

The *Oracle OpenSSO STS Administrator's Guide* describes the features, architecture, and administration of Oracle OpenSSO Security Token Service (OpenSSO STS). This document also includes detailed installation instructions, including system requirements and other presrequisite information.

Audience

This manual is intended for anyone who performs system administration tasks for the OpenSSO STS server. You should be familiar with either the UNIX operating system or the Microsoft Windows NT operating system in order to understand the line-mode commands and examples. You can perform most of the tasks through the OpenSSO STS administration console which is platform-independent.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/support/contact.html` or visit

http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Related Documents

For more information, see the Oracle OpenSSO Security Token Service Release Notes at the following URL: http://download.oracle.com/docs/cd/E17842_01/doc.1111/e17846/toc.htm

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Installation

Part I contains one chapter, Chapter 1, "Installing OpenSSO Security Token Service" which contains the following topics:

- Meeting the System Requirements
- Installing a Web Container
- Installing the OpenSSO STS Server
- Configuring OpenSSO STS Using the Command-Line Configurator
- Configuring the OpenSSO STS Administrator Password
- Installing the OpenSSO STS Command-Line Utility
- Uninstalling the OpenSSO STS Server

1

Installing OpenSSO Security Token Service

This chapter provides important prerequisite information and instructions for installing the Oracle OpenSSO Security Token Service (OpenSSO STS) server. The following topics are contained in this chapter:

1.1 Meeting the System Requirements

For information about system hardware requirements and supported platforms, see http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html

Also be sure you meet the additional database logging requirements.

1.2 Installing a Web Container

Before you can install OpenSSO STS, you must install a supported web container. See the product documentation for installation instructions for the web container into which you will deploy OpenSSO STS. The following sections provide useful links and important information about additional steps you must complete for supported web containers.

1.2.1 Using Oracle WebLogic Server

Download site

http://www.oracle.com/technology/software/products/ias/htdocs/101310.html

Product Documentation

http://download.oracle.com/docs/cd/E14571_01/doc.1111/e14142/toc.htm

Java Permission
Required if Java Security Manager is enabled: server.policy

OpenSSO STS Pre-Deployment Steps
After installing Oracle WebLogic Server, complete the following steps.

1. Set the `MaxPermSize` JVM option to a minimum value of 256 MB. For example:

   ```
   -XX:MaxPermSize=256M
   ```

2. If the Java Security Manager is enabled, add the security permissions to the weblogic.policy file. After you edit the file, restart the web container.

1.2.2 Using GlassFish Application Server

Download Sites
- GlassFish V2 UR1:
  https://glassfish.dev.java.net/downloads/v2ur1-b09d.html
- GlassFish V2 UR2:
  https://glassfish.dev.java.net/downloads/v2ur2-b04.html

Product Documentation
https://glassfish.dev.java.net/

Java Permission
Required if Java Security Manager is enabled: `server.policy`

OpenSSO STS Pre-Deployment Steps
After installing GlassFish Application Server, complete the following steps.

1. In the GlassFish domain where you plan to deploy OpenSSO STS server, change the following JVM options either using the GlassFish administration console or by editing the `domain.xml file`:
   - Change `-client` to `-server`.
   - Change `-Xmx512m` to `-Xmx1024m`.
2. If the Java Security Manager is enabled:
   - Set the following JVM option:

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

   - Add the security permissions to the `server.policy` file. After you edit the file, restart the web container.

1.3 Installing the OpenSSO STS Server

Before you can install the OpenSSO STS Server, a supported web container must already be installed and running.

1.3.1 Downloading the OpenSSO STS WAR

OpenSSO STS is contained in the openssosts_11gr1.zip file. Download openssosts_11gr1.zip from the Oracle Downloads site:

http://www.oracle.com/technology/software/products/middleware/htdocs/fmw_11_download.html

1.3.2 Unpacking openssosts.war

To unzip the openssosts_11gr1.zip file, run the following command:

```
# unzip openssosts_11gr1.zip
```

The following table describes the file layout after you unzip the openssssts.zip file. The directory where you unzip the file is represented by zip-root.

Table 1–1 OpenSSO STS opensso_sts.zip File Layout

File or Directory	Description
openssosts.war	WAR file containing all OpenSSO STS components. Use this file you to deploy the OpenSSO STS server.
tools	Directory containing the following: ▪ stsAdminTools.zip contains files to setup and run the OpenSSO STS command-line (CLI) ssoadm utility. ▪ sts.Configurator.zip contains files the interface for configuring OpenSSO STS at first-time login.
version	File containing version and release indentifiers.

1.3.3 Deploying the OpenSSO STS WAR File

Caution: If you plan to use the OpenSSO configuration data store, you must deploy OpenSSO STS on a local file system and not on an NFS-mounted file system. The OpenSSO STS configuration data store, which is deployed with OpenSSO STS, is not supported on an NFS-mounted file system.

1.3.4 To Deploy the OpenSSO STS WAR

1. Be sure you have proper access permissions on the host computer where you will openssosts.war.
   - If you plan to deploy openssosts.war using the web container administration console, then set permissions for accessing the administration console.
   - If you plan to deploy openssosts.war using the web container's deploy command-line utility, then set permissions for accessing the command-line utility.
2. Log in to the host computer where you want to deploy openssosts.war.
3. Copy openssosts.war to the host computer where you want to deploy OpenSSO STS.
4. Deploy openssosts.war using either the web container administration console or deploy command.

   See Section 1.3.4.2, "Deploying openssosts.war GlassFish Version 3" and Section 1.3.4.1, "Deploying openssosts.war on Oracle WebLogic Application Server".

1.3.4.1 Deploying openssosts.war on Oracle WebLogic Application Server

When deploying openssosts.war, WebLogic Application takes a python script as a configuration parameter. See http://download.oracle.com/docs/cd/E14571_01/web.1111/e13715/using_wlst.htm#i1063337 for information about using the WebLogic Scripting Tool.

1. Create a python script that includes the following:

```
evaluatePrompt()
connect('weblogic','password','hostname.domain.com:port')
updateGlobals()
print ''
restoreDisplay()
deploy(appName='openssosts', path='/export/deploy-directory/openssosts.war',
targets='AdminServer', plan='true')
restoreDisplay()
disconnect()

now run the command

$MIDDLEWARE_HOME/wlserver_10.3/common/bin/wlst.sh $PYTHON_DEPLOY_SCRIPT_FILE
```

2. Run the following command:

```
$MIDDLEWARE_HOME/wlserver_10.3/common/bin/wlst.sh $PYTHON_DEPLOY_SCRIPT_FILE
```

1.3.4.2 Deploying openssosts.war GlassFish Version 3

The following command deploys opensso.war on the GlassFishv3 container on Solaris systems:

```
# cd /opt/glassfishv3/glassfish/bin
# ./asadmin deploy --user admin --passwordfile /tmp/pwdfile
--port 4848 zip-root/opensso/deployable-war/opensso.war
```

where:

- zip-root is where you unzipped the openssosts_11gr1.zip Or, if you copied openssosts.war to a different location, use that location in the command.
- /tmp/*pwdfile* is the GlassFishv3 password file. This ASCII text file contains the AS_ADMIN_PASSWORD variable set to the administrator password.

1.4 Configuring OpenSSO STS Using the Command-Line Configurator

OpenSSO STS includes the command-line Configurator to perform the initial configuration of an OpenSSO STS server instance. To configure the OpenSSO STS server using the command-line Configurator, you set parameters in a configuration file and then run the Configurator from the command line using the configuration file as input. You can run the Configurator on the same system as OpenSSO STS server or from a remote system.

1.4.1 Before You Begin

Before running the command-line Configurator, be sure you have met the following requirements:

- You have downloaded and unzipped the openssosts_11gr1.zip file.
- You have deployed the opensso.war file in a supported web container.
- The web container is running.
- Your JAVA_HOME environment variable points to a JDK 1.6 or later.

1.4.2 Installing the Command-Line Configurator

After you unzip the openssosts_11gr1.zip file, the command-line Configurator and related files are in the following file:

zip-root/opensso/tools/stsConfiguratorTools.zip

where zip-root is the directory where you unzipped openssssts.zip.

1.4.2.1 To Install the Command-Line Configurator

1. Change to the *zip-root*/opensso/tools directory.
2. Unzip the stsConfiguratorTools.zip file to get these files:

File	Description
README.setup	Explains how to run the Configurator.
configurator.jar	Contains the binary files OpenSSOConfigurator.class and OpenSSOConfigurator.properties.
sampleconfiguration	Sample input file that you edit before you run the Configurator
license.txt	Describes the Common Development and Distribution License (CDDL)

Remote system. If you plan to run the Configurator on a remote system, copy the stsConfiguratorTools.zip file to the remote system before you unzip it.

1.4.3 To Configure the OpenSSO STS Server

1. Make sure your JAVA_HOME environment variable points to JDK 1.6 or later.
2. Change to the directory where you unzipped the stsConfiguratorTools.zip file.
3. Create a configuration file and set the properties required for your deployment.

   OpenSSO STS provides server configuration parameters in the sampleconfiguration file. Either edit sampleconfiguration and use it when you run the Configurator, or copy this file and edit the new file. See Section 1.4.3.1, "OpenSSO STS Configuration Parameters For the Command-Line Configurator" for more information.
4. Run the Configurator. For example:

   ```
   # java -jar configurator.jar -f configuration-file
   ```

   where configuration-file contains the configuration properties you set in the previous step.

1.4.3.1 OpenSSO STS Configuration Parameters For the Command-Line Configurator

The following table lists General and Server parameters with a description for each parameter.

Table 1–2 Configurator - General and Server Parameters

Parameter	Description
SERVER_URL	The URL of the web container on which OpenSSO STS server is deployed. For example: SERVER_URL=http://stshost.example.com:58080
DEPLOYMENT_URI	The OpenSSO STS server deployment URI. Default: DEPLOYMENT_URI=/opensso
BASE_DIR	The configuration directory. Default: BASE_DIR=/opensso
PLATFORM_LOCALE	The OpenSSO STS server locale. Default: locale=en_US The default is en_US (US English). Other values can be de (German), es (Spanish), fr (French), ja (Japanese), zh (Chinese), or zh_TW (Simplified Chinese).
AM_ENC_KEY	The password encryption key. In a multi-server installation, this parameter must have the same value as the other servers. By default, AM_ENC_KEY is set to blank, which means that OpenSSO STS Server will generate a random password encryption key. If you specify a password encryption key, the key must be at least 8 characters. If this configuration will be part of an existing deployment, the password encryption key you enter must match that of the original deployment.
ADMIN_PWD	The password for the default OpenSSO STS administrator, amAdmin. The password must be at least 8 characters in length. If this configuration will be part of an existing deployment, the password you enter must match that of the original deployment.
COOKIE_DOMAIN	The name of the trusted DNS domain that OpenSSO STS server returns to a browser when it grants a session ID to a user. For example: COOKIE_DOMAIN=.example.com
AMLDAPUSERPASSWD	The password for default policy agent user [UrlAccessAgent].

The following table lists Configuration Date Store parameters with a description for each parameter.

Table 1–3 Configurator - Configuration Data Store Parameters

Parameter	Description

DATA_STORE	The type of configuration data store. Values can be:
	■ embedded - OpenSSO configuration data store
	■ dirServer - Sun Java System Directory Server
	If `DATA_STORE=dirServer`is specified, then:
	■ The value for USERSTORE_TYPE under the "User Data Store Parameters" must be either LDAPv3ForAMDS or LDAPv3. The USERSTORE_TYPE cannot be blank or commented out.
	You must specify all of the relevant parameters for the user data store. For example:
	`#Config Store Details`
	`DATA_STORE=dirServer`
	`DIRECTORY_SSL=SIMPLE`
	`DIRECTORY_SERVER=configurationdatastore.example.com`
	`DIRECTORY_PORT=5002`
	`ROOT_SUFFIX=dc=opensso,dc=java,dc=net`
	`DS_DIRMGRDN=cn=puser,ou=DSAME`
	`Users,dc=opensso,dc=java,dc=net`
	`DS_DIRMGRPASSWD=password`
	
	`# User Store Details`
	`USERSTORE_TYPE=LDAPv3ForAMDS`
	`USERSTORE_SSL=SIMPLE`
	`USERSTORE_HOST=userdatastore.example.com`
	`USERSTORE_PORT=5002`
	`USERSTORE_SUFFIX=dc=opensso,dc=java,dc=net`
	`USERSTORE_MGRDN=cn=puser,ou=DSAME`
	`Users,dc=opensso,dc=java,dc=net`
	`USERSTORE_PASSWD=password`
	■ If the configuration data store contains the configuration of existing OpenSSO STS servers, this OpenSSO STS server will be added to the existing multi-server setup.

DIRECTORY_SSL	Specifies if the configuration data store is using SSL. ■ SSL specifies that SSL is used. ■ SIMPLE specifies that SSL is not used. Example: `DIRECTORY_SSL=SIMPLE`
DIRECTORY_SERVER	The fully qualified host name of the configuration data store. For example: `DIRECTORY_SERVER=ds.example.com`
DIRECTORY_PORT	The port on which the configuration data store is listening for connections. For example: `DIRECTORY_PORT=50389`
ROOT_SUFFIX	The initial or root suffix of the configuration data store. For example: `ROOT_SUFFIX=dc=opensso,dc=java,dc=net`
DS_DIRMGRDN	The user who has read and write privileges to the root suffix and schema (cn=schema) in the configuration data store. Default: `DS_DIRMGRDN=cn=Directory Manager`
DS_DIRMGRPASSWD	The password for the `DS_DIRMGRDN` user.

The following table lists Multi-Server Deployment parameters with a description for each parameter.

Table 1–4 Configurator - Multi-Server Deployment Parameters

Parameter	Description
S_EMB_REPL_FLAG	Flag that enables the configuration data store in a multi-server setup. This flag is valid only if `DATA_STORE=embedded`. To enable this flag, set the value to embReplFlag. For example: `DS_EMB_REPL_FLAG=embReplFlag`
DS_EMB_REPL_ REPLPORT1	The replication port of the configuration data store of the new OpenSSO STS server. For example: `DS_EMB_REPL_REPLPORT1=58989`
DS_EMB_REPL_HOST2	The host name of the existing OpenSSO STS server. For example: `DS_EMB_REPL_HOST2=host2.example.com`
S_EMB_REPL_PORT2	The listening port of the configuration data store of the existing OpenSSO STS server. For example: `DS_EMB_REPL_PORT2=50389`
DS_EMB_REPL_ REPLPORT2	The replication port of the configuration data store of the existing OpenSSO STS server. For example: `DS_EMB_REPL_REPLPORT2=50889`

1.4.3.2 User Data Store Parameters

The following table lists User Data Store parameters with a description for each parameter.

Table 1–5 Configurator - User Data Store Parameters

Parameter	Description
USERSTORE_TYPE	The type of user data store. Values can be: ■ LDAPv3ForAMDS: LDAP with OpenSSO Schema ■ LDAPv3: Generic LDAP (no OpenSSO Schema) ■ Blank (`USERSTORE_TYPE=`): The configuration data store will be the same as the user data store. DATA_STORE must be embedded. The remaining user data store properties will be ignored.
USERSTORE_SSL	Specifies if the user data store is using SSL. Values can be: ■ SSL specifies that SSL is used. ■ SIMPLE specifies that SSL is not used.
USERSTORE_HOST	The host name of the user data store. For example: `ssohost.example.com`
USERSTORE_PORT	The port on which the user data store is listening for connections. Default is 389.
USERSTORE_SUFFIX	The initial or root suffix of the user data store. For example: `dc=opensso,dc=java,dc=net`
USERSTORE_MGRDN	The DN (distinguished name) of the directory manager, which is the user who has unrestricted access to the user data store. Default is `cn=Directory Manager`
USERSTORE_PASSWD	The password for the directory manager of the user data store.

The following table lists Site Configuration parameters with a description for each parameter.

Table 1–6 Configurator - Site Configuration Parameters

Parameter	Description
LB_SITE_NAME i	The name of the site.
LB_PRIMARY_URL	The load balancer URL. For example: `http://lb.example.com:58080/opensso`

1.4.4 To Configure Multiple OpenSSO STS Servers with Identical Configuration Settings

You can install multiple OpenSSO STS servers with the same configuration settings. This is useful in high availability deployments where the primary OpenSSO STS server and a backup OpenSSO STS server must have the same settings.

1. Create a configuration file following the instructions in Configuring OpenSSO STS Using the Command-Line Configurator.
2. Use the file to install the first OpenSSO STS server.
3. Make a copy of the file, and replace the OpenSSO STS server host name and port number with the server host name and port number the second OpenSSO STS server.
4. Use the edited file to run the Configurator again on the second host computer.

1.5 Configuring the OpenSSO STS Administrator Password

Log in to the OpenSSO STS administration console to configure the administrator passwords and policy agent passwords. See Chapter 3, "Getting Started Using the OpenSSO STS Console."

1.6 Installing the OpenSSO STS Command-Line Utility

The ssoadm command-line utility has two main purposes: to load configuration data into the data store, and to perform batch administrative tasks. For detailed command-line reference, see Chapter A, "Using the ssoadm Command-Line Interface."

The stsAdminTools.zip file is in the following directory:*zip-root*/tools

where zip-root is where you unzipped the openssosts_11gr1.zip file.

The following table describes the files after you unzip stsAdminTools.zip.

Table 1–7 ssoadmTools.zip Files

File or Directory	Description
README.setup	Description of the stsAdminTools.zip file.
license.txt	CDDL license agreement.
setup	Script to install the tools on Solaris and Linux systems.
setup.bat	Script to install the tools on Windows systems.
lib	JAR files required to run the scripts.
resources	Properties files required for the scripts for the various locales.

Table 1–7 (Cont.) ssoadmTools.zip Files

File or Directory	Description
template	Script templates for Solaris, Linux, and Windows systems.

1.6.1 To Install the ssoadm Command-Line Utility

1. Make sure the OpenSSO STS web container is running.
2. Make sure that your JAVA_HOME environment variable points to JDK 1.5 or later.
3. Create a new directory to unzip the stsAdminTools.zip file. For example: *tools-zip-root*.
4. Unzip the stsAdminTools.zip file in the new directory.
5. In the directory where you unzipped the stsAdminTools.zip file, run the setup script on Solaris and Linux systems or the setup.bat script on Windows. For example, on Solaris and Linux systems:

   # ./setup
6. When you are prompted, enter the path to the OpenSSO STS configuration, log, and debug directories. For example: /opensso

You can now run the OpenSSO STS CLI tools and utilities from the following directory:*tools-zip-root*/deploy_uri/bin

where:

- *tools-zip-root i*s the directory where you unzipped the stsAdminTools.zip file.
- `deploy_uri` is the name of the OpenSSO STS deploy URI. For example: opensso

See Appendix A, "Using the ssoadm Command-Line Interface"for detailed command information.

1.7 Uninstalling the OpenSSO STS Server

Before you begin. If the OpenSSO STS server instance was using the OpenSSO STS data store, the data store port was in use by the LISTEN socket. Stopping the web container server instance or domain should release this port. To check the data store port, use the `netstat` command. For example, if the OpenSSO data store used default port 50389:

```
netstat -a | grep 50389
```

Port 50389 should not be in use for the LISTEN socket. If necessary, release this port.

1.7.1 To Uninstall the OpenSSO STS Server

1. Undeploy opensso.war in the web container using the web container administration console or command-line utility.
2. Stop the OpenSSO STS web container.
3. Remove the following directories and all of their contents:
   - ConfigurationDirectory is the directory created when the OpenSSO STS instance is initially configured.

     The default directory is opensso in the home directory of the user running the Configurator. If the Configurator is run by root, ConfigurationDirectory is created in the root home directory (/).

- user-home-directory.openssocfg where *user-home-directory* is the home directory of the user who deployed the opensso.war file. If this user is root, the directory is /.openssocfg.

4. Optionally, remove the penssosts_11gr1.zip file and extracted files.

1.7.2 To Uninstall the OpenSSO STS Utilities and Scripts

1. Remove the directory and its contents where stsAdminTools.zip was extracted.
2. Optionally, remove the stsAdminTools.zip file.

Part II

Basic Server Administration

Part II contains the following chapters:

2

Overview of OpenSSO Security Token Service

Oracle OpenSSO Security Token Service (OpenSSO STS) provides a secure way to handle identity propagation that is controllable by policy. As a trusted authority service, OpenSSO STS issues and validates security tokens. As a web services security provider, OpenSSO STS secures communication between web service clients and the OpenSSO STS server itself. This chapter provides a high-level overview of how OpenSSO STS works, and what it can do for your enterprise.

The following topics are contained in this chapter:

2.1 About OpenSSO STS

In most enterprises, users are assigned to a single authentication mechanism which is mapped to a security role. But when a business requires applications and services that are not tied to a particular credential type or to a particular set of roles, then the claims-based security model work best.

OpenSSO STS is a service that assigns a claim, or security token, to an authenticated user. A security token is a more granular artifact than a role. Based on the assigned security token, the user can then be authorized to use a protected web application. In this way, Open SSO STS decouples applications and services from roles, and enables you to change the name and meaning of roles without affecting the system.

You can configure OpenSSO STS to act as a security token service, and as a web service security provider. When you configure OpenSSO STS as a security token service, applications can delegate authentication, user identity mapping, and user identity management to the OpenSSO STS authority.

When configured as a web service security provider, OpenSSO STS protects the security token service itself from unauthorized use or security breach. When you configure OpenSSO STS to act as a web service security provider, you must configure both the web service client and the web service provider.

2.1.1 The OpenSSO Security Token Service

The OpenSSO Security Token Service was developed from the WS-Trust protocol. The WS-Trust protocol defines extensions to the WS-Security specification for issuing and exchanging security tokens and establishing and accessing the presence of trust

relationships. The OpenSSO Security Token Service is hosted as a servlet endpoint and coordinates security-based interactions between a Web Service Client and a Web Service Provider. The OpenSSO Security Token Service is a stand-alone service that issues SAML tokens with inbound tokens that are compliant with the WS-I Basic Security Profile.

2.1.2 OpenSSO STS as a Web Service Security Provider

OpenSSO STS provides web service security support for client applications that are based on the Java API for XML Web Services (JAX-WS) or SOAP with Attachments API for Java (SAAJ). For JAX-WS based clients, web services security can be enforced at either the web or J2EE container level using container-provided security authentication and authorization plug-ins, or using JAX-WS Handlers. Handlers are interceptors that can be easily plugged into the Java API for XML-Based Web Services (JAX-WS) 2.0 runtime environment to do additional processing of inbound and outbound messages.

2.1.3 OpenSSO STS Agent Profiles

OpenSSO STS uses agent profiles to store configuration data for securing token requests sent to the Security Token Service. Agent profiles include web service client, OpenSSO STS client, and web service provider profiles.

OpenSSO STS client and web service client profiles can be used by client applications to secure communication between a client and OpenSSO STS or other web service provider. A web service provider configuration profile is used by both OpenSSO STS as well as a stand-alone web service provider.

2.2 Common Uses for OpenSSO STS

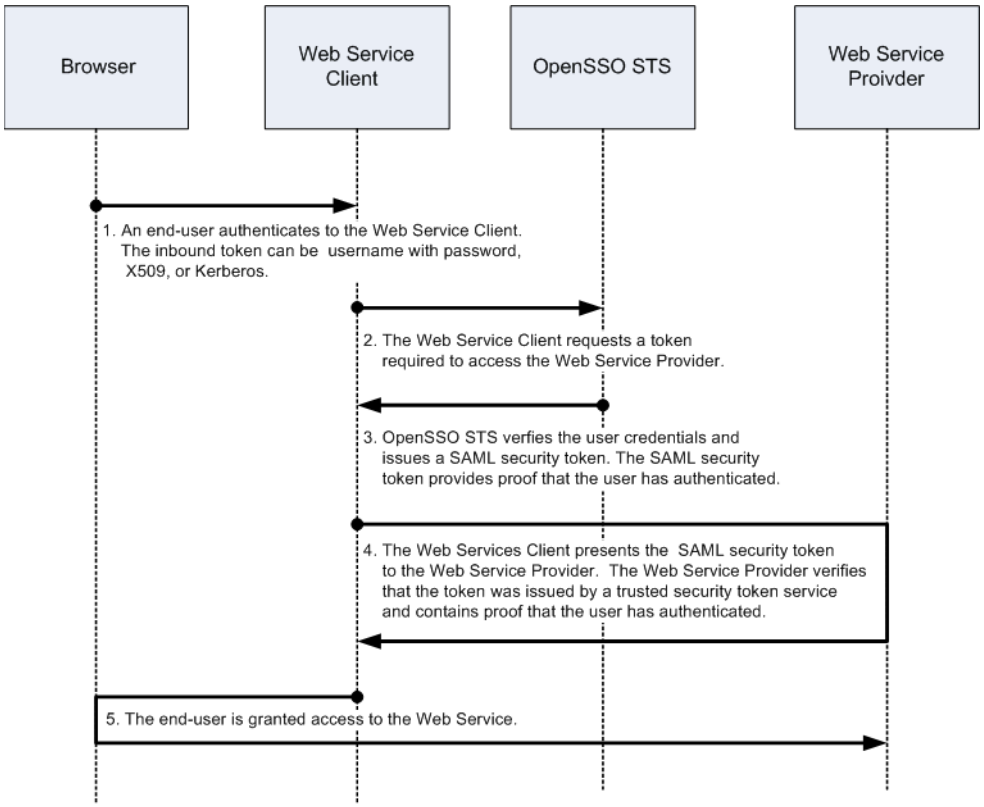
You can deploy OpenSSO STS as a stand-alone token security service, or as a component in a web service security solution.

2.2.1 Stand-Alone Security Token Service

OpenSSO STS leverages WS-Trust to manage token exchange between a web service client and a web service provider. The WS-Trust specification provides a standards-based way to send security token requests to any security token service. WS-Trust can be used to manage token transformation when crossing the various security boundaries of the information system.

The following figure illustrates how OpenSSO STS facilitates an interaction between a web service client and web service provider through brokered authentication.

Figure 2–1 Security Token Service Process Flow

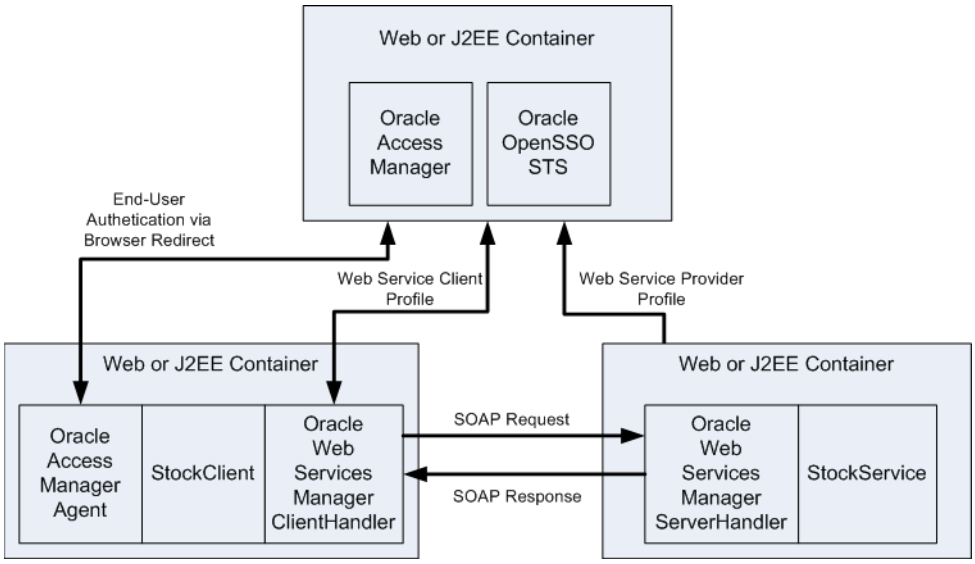


2.2.2 Web Services Security Provider

OpenSSO STS can be deployed to work with other Oracle components to provide comprehensive web services security. In web service deployments that require WS-Security, Oracle Access Manager is often used to address the authentication requirements in the environment. Additionally, Oracle Web Services Manager can provide a standards-compliant solution that enforces authorization and ensures confidentiality, privacy, and data integrity. The following example demonstrates how OpenSSO STS is used for identity propagation from a web service client to communicate with a web service provider.

An end-user logs in to an application named StockClient to configure a list of company stocks, and to periodically view current stock prices. In this example, StockClient is the web service client that communicates with the web service provider named StockService. OpenSSO STS propagates a user identity from the web service client. SOAP Messages are used to transfer the security tokens and to communicate between web services client and web service provider.

Figure 2–2 OpenSSO STS as a Web Service Security Provider



Access Manager handles the initial user authentication though a browser redirect by the Access Manager policy agent. Both StockClient and StockService are protected by the Web Services Manager policy agent that intercepts the request at the Web Service Provider, and the response at the Web Service Client. Web Services Manager then executes policies attached to each request and response in the transaction. Web Services Manager policy agents look up policy definition details in the Web Service Manager Policy Manager, and caches the policies to increase performance. Any changes to policy are dynamically updated by the Policy Manager. The Policy Manager propagates the changes to the policy agent which refreshes the policy cache and applies the changed policy immediately to the next request received.

If WS-Security is not a requirement, then Web Services Manager can be replaced with standard WS-Trust clients such as a WebLogic Server client. The WebLogic Server client communicates with OpenSTS on the web service client side, and uses a J2EE agent on the web service provider side.

2.3 Single-Realm Administration Console

After OpenSSO STS is deployed and configured, a single top-level realm is created. A realm is the administrative unit for OpenSSO STS. The Top Level Realm contains all configuration data for the OpenSSO STS instance except for bootstrapping information configured during installation. The Top Level Realm cannot contain subrealms. Information defined in the Top Level Realm includes:

- The location of one or more identity repositories containing users to whom the remaining configuration information applies.
- An authentication process that defines, among other information, the location of an authentication repository and the type of authentication required.
- Configuration data for the OpenSSO STS service itself. This includes configuration data for the OpenSSO STS authentication service, host server, global settings, and server and site management.

3

Getting Started Using the OpenSSO STS Console

OpenSSO STS provides a graphical user interface for centralized server and agent profile management. This chapter contains the following topics:

3.1 Logging In to the OpenSSO STS Console

To access the OpenSSO STS administration console, use a browser to go to the following URL:

`http://HostName.Domain: 8080/openssosts`.

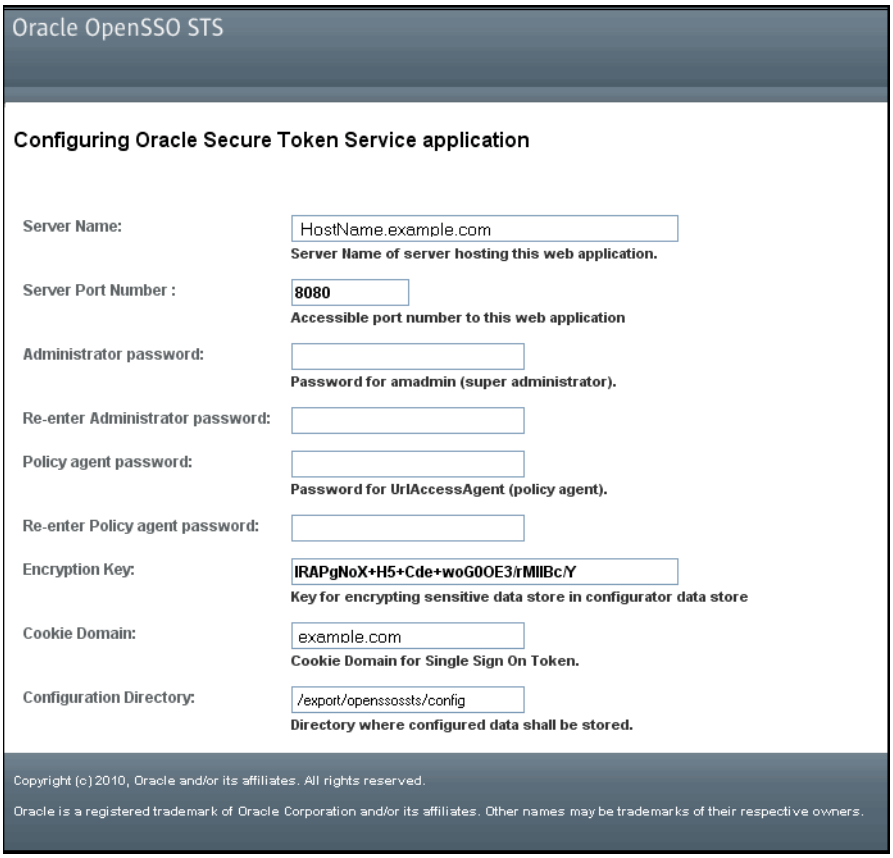
On the OpenSSO STS login page, enter the administrator username and password, and then click Login.

> **Important:** The first time you log into OpenSSO STS, you must use the default administrator username and password, and complete a one-time configuration task. See Section 3.2, "First-Time Login Configuration."

3.2 First-Time Login Configuration

After installing OpenSSO STS, when you access the OpenSSO STS server for the first time, the following page is displayed:

Figure 3–1 First-Time Login Configuration



Configuring the OpenSSO Security Token Application is a one-time only configuration task. See Section 3.2.1, "To Configure the OpenSSO STS Application."

3.2.1 To Configure the OpenSSO STS Application

1. On the Configuring the OpenSSO Security Token Service Application page, provide the following password values:

Table 3–1 OpenSSO STS Application Passwords

Password	Description
Administrator	This is the password administrators use to access OpenSSO STS.
Policy agent password	This password is used when Java EE agents are configured against OpenSSO STS.

   Other server information on this page is retrieved from the server itself. Click Configure when you're ready to proceed. When configuration is complete, a "Login to Console" link is displayed. Click the link open the administration console.

2. Click Configure.

   When configuration is complete, a link "Login to the Console" is displayed.

3. Click Login to the Console.

4. On the OpenSSO STS Login page, provide the following values:

Table 3–2 Default Administrator Login Values

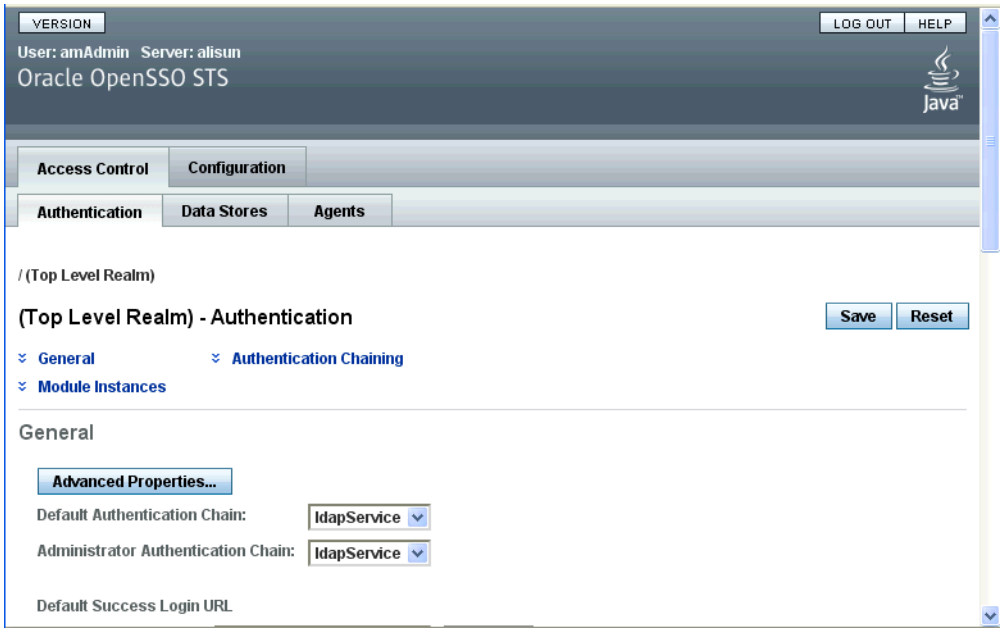
Parameter	Default Value
Username	amadmin
Password	Type the password specified in step 1.

5. Click Log In.

3.3 About the Single-Realm OpenSSO STS Console

In OpenSSO STS, a realm is an administration unit under which all service configuration data is stored. There is only one realm, the Top-Level Realm. You cannot add peer realms or subrealms. For detailed information about configuring OpenSSO STS, see Part II, "Basic Server Administration".

Figure 3–2 OpenSSO STS Administration Console



4

Managing the Security Token Service

This chapter provides instructions for determining your security token needs, and for configuring the Security Token Service to generate and validate security tokens to meet those needs.

The following topics are contained in this chapter:

4.1 About the OpenSSO Security Token Service

Oracle OpenSSO Security Token Service (OpenSSO STS) establishes a trust relationship between a web service client and a web service provider, and then brokers the trust between them. The web service can trust tokens issued by just one entity—OpenSSO STS— instead of having to communicate with several clients. In this way, OpenSSO STS significantly reduces trustpoint management overhead.

4.1.1 Security Token Generation Process Flow

An HTTP client, or browser, sends an access request through the web service client to the web service provider. A web services security agent redirects the request to the OpenSSO STS authentication service. A SOAP security agent issues the redirect.

The OpenSSO STS authentication service determines the security mechanism registered by the web service provider, and retrieves the appropriate security tokens. After successful authentication, the web service client provides a SOAP message body while the SOAP security agent on the web service client side inserts the security header and a token. The message is then signed before the request is sent to the WSP.

The SOAP security agent on the web service provider side verifies the signature and security token in the SOAP request before forwarding the request on to the web service provider itself. The web service provider then processes it and returns a response, signed by the SOAP security agent, back to the web service client. The SOAP security agent on the web service client side then verifies the signature before forwarding the response on to the web service client.

4.1.2 Supported Security Tokens and Security Mechanisms

The OpenSSO Security Token Service issues and authenticates the following security tokens:

Inbound Tokens

- UserName
- X.509
- Kerberos

Outbound Tokens

- UserName
- SAML 1.1 and SAML2.0
- Holder-of-Key, Bearer, and Sender Vouches

Token Translation

- Converts UserName to SAML.
- Converts OpenSSO STS SSOToken to SAML 1.1 or SAML 2.0 token.
- Converts SAML 1.1 to SAML 2.0 token, and SAML 2.0 to SAML1.1 token.
- Converts OpenSSOToken to SAML Assertion.
- Converts Oracle Access Manager SSOToken to SAML Assertion.
- Supports custom token plugability.

Additionally, end user tokens can be converted or validated after customization. In this case, the new token is an `OnBehalfOftoken` (a WS-Trust protocol element) carried in the WS-Trust request as part of the SOAP body. The new token becomes an authentication token carried as part of the SOAP header. Custom tokens can also be created and sent on behalf of an end user token for conversion or validation by the Security Token Service.

For detailed descriptions of security tokens, see the links to comprehensive Web Service Security specifications in Section 4.1.3, "Supported Standards."

Message-Level Security Bindings

- Asymmetric, Symmetric, and Transport-Layer security bindings
- WS-Security 1.0 and WS-Security 1.1 (for Kerberos)

For detailed descriptions of supported security tokens and security mechanisms, see the links to comprehensive Web Service Security specifications in Section 4.1.3, "Supported Standards."

4.1.3 Supported Standards

OpenSSO STS supports the following industry standards for web services security:

WS-Trust 1.3 and 1.0

Protocol for communicating requests and responses to the Security Token Service. See the following URLs for the comprehensive WS-Trust specification:

- `http://docs.oasis-open.org/ws-sx/ws-trust/200512`.

This specification describes the Web Services Trust Model and provides detailed information about security tokens and the Security Token Service Framework.

WS-Security 1.x
Standards that define processing rules for security when SOAP is used as a communication protocol for web services. See the following URL for the WS-Security specification:

http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

WS-Policy 1.5
XML-based language for expressing web service policy metadata through WSDL. See the following URL for the WS-Policy specification:

http://www.w3.org/TR/ws-policy/

This specification provides detailed information about the Policy Model and Policy Expression.

WS-SecurityPolicy 1.2
Security policy metadata expressed in WSDL. See the following URL for the comprehensive WS-Security Policy specification:

http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.html

This specification defines a set of security policy assertions for use with the WS-Policy framework. The specification includes detailed information about security token properties and assertions, security bindings, and SOAP message security options.

4.1.4 Leveraging Dynamic Policy For OpenSSO STS WSDL

Any changes you make to the Security Token Service configuration are automatically and dynamically reflected in the OpenSSO STS WSDL. For example, if you want to disable a security mechanism, you can use the OpenSSO STS console to make the change once, and that change is propagated throughout the Security Token Service. You do not have to make additional manual changes or redeploy OpenSSO STS. You can inspect the corresponding WSDL to be sure that the change has been made appropriately. Go to the WSDL URL. Example:

`http://`*`OpenSSOhost:OpenSSOport`*`/`*`deployURI`*`/sts?wsdl`

4.2 To Configure the Security Token Service

When you configure the Security Token Service, you specify how OpenSSO STS authenticates remote clients that request security tokens.

1. On the Configuration tab, click the Global subtab.
2. In the Global Properties list, click Security Token Service.
3. On the Security Token Service page, provide values for the Token Issuance Attributes, Security, Signing and Encryption, Key Store, Kerberos Configuration, and SAML Configuration attributes.

   The following table provides a listing and descriptions of the attributes you can configure.

Table 4–1 Security Token Service Attributes

Attribute	Description
Token Issuance Attributes	

Table 4–1 (Cont.) Security Token Service Attributes

Attribute	Description
Issuer	Specify the name of the Security Token service that issues the security tokens.
End Point	The OpenSSO STS end point that a client communicates with. Use the form: `HTTP://OpenSSOhost:OpenSSOport/URI/sts` If the OpenSSO STS is fronted by a load-balancer, then use the load balancer host name and port number. This syntax allows for dynamic substitution of the Security Token Service Endpoint URL based on the specific session parameters.
SSL End Point	The SSL-enabled OpenSSO STS end point that a client communicates with. Use the form: `HTTPS://host:port/URI/sts` If the OpenSSO STS is fronted by a load-balancer, then use the load balancer host name and port number. This syntax allows for dynamic substitution of the Security Token Service Endpoint URL based on the specific session parameters.
Lifetime for Security Token	Specify the number of milliseconds the issued token is valid.
Certificate Alias Name	Specify the alias name for the certificate or key used to sign the security token issued by the Security Token service.
STS End User Token Plugin class	Specify the implementation class for the end user token conversion.
Security	
Security Mechanism	Specify the type of security credential that is used to secure either the security token itself, or the security credential accepted by the Security Token service from the incoming WS-Trust request sent the by the client. Choose from the following security types: ■ UserNameToken-Plain — Uses a user name token with a clear text password for securing Security Token service requests. ■ X.509Token — Uses the X.509 certificate to secure the Security token. ■ TLS-UserNameToken-Plain — ■ KerberosToken — Uses Kerberos tokens.
Authentication Chain	Choose the authentication chain or service name to be used to authenticate to the OpenSSO STS authentication service. OpenSSO STS uses the credentials from an incoming issuer request's security token to generate an OpenSSO STS-authenticated security token.

Table 4–1 (Cont.) Security Token Service Attributes

Attribute	Description
Credential for User Token	This attribute is not used when an authentication chain is configured. The list displays the username and password shared secrets to be used by the Security Token service to validate a UserName token. The UserName token is sent by the client as part of the incoming WS-Trust request. ■ To add a credential user token to the list, Click Add. In the Add User Credential page, type a username and password, and then click Add. ■ To remove an entry from the Credential for User Token list, click to mark the box corresponding to the entry, and then click Remove.
On Behalf Of Token	This attribute represents the WS-Trust protocol OnBehalfOf element. These elements are used for token translations. Choose one or more of the following: OpenSSO UserName UserName with Password SAML 1 SAML 2 Custom
Authentication Chain for On Behalf Of Token	Choose the authentication chain or service to be used to authenticate to the OpenSSO STS Authentication service.
Detect Message Replay	This attribute is used to detect message replays from a client in a malicious attack. When enabled, Yes is checkmarked and replay are automatically detected.
Detect User Token Replay	This attribute is used to detect message replays from a client in a malicious attack when the security mechanism is the Username token. When enabled, Yes is checkmarked, and
Signing and Encryption	
Is Request Signature Verified	When enabled, Yes is checkmarked, and the Security Token service must verify the signature of the incoming WS-Trust request.
Disable signature validation when transport is secured with SSL	When enabled, Yes is checkmarked, and OpenSSO STS does not verify the signature if the Transport Layer security binding is used.
Is Response Signed Enabled	When enabled, Yes is checkmarked. Specify the responses received by the Security Token Service that must be signed. Choose from the following: Body SecurityToken Timestamp To From ReplyTo Action MessageID

Table 4–1 (Cont.) Security Token Service Attributes

Attribute	Description
Signing Reference Type	Specify how to detect the security token used to sign the SOAP response. ■ Direct Reference ■ Key IdentifierRef ■ X.509 Issuer Serial Reference For detailed information see http://www.oasis-open.org/committees/download.php/5943/oasis-200401-wss-x509-token-profile-1.0.pdf.
Is Request Decrypted	When enabled, Yes is checkmarked, OpenSSO STS decrypts the incoming request message. Body: When checkmarked, the body is decrypted. Header: When checkmarked, the header is decrypted.
Disable decryption when transport is secured with SSL	When enabled, Yes is checkmarked, and OpenSSO STS does not decrypt the incoming request message when transport layer security binding is used.
Is Response Encrypted	When enabled, Yes is checkmarked, and all responses sent by the Security Token service must be encrypted.
Encryption Algorithm	Choose the encryption algorithm used by the Security Token service to encrypt the WS-Trust response. Choose from the following: ■ AES ■ 3DES Depending upon the encryption algorithm you use, choose the corresponding Encryption Strength (below).
Encryption Strength	Choose the encryption strength to be used by the Security Token service to encrypt the WS-Trust response. A high value corresponds to a high encryption strength level. Choose a value based on the algorithm specified in the Encryption Algorithm field. ■ For AES, values of 28, 192, and 256 are supported. ■ For 3DES, values of 0, 112, and 168 are supported. Depending upon the Encryption Strength, you may have to provision the appropriate encryption policies for the JDK.
Key Store	
Private Key Alias	Specify the private certificate key alias to be used to sign the WS-Trust response or to decrypt the incoming WS-Trust request.
Private Key Type	Specify the certificate private key type to be used for signing WS-Trust responses or decrypting WS-Trust requests. Choose from the following: PublicKey, SymmetricKey, or NoProofKey. ■ PublicKey ■ SymmetricKey ■ NoProofKey
Public Key Alias of Web Service (WS-Trust) Client	Specify the public certificate key alias to be used to verify the signature of the incoming WS-Trust request or to encrypt the WS-Trust response.
Kerberos Configuration	

Table 4–1 (Cont.) Security Token Service Attributes

Attribute	Description
Kerberos Domain Server	Specify the Kerberos Distribution Center (the domain controller) hostname. Use the fully qualified domain name of the domain controller.
Kerberos Domain	Specify the Kerberos Distribution Center (domain controller) domain name. Depending up on your configuration, the domain name of the domain controller may be different than the OpenSSO STS domain name.
Kerberos Service Principal	Specify the Kerberos principal designated as the owner of the generated Security token. Use the following format: `HTTP/hostname.domainname@dc_domain_name` where *hostname* and *domainame* represent the host name and domain name of the OpenSSO STS instance, and *dc_domain_name* is the Kerberos domain in which the Windows Kerberos server (domain controller) resides. The Kerberos server may be different from the domain name of the OpenSSO STS instance.
Kerberos Key Tab File	Specify the Kerberos keytab file to be used for issuing the token. Use the following format, although the format is not required: *hostname*`.HTTP.keytab` where *hostname* is the host name of the OpenSSO STS instance.
Is Verify Kerberos Signature	When enabled, Yes is checkmarked, and OpenSSO STS verifies the incoming request signature using the Kerberos token. This is optional and must be enabled only when JDK6 is used.
SAML Configuration	
SAML Attribute Mapping	The Current Values list displays the SAML attribute that must be generated as an Attribute Statement when the Security Token Service creates a SAML assertion for a web service provider. Use the following format: ■ To add a SAML attribute mapping to the list, in the New Value field type an attribute value-pair, and then click Add. Use the following format: *SAML_attr_name=Real_attr_name* where *SAML_attr_name* is the SAML attribute name from a SAML assertion contained in an incoming web service request, and *Real_attr_name* is the attribute name that is obtained from either the authenticated SSOToken or the identity repository. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
NameID Mapper	Specify the SAML NameID Mapper to be used in an assertion that is generated for the Security Token service.
Should Include Memberships	When enabled, Yes is checkmarked, and the generated assertion contains user memberships as SAML attributes.
Attribute Namespace	Specify the SAML Attribute Namespace for an assertion that is generated for the Security Token service.

Table 4–1 (Cont.) Security Token Service Attributes

Attribute	Description
Trusted Issuers	The Current Values list displays trusted issuers that can be trusted to send security tokens to OpenSSO STS. OpenSSO STS must verify that a security token was sent from one of these issuers. ■ To add a trusted issuer to the list, in the New Value field type the trusted issuer name, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Trusted IP Addresses	The Current Values list displays IP addresses that can be trusted to send security tokens to OpenSSO STS. OpenSSO STS must verify whether a security token was sent from one of these hosts. ■ To add an IP address to the list, in the New Value field type an IP address, and then click Add. ■ To remove an entry from the Current Values list, select the entry and then click Remove.

4. Click Save.

4.3 Generating Security Tokens

The following summarizes the steps you take to use OpenSSO STS to generate security tokens:

1. Register a web service provider.

   See Section 4.4, "To Register a Web Service Provider to OpenSSO STS.".

2. Configure the web service provider. See the following sections:
   - Section 4.3.1, "Using the Security Token Generation Matrix."
   - Section 4.3.2, "To Read the Security Token Generation Matrix."
   - Section 4.5, "To Configure a Web Service Provider."

4.3.1 Using the Security Token Generation Matrix

Use the Security Token Generation Matrix to help you configure OpenSSO STS to generate a web service client security token required by the web service provider.

4.3.1.1 Token Generation Matrix Legend

Message-Level Security Binding

- Transport binding provides message protection only at the transport layer level.
- Asymmetric binding uses public key cryptography to provide message protection at the SOAP message encoding layer.
- Symmetric binding uses Kerberos tokens to provide message protection at the SOAP message encoding layer.

Web Service Client Token

- Username

See
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

- X.509

  See
  http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-x509TokenProfile.pdf

- Kerberos

  See
  http://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf

KeyType

- WS-Security SAML tokens: Bearer, Sender-Vouches, and Holder-of-Key
- WS-Trust RST keytype values: Symmetric, Public, and Bearer

OnBehalfOf Token
Used when the requestor is obtaining a token on behalf of another party.

UseKey
Used when the client supplies a public-key to be embedded in the issued token as the proof key.

OpenSSO STS Output Token
Attributes to be included in the SOAP message.

For general information about Web Service Security and related terminology, see the following URLs:

- http://www.oracle.com/technology/tech/standards/pdf/security.pdf
- http://download.oracle.com/docs/cd/E15523_01/web.1111/b32511/intro_security.htm#CDDHHGEE

4.3.2 To Read the Security Token Generation Matrix

The Security Token Generation Matrix summarizes frequently-used Security Token Service parameter settings and the types of security tokens OpenSSO STS generates based on these settings. An example follows the table.

Table 4–2 Security Token Generation Matrix

Row	Message-Level Security Binding	Web Service Client Token	KeyType	OnBehalfOf Token	Use Key	OpenSSO STS Output Token
1	Asymmetric	X.509	Bearer	Yes	No	SAML Bearer, no proof key
2	Asymmetric	Username	Bearer	Yes	No	SAML Bearer, no proof key
3	Asymmetric	X.509	Bearer	No	No	SAML Bearer, no proof key
4	Asymmetric	Username	Bearer	No	No	SAML Bearer, no proof key

Table 4–2 (Cont.) Security Token Generation Matrix

Row	Message-Level Security Binding	Web Service Client Token	KeyType	OnBehalfOf Token	Use Key	OpenSSO STS Output Token
5	Asymmetric	X.509	Symmetric	Yes	No	SAML Holder-of-Key, Symmetric proof key
6	Asymmetric	Username	Symmetric	Yes	No	SAML Holder-of-Key, Symmetric proof key
7	Asymmetric	X.509	Symmetric	No	No	SAML Holder-of-Key, Symmetric proof key
8	Asymmetric	Username	Symmetric	No	No	SAML Holder-of-Key, Symmetric proof key
9	Asymmetric	X.509	Asymmetric	No	Web Service Client public key	SAML Holder-of-Key, Asymmetric proof key
10	Asymmetric	X.509	Oracle-proprietary for SAML sender-vouches	Yes	No	SAML sender-vouches, no proof key
11	Asymmetric	Username	Oracle-proprietary for SAML sender-vouches	Yes	No	SAML sender-vouches, no proof key
12	Transport	Username	Bearer	Yes	No	SAML Bearer, no proof key
13	Transport	Username	Bearer	No	No	SAML Bearer, no proof key
14	Transport	Username	Symmetric	Yes	No	SAML Holder-of-Key, Symmetric proof key
15	Transport	Username	Symmetric	No	No	SAML Holder-of-Key, Symmetric proof key
16	Transport	Username	Oracle-proprietary for SAML sender-vouches	Yes	No	SAML sender-vouches, no proof key
17	Asymmetric	X.509	Asymmetric	No	No	SAML Holder-of-Key, Asymmetric proof key

Table 4–2 (Cont.) Security Token Generation Matrix

Row	Message-Level Security Binding	Web Service Client Token	KeyType	OnBehalfOf Token	Use Key	OpenSSO STS Output Token
18	Asymmetric	X.509	No	No	No	SAML Holder-of-Key, Asymmetric proof key
19	Asymmetric	Username	No	No	No	SAML Holder-of-Key, Symmetric proof key
20	Transport	Username	No	No	No	SAML Holder-of-Key, Symmetric proof key

4.3.2.1 Example: Using the Token Generation Matrix

In the last column titled OpenSSO STS Output Token, find a description that meets the web service provider token requirements. Then make note of the parameter values in the same row, and use those values when you configure the Web Service Provider profile.

For example, after installing OpenSSO STS, you may want to add a web service provider. You must gather some information before you can configure the Security Token Service to generate the required tokens. First, determine how a client should authenticate to OpenSSO STS: using X.509, Kerberos, or Username tokens. See Web Service Client Token. For this example choose X.509.

Next determine what type of security token the web service requires. You can inspect the web service provider security policy which defines the web service provider requirements. For example, the security policy may indicate that the web service provider will accept a SAML bearer token using Asymmetric binding.

Now you can use the OpenSSO STS console to add and register a web service provider, and import the web service provider certificate alias. Make note of the certificate alias name. For this example, use CertAliasTest.

This is where you read the Security Token Generation Matrix. Based on the information you've gathered so far, look in the last column of the matrix for SAML bearer token. Now exclude any rows that use Web Service Client Tokens other than X.509. Row 1 meets your needs so far, and you can use the remaining values in Row 1 as guidelines for configuring the web service provider, making adjustments as necessary.

Now you are ready to configure the web service provider. Following the instructions in Section 4.5, "To Configure a Web Service Provider," you can provide the following values:

Parameter	Value
Security Binding	Asymmetric
Web Service Client Token	X.509
KeyType	Bearer
Certificate Alias Name	CertAliasTest

4.4 To Register a Web Service Provider to OpenSSO STS

When you add a new web service provider security agent profile, the web service provider is automatically registered to OpenSSO STS.

1. On the Access Control tab, click the Agents subtab.
2. In the Agent section, click New.
3. In the New Agent page, provide the following information:

Property	Description
Name	Specify a name for the new web service provider.
Password	Specify an administrator password for accessing the new web service.
Re-Enter Password	Re-enter the password to confirm it.

4. Click Create.

Once you've registered a web service provider to OpenSSO STS, you can configure the web service provider. See Section 4.5, "To Configure a Web Service Provider."

4.5 To Configure a Web Service Provider

1. On the Access Control tab, click the Agents subtab.
2. In the Agent section, click name of the web service provider you want to configure.
3. Provide the following General, End Points, Key Store,Security, and SAML Configuration information:

Property	Description
General	
Group	Select a group to which this web service provider belongs.
End Points	
Web Service End Point	This is the end point that the web service is listening to. Specify a URL for the service end point.
Key Store	
Public Key Alias	Specify the public key alias used for encrypting shared secrets or any other SAML assertion intended for the web service provider.
Security	
Security Mechanism	Choose one of the following: SAML-Bearer SAML-Holder-of-Key SAML-SenderVouches SAML2-Bearer SAML2-Holder-of-Key SAML2-SenderVouches See http://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTokenProfile.pdf for detailed information about SAML security tokens.

Property	Description
SAML Configuration	
SAML Attribute Mapping	This configuration represents a SAML attribute that needs to be generated as an Attribute Statement during SAML assertion creation by the Security Token Service for a web service provider. The format is *SAML_attr_name=Real_attr_name*. *SAML_attr_name* is the SAML attribute name from a SAML assertion from an incoming web service request. *Real_attr_name* is the attribute name that is fetched from either the authenticated SSOToken or the identity repository.
SAML NameID Mapper Plugin	Specify the NameID mapper plug-in class that is used for SAML account mapping.
SAML Attributes Namespace	Specify the name space used for generating SAML attributes.
Include Memberships	If enabled, this attribute specifies that the principal's membership must be included as a SAML attribute.
Generate attribute assertion only:	If enabled, the generated SAML assertion for a web service provider will contain only SAML attribute statements, and no authentication statements.

4. Click Save.

   Once you've registered a web service provider to OpenSSO STS, you can configure a WS-Trust client. See Section 4.6, "To Register a WS-Trust Client."

4.6 To Register a WS-Trust Client

When import a X.509 client certificate into the OpenSSO STS keystore, the WS-Trust client is automatically registered with OpenSSO STS.

1. Obtain a client X.509 certificate or public key in ASCII format from the web service client.
2. Use the keytool command to import the certificate or public key into the OpenSSO STS keystore.

```
cd $OPENSSO_CONFIG/openssosts/openssosts
keytool -import -keystore keystore.jks -alias <pkaliasname> -file <pkfilename>
```

3. When prompted, enter the default keystore password.

   The default keystore password is changeit.

4. (Optional) Add the certificate into the Advanced Properties list.

   See Section 5.1.3.6, "To Configure OpenSSO STS Server Advanced Properties."

5. (Optional) You can use this certificate to configure a Certificate-based authentication module in the authentication chain.

   See Section 6.3, "Managing Authentication Module Instances" and Section 6.4, "Managing Authentication Chains".

5

Configuring OpenSSO STS System Properties

When you first install the Oracle OpenSSO Security Token Service (OpenSSO STS) server, by default the server is configured to secure all communication between the web service client and the OpenSSO STS. No entity can access the Security Token Service or the server itself until you configure the OpenSSO STS system properties. OpenSSO STS system properties define user access criteria, and also specify the various security mechanisms and other processes OpenSSO STS uses. The following topics are contained in this chapter:

5.1 Managing OpenSSO STS Servers

Whenever you install an OpenSSO STS server, you must edit the default server settings to suit your enterprise needs. When you install multiple servers, you must configure the servers to communicate with each other and to function as a single site or cluster.

5.1.1 To Edit the Default OpenSSO STS Server Settings

1. On the Configuration tab, click the Servers and Sites Subtab.
2. Click the Default Server Settings.

   On the Edit *server-default* page, the Advanced Properties section lists all properties and default values that apply to the default OpenSSO STS server.

   - To add a new property, click Add.

     A new row is added to the bottom of the list. In the appropriate columns, type a Property Name and Property Value.

   - To delete a property from the Advanced Properties list, click to check the box corresponding to the property, and then click Delete.
3. Click Save.
4. Click "Back to Servers and Sites."

5.1.2 To Add a New OpenSTS Server

1. On the Configuration tab, click the Servers and Sites subtab.
2. Click New.
3. Configure the OpenSSO STS server. See Section 5.1.3, "To Configure an OpenSSO STS Server."

5.1.3 To Configure an OpenSSO STS Server

1. On the Configuration tab, click the Servers and Sites subtab.

   The Servers list displays the Server Name and Site Name of
2. Click the name URL of the server you want to configure.
3. Click the General tab to configure centralized server management properties.

   See Section 5.1.3.1, "To Configure OpenSSO STS Server General Properties."
4. Click the Security tab to configure encryption, validation, and cookie properties that control the level of security for the OpenSSO STS server.

   See Section 5.1.3.2, "To Configure OpenSSO STS Server Security Properties."
5. Click the Session tab to configure OpenSSO STS server sessions.

   See Section 5.1.3.3, "To Configure OpenSSO STS Server Session Properties."
6. Click the SDK tab to configure the back-end data store settings.

   See Section 5.1.3.4, "To Configure OpenSSO STS Server SDK Properties."
7. Click the Directory Configuration tab to edit the embedded Directory Server settings.

   See Section 5.1.3.5, "To Configure OpenSSO STS Server Directory Configuration Properties."
8. Click the Advanced tab to select and add values to server properties that are not present in the OpenSSO STS Console.

   See Section 5.1.3.6, "To Configure OpenSSO STS Server Advanced Properties."
9. Click "Back to Servers and Sites."

5.1.3.1 To Configure OpenSSO STS Server General Properties

1. On the Configuration tab, click the Servers and Sites tab.
2. In the Servers section, click the URL of the OpenSSO STS server you want to configure.
3. Click the General tab.

   Provide values for Site, System, Debugging, and Mail Server properties.

   The following table provides a listing and descriptions of the properties you can configure.

Table 5–1 OpenSSO STS Server General Properties

Property	Description
Site	

Table 5–1 (Cont.) OpenSSO STS Server General Properties

Property	Description
Parent Site	Choose the load balancer Site Name (site ID) that maps to the OpenSSO STS server. The site must already exist before you can add the site.
System	
Base installation directory	Specify the base directory where product data resides. This information is specified in the property `com.iplanet.services.configpath`.
Default Locale	Specify the default language subtype that OpenSSO STS was installed with. The default is en_us and is specified in the property `com.iplanet.am.locale`.
Notification URL	Specify the location of the Notification service end point. This value is usually the product deployment and uses the form URI/notificationservice. This information is specified in the property `com.sun.identity.client.notification.url`.
XML Validation	When enabled, this property is set to On, and validation is required when parsing XML documents. This information is set in the property `com.iplanet.am.util.xml.validating`.
Debugging	
Debug Level	Specify a debug level for all components in the product. Choose one of the following levels: Off - No debug information is recorded. Error - Used for production. During production, there should be no errors in the debug files. Warning - Enables Error and Warning debug messages to be written. Message - Enables detailed code tracing. **Note:** Warning and Message levels should not be used in production. They cause severe performance degradation and an abundance of debug messages. This value is set in the property `com.iplanet.services.debug.level`.
Merge Debug Files	When enabled, this property is set to On, and all debug data is directed to a single file named debug.out. When disabled, this property is set to Off, and OpenSSO STS creates a separate component debug file per component. This value is set in the property `com.sun.services.debug.mergeall`.
Debug Directory	Specify the directory where debug files reside. Use the form *BASE_DIR/SERVER_URI*/`debug` This value is set in the property `com.iplanet.services.debug.directory`.
Mail Server	
Mail Server Host Name	Specify the mail server h ost name to use for sending email notifications. Example: `localhost` This value is set in the property `com.iplanet.am.smtphost`.
Mail Server Port Number	Specify the mail server port number. The default is 25. This value is set in the property `com.iplanet.am.smtpport`.

4. Click Save.
5. (Optional) Click Inheritance Settings.

   The Inheritance Settings section lists server properties containing default values. A checked box indicates a property that can inherit default server properties. Checked properties will be overwritten for each server instance.

   - To select a property to be overwritten with default values, click its corresponding box until a check appears in the box.
   - To deselect a property and retain any custom configuration, click the property's corresponding box until the box contains no check mark.
6. (Optional) Click Export Configuration.

   The OpenSSO STS console displays your settings so that you can inspect the settings for accuracy.
7. Click Save.
8. Click "Back to Server Profile."
9. Click "Back to Servers and Sites."

5.1.3.2 To Configure OpenSSO STS Server Security Properties

1. On the Configuration tab, click the Servers and Sites tab.
2. In the Servers section, click the URL of the OpenSSO STS server you want to configure.
3. Click the Security tab.
4. Provide values for Encryption, Validation, Cookie, Key Store, Certificate Revocation List Caching, Online Certificate Status Protocol Check, and Federal Information Processing Standards properties.

   The following table provides a listing and descriptions of the properties you can configure.

Table 5–2 OpenSSO STS Server Security Properties

Property	Description
Encryption	
Password Encryption Key	Specify the key to be used to encrypt and decrypt passwords. This key is stored in the Service Management System configuration and its value is set during installation. Example: `dSB9LkwPCSoXfIKHVMhIt3bKgibtsggd` This value is set in the property `am.encryption.pwd`.
Authentication Service Shared Secret	Specify the shared secret for the application authentication module. Value is set during installation. Example: `AQICPX9e1cxSxB2RSy1WG1+O4msWpt/6djZl` This value is set in the property `com.iplanet.am.service.secret`.

Table 5–2 (Cont.) OpenSSO STS Server Security Properties

Property	Description
Encryption class	Specifies the encrypting class implementation. Available classes are: `com.iplanet.services.util.JCEEncryption` and `com.iplanet.services.util.JSSEncryption.` The default value is `com.iplanet.services.util.JCEEncryption`. This value is set in the property `com.iplanet.services.util.JCEEncryption`.
Secure Random Factory Class	Specifies the factory class name for `SecureRandomFactory.` Available implementation classes are: `com.iplanet.am.util.JSSSecureRandomFactoryImpl`, which uses JSS, and `com.iplanet.am.util.SecureRandomFactoryImpl` which uses pure Java. `HttpRequest` default value is `com.iplanet.am.util.JSSSecureRandomFactoryImpl.` This value is set in the property `com.iplanet.security.SecureRandomFactoryImpl.`
Validation	
Platform Low Level Comm. Max. Content Length	Specifies the maximum number of bytes allowable for content in an `HttpRequest` that OpenSSO STS will accept. The default value is 1638. This value is set in the property `com.iplanet.services.comm.server.pllrequest.maxContentLength`
Client IP Address Check	When enabled, the property is set to Yes, and the IP address of the client is checkmarked in all single sign-on token creations or validations. The default value is No. This value is set in the property `com.iplanet.am.clientIP.`
Cookie	
Cookie Name	Specifies the Cookie name to be used by the Authentication service to set the valid session handler ID. The value of this cookie name is used to retrieve the valid session information. The default value is iPlanetDirectoryPro. This value is set in the property `com.iplanet.am.cookie.name.`
Secure Cookie	When enabled, this property is set to Yes, and the cookie is set in a secure mode. In secure mode, when a secure protocol such as HTTPS is used, the browser will return only the cookie. The default is No. This value is set in the property `com.iplanet.am.cookie.secure`.
Encode Cookie Value	When enabled, this property is set to Yes, and OpenSSO STS URL-encodes the cookie value which converts characters so they are understandable by HTTP. The default value is No. This value is set in the `property com.iplanet.am.cookie.encode`.
Key Store	

Table 5–2 (Cont.) OpenSSO STS Server Security Properties

Property	Description
Keystore File	Specifies the path to the SAML XML keystore password file. Example: *OpenSSO-deploy-base/URI*/`keystore.jks`. This value is set during installation in the property propertycom.sun.identity.saml.xmlsig.keystore. Example: *OpenSSO-deploy-base/URI*/`keystore.jks`.
Keystore Password File	Specifies the path to the SAML XML key storepass file. Example: *OpenSSO-deply-base/URI*/`.storepass`. This value is set during installation in the property `com.sun.identity.saml.xmlsig.storepass`.
Private Key Password File	Specifies the path to the SAML XML key password file. Example: *OpenSSO-deploy-base/URI*/`.keypass` The key password file contains the password that protects the private key of a generated key pair. This value is set during installation in the property `com.sun.identity.saml.xmlsig.keypass`.
Certificate Alias	This is the private key alias that is used to sign SOAP responses. Default value is test. This value is set in the property `com.sun.identity.saml.xmlsig.certalias`.
Certificate Revocation List Caching	
LDAP server port number:	Specifies the port number of the LDAP server where the certificates are stored. The default value is the port specified when OpenSSO STS was installed. You can use port number of any LDAP Server where the certificates are stored.
SSL/TLS Enabled	When enabled, the value is set to Yes, and the Certificate authentication service uses SSL to access the LDAP server. The default value is No.
LDAP server bind user name	Specifies the bind DN in the LDAP server.
LDAP server bind password	Specifies the password for the bind DN. By default, the amldapuser password that was specified during installation is used as the bind user.
LDAP search base DN	Specifies the base DN used by the LDAP Users subject in the LDAP server from which to begin the search. By default, the value is the top-level realm of the OpenSSO STS installation base.
Search Attributes	Specifies any DN component of the issuer's subjectDN to be used to retrieve a CRL from a local LDAP server. All Root CAs must use the same search attribute.
Online Certificate Status Protocol Check	
Check Enabled	When enabled, the value is set at Yes, and OCSP checking occurs. The default value is No.

Table 5–2 (Cont.) OpenSSO STS Server Security Properties

Property	Description
Responder URL:	Specifies a URL that identifies the location of the OCSP responder. Example: `http://ocsp.example.net:80`. By default, the location of the OCSP responder is determined implicitly from the certificate being validated. This property is used when the Authority Information Access extension defined in RFC 3280 is absent from the certificate, or when the Authority Information Access extension must be overridden.
Certificate Nickname	Specifies the CA certificate nick name for the OCSP responder. Example: Certificate Manager - MyCompany. ■ If set, then the CA certificate must be presented in the web server's certificate database. ■ If the OCSP URL is set, the OCSP responder nickname must be set also. Otherwise, both will be ignored. ■ If they are not set, the OCSP responder URL presented in user's certificate will be used for OCSP validation. If the OCSP responder URL is not presented in user's certificate, no OCSP validation will be performed. ■ If the OCSP responder URL is not presented in user's certificate, no OCSP validation will be performed.
Federal Information Processing Standards	
FIPS Mode:	When enabled, this value is set to True, and all cryptography operations will run in FIPS-compliant mode. Federal Information Processing Standards Under the Information Technology Management Reform Act (Public Law 104-106), the Secretary of Commerce approves standards and guidelines that are developed by the National Institute of Standards and Technology (NIST) for Federal computer systems. These standards and guidelines are issued by NIST as Federal Information Processing Standards (FIPS) for use government-wide. NIST develops FIPS when there are compelling Federal government requirements such as for security and interoperability and there are no acceptable industry standards or solutions.

5. Click Save.
6. (Optional) Click Inheritance Settings.

   The Inheritance Settings section lists server properties containing default values. A checked box indicates a property that can inherit default server properties. Unchecked properties can be overwritten for each server instance.

   - To select a property to be overwritten with default values, click its corresponding box until a check appears in the box.
   - To deselect a property and retain any custom configuration, click the property's corresponding box until the box contains no checkmark.
7. (Optional) Click Export Configuration.

   The OpenSSO STS console displays your settings so that you can inspect the settings for accuracy.
8. Click Save.
9. Click "Back to Server Profile."

10. Click "Back to Servers and Sites."

5.1.3.3 To Configure OpenSSO STS Server Session Properties

1. On the Configuration tab, click the Servers and Sites tab.
2. In the Servers section, click the URL of the OpenSSO STS server you want to configure.
3. Click the Session tab.
4. Provide values for Session Limits, Statistics, Notification, and Validation properties.

   The following table provides a listing and descriptions of the properties you can configure.

Table 5–3 OpenSSO STS Server Session Properties

Property	Description
Session Limits	
Maximum Sessions	Specifies the maximum number of concurrent sessions allowed. This value is set in the property `com.iplanet.am.session.maxSessions`.
Invalidate Session Max Time	Specifies the number of minutes after which an invalid session will be removed from the session table when a session created but the user does not login. Use a value greater than the timeout value set in the Authentication module properties file. The Invalidate Session Max Time value is set in the property `com.iplanet.am.session.invalidsessionmaxtime`.
Sessions Purge Delay	Specifies the number of minutes to delay to purge sessions. This value is set in the property `com.iplanet.am.session.purgedelay`.
Statistics	
Logging Interval	Specifies the number of seconds to elapse between statistics logging. The interval should be at least 5 seconds to avoid CPU saturation. An interval value less than 5 seconds will be interpreted as 5 seconds. This value is set in the property `com.iplanet.am.stats.interval`.
State	Specifies the location of the statistics log. The following are possible settings: ■ off - No statistics are logged. ■ file - Statistics are written to a file under the specified directory. ■ console - Statistics are written into Web Server log files. This value is set in the property `propertycom.iplanet.services.stats.state`.
Directory	Specifies the directory where the statistic files will be created. Example: OpenSSO *STS-base/server-URI*`/stats` Uses forward slashes "/" to separate directories. Spaces in the file name are allowed on only the Windows platform. This value is set in the property `com.iplanet.services.stats.directory`.

Table 5–3 (Cont.) OpenSSO STS Server Session Properties

Property	Description
Enable Host Lookup	When enabled, this value is set to Yes, and host lookup occurs during session logging. This value is set in the property `com.sun.am.session.enableHostLookUp`.
Notification	
Notification Pool Size	Specifies the total number of threads allowed in the notification thread pool. This value is set in the property `com.iplanet.am.notification.threadpool.size`.
Notification Thread Pool Threshold	Specifies the maximum task queue length for serving notification threads. This value is set in the property `com.iplanet.am.notification.threadpool.threshold`.
Validation	
Case Insensitive client DN comparison	Yes When enabled, the value is set to Yes, and the client distinguished name comparison is case-insensitive. This value is set in the property `com.sun.am.session.caseInsensitiveDN`.

5. Click Save.
6. (Optional) Click Inheritance Settings.

   The Inheritance Settings section lists server properties containing default values. A checked box indicates a property that can inherit default server properties. Unchecked properties can be overwritten for each server instance.

   - To select a property to be overwritten with default values, click its corresponding box until a check appears in the box.
   - To deselect a property and retain any custom configuration, click the property's corresponding box until the box contains no checkmark.
7. (Optional) Click Export Configuration.

   The OpenSSO STS console displays your settings so that you can inspect the settings for accuracy.
8. Click Save.
9. Click "Back to Server Profile."
10. Click "Back to Servers and Sites."

5.1.3.4 To Configure OpenSSO STS Server SDK Properties

1. On the Configuration tab, click the Servers and Sites tab.
2. In the Servers section, click the URL of the OpenSSO STS server you want to configure.
3. Click the SDK tab.
4. Provide values for Data Store, Event Service, LDAP Connection, Caching and Replica, and Time to Live Configuration properties.

   The following table provides a listing and descriptions of the properties you can configure.

Table 5–4 OpenSSO STS Server SDK Properties

Property	Description
Data Store	
Enable Datastore Notification	When enabled, the value is set to Yes, and backend datastore notification occurs. If this value is set to No, then in-memory notification is enabled. This value is set in the property `com.sun.identity.sm.enableDataStoreNotification`.
Enable Directory Proxy	When enabled, this value is set to Yes, and the Directory Proxy must be used for read, write, and/or modify operations to the Directory Server. This flag also determines if ACIs or delegation privileges are to be used. This value is set in the property `com.sun.identity.sm.ldap.enableProxy`.
Notification Pool Size	Specifies the size of the sm notification thread pool (total number of threads). This value is set in the property `com.sun.identity.sm.notification.threadpool.size`.
Event Service	
Number of retries for Event Service connections	Specifies the number of attempts to be made to successfully re-establish the Event Service connections. This value is set in the property `com.iplanet.am.event.connection.num.retries`.
Delay between Event Service connection retries	Specifies the number of milliseconds to delay between retries at re-establishing Event Service connections. This value i set in the property `com.iplanet.am.event.connection.delay.between.retries`.
Error codes for Event Service connection retries	Specifies the LDAP exception error codes to be triggered by retries at re-establishing Event Service connections. This value is set in the property `com.iplanet.am.event.connection.ldap.error.codes.retries`.
Idle Time Out	Specifies the number of minutes after which persistent searches will be restarted. This value is set in the property `com.sun.am.event.connection.idle.timeout`.
Disabled Event Service Connection	Specify which event connection (persistent search) is to be disabled. There are three valid values. Entries are case-sensitive: ■ aci - Access Control Instructions ■ sm - Service Management ■ um - User Management Multiple values are comma-separated. This value is set in the property `com.sun.am.event.connection.disable.list`.
LDAP Connection	
Number of retries for LDAP Connection	Specifies the number of attempts to be made to successfully re-establish LDAP Connection. This value is set in the property `com.iplanet.am.ldap.connection.delay.between.retries`.
Delay between LDAP connection retries	Specifies the number of milliseconds to delay between retries at re-establishing LDAP connections. This value is set in the property `com.iplanet.am.ldap.connection.num.retries`.

Table 5–4 (Cont.) OpenSSO STS Server SDK Properties

Property	Description
Error codes for LDAP connection retries	Specify the LDAP exception error codes to be triggered by retries at re-establishing LDAP connections. This value is set in the property `com.iplanet.am.ldap.connection.ldap.error.codes.retries`.
Caching and Replica	
SDK Caching Max. Size	Specifies the maximum size of the cache when SDK caching is enabled. The size should be an integer greater than 0, or default size (10000) will be used. This value is set in the property `com.iplanet.am.sdk.cache.maxSize`.
SDK Replica Retries	Specifies the number of times to retry when an Entry Not Found error is returned to the SDK. This value is set in the property `com.iplanet.am.replica.num.retries`.
Delay between SDK Replica Retries	Specifies the number of milliseconds to delay between the retries. This value is set in the property `com.iplanet.am.replica.delay.between.retries`.
Time to Live Configuration	
Cache Entry Expiration Enabled	When enabled, this value is set to Yes, and the cache entries expire based on the time specified in User Entry Expiration Time property. The default value is No. This value is set in the property `com.iplanet.am.sdk.cache.entry.expire.enabled`.
User Entry Expiration Time	Specifies the number of minutes entries remain valid in the cache after their last modification. After the time elapses (after the last modification/read from the directory), the data for the entry that is cached will expire. At that instant, new requests for data for these user entries will be read from the Directory. This value is set in the property `com.iplanet.am.sdk.cache.entry.user.expire.time`.
Default Entry Expiration Time	Specifies the number of minutes that non-user entries remain valid in the cache after their last modification. After this specified period of time elapses (after the last modification/read from the directory), the data for the entry that is cached will expire. At that instant, new requests for data for these non-user entries will be read from the Directory. This value is set in the property `com.iplanet.am.sdk.cache.entry.default.expire.time`.

5. Click Save.
6. (Optional) Click Inheritance Settings.

   The Inheritance Settings section lists server properties containing default values. A checked box indicates a property that can inherit default server properties. Unchecked properties can be overwritten for each server instance.

   - To select a property to be overwritten with default values, click its corresponding box until a check appears in the box.
   - To deselect a property and retain any custom configuration, click the property's corresponding box until the box contains no checkmark.
7. (Optional) Click Export Configuration.

The OpenSSO STS console displays your settings so that you can inspect the settings for accuracy.

8. Click Save.
9. Click "Back to Server Profile."
10. Click "Back to Servers and Sites."

5.1.3.5 To Configure OpenSSO STS Server Directory Configuration Properties

1. On the Configuration tab, click the Servers and Sites tab.
2. In the Servers section, click the URL of the OpenSSO STS server you want to configure.
3. Click the Directory Configuration tab.

   Provide values for the OpenSSO STS Server Directory Configuration properties. The following table provides a listing and descriptions of the properties you can configure.

Table 5–5 OpenSSO STS Server Directory Configuration Properties

Property	Description
Minimum Connection Pool	Specify the minimal size of connection pools to be used for connecting to the Directory Server, as specified in the LDAP server attribute. The default is value is 1.
Maximum Connection Pool	Specify the maximum size of connection pools to be used for connecting to the Directory Server, as specified in the LDAP server attribute. The default value is 10.
Bind DN	Specify the bind DN in the LDAP server.
Bind Password	Specify the password to be used for binding to the LDAP server. By default, the amldapuser password that was entered during installation is used as the bind user.

4. To add a configuration directory server to the Server list, click Add.
   - In the New Server page, provide the values for the New Directory Server properties, and then click OK. The following table provides a listing and descriptions of the properties you can configure.

Table 5–6 New Directory Server Properties

Property	Description
Name	Specify an identifier for this server.
Host Name	Specify the fully-qualified host name of the Directory Server. Example: *DirectoryServerHost.domainName.com*
Port Number	Specify the Directory Server port number.
Connection Type	Choose one of the following the connection type for the Directory Server: Simple SSL/TLS The default value is Simple.

   - To delete a Directory Server from the Server list, click to check the box corresponding to the Directory Server name, then click Delete.

5. Click Save.
6. Click "Back to Servers and Sites."

5.1.3.6 To Configure OpenSSO STS Server Advanced Properties

1. On the Configuration tab, click the Servers and Sites tab.
2. In the Servers section, click the URL of the OpenSSO STS server you want to configure.
3. Click the Advanced tab.

   The Advanced Properties section lists server properties containing default values.

   - To add a custom property to the list, click Add.

     A new row is added at the bottom of the list. In the appropriate columns, type a Property Name and Property Value.

   - To remove a property from the list, click to check the box corresponding to the property and then click Delete.
4. Click Save.
5. Click "Back to Servers and Sites."

5.1.4 To Clone an OpenSSO STS Server

1. On the Configuration tab, click the Servers and Sites subtab.
2. Click to mark the box corresponding to the server you want to clone.
3. Click Clone

   In the New Server page, in the Server URL field type the URL for the cloned server, and then click OK.
4. Configure the OpenSSO STS server. See Section 5.1.3, "To Configure an OpenSSO STS Server."

5.2 Managing OpenSSO STS Sites

The Servers and Sites configuration enables and administrator to manage multiple OpenSSO STS site and server configurations from a single console.

Multiple OpenSSO STS instances can be deployed on at least two different host servers. For example, you might deploy two instances on one server and a third instance on another server. Or you might deploy all instances on different servers. You can also configure the OpenSSO STS instances in session failover mode if required for your deployment.

One or more load balancers route client requests to the various OpenSSO STS instances in the environment. You configure each load balancer according to your deployment requirements. For example, you could use round-robin or load average load-balancing to distribute the load between the OpenSSO STS instances. A load balancer simplifies the deployment, as well as resolves issues caused by having a firewall between the client and the back-end OpenSSO STS servers. You can use a hardware or software load balancer with your OpenSSO STS deployment. All OpenSSO STS instances access the same Directory Server.

Important: If you make any changes to the configuration attributes for Servers and Sites, either through the console or the command line interface, you must restart the web container on which OpenSSO STS is deployed for the changes to take effect.

5.2.1 To Add a New OpenSSO STS Site

1. On the Configuration tab, click the Servers and Sites subtab.
2. In the Sites section, click New.
3. In the New Site page, in the Name field type a name for the new site.
4. In the Primary URL field, specify the Primary URL for the site instance, including the site URI.

   Use the form *protocol://hostname.domain:port/URI.*
5. Configure the new OpenSSO STS Site.

   See Section 5.2.2, "To Configure an OpenSSO STS Site."

5.2.2 To Configure an OpenSSO STS Site

1. On the Configuration tab, click the Servers and Sites subtab.
2. In the Sites section, click the name of the site you want to configure.
3. Provide values for the OpenSSO STS Site properties. The following table provides a listing and descriptions of the properties you can configure.

Table 5–7 OpenSSO STS Site Properties

Property	Description
Primary URL	Specify the primary URL used to access the site.
Secondary URLs	The Current Values list displays session repositories used for the session failover in an OpenSSO STS deployment. Use the URL of the load balancer as the identifier for this secondary configuration. If the secondary configuration is defined in this case, session failover is automatically enabled and becomes effective after the server restart. ▪ To add a new URL to the list, in the New Value field type the new URL, and then click Add. ▪ To remove an entry from the Current Values list, select the entry, and then click Remove.
Assigned Server	Servers assigned to the site.

4. Click Save.
5. Click "Back to Servers and Sites."

5.2.3 To Delete an OpenSSO STS Site

1. On the Configuration tab, click the Servers and Sites subtab.
2. In the Sites section, click check the box corresponding to the server you want to delete, and then click Delete.
3. Click Save.

5.3 Managing User Data Stores

A user data store, also called an identity repository, is a database where OpenSSO STS stores user attributes and user configuration data. Example: a user data store might contain a user's identifier and password, email address, application preferences and other forms of identity data. The OpenSSO STS interface enables a realm administrator to plug in one or more user data stores into the OpenSSO STS realm. OpenSSO STS provides identity repository plug-ins that in turn connect to a single LDAPv3 identity repository framework. The user data store plug-ins enable you to view and retrieve OpenSSO STS user information without having to make changes in your existing user database.

OpenSSO STS integrates data from the identity repository plug-in with data from other OpenSSO STS plug-ins to form a virtual identity for each user in the repository. OpenSSO STS can then use the universal identity in authentication and authorization processes among more than one identity repository. The virtual user identity is destroyed when the user's session ends.

All OpenSSO STS user data stores share the same underlying plug-in. Although most of the configuration attributes are the same for each of user data stores, the default attribute values vary depending upon the user data store type.

OpenSSO STS supports the following types of user data stores.

Active Directory
An Active Directory user data store uses the LDAP version 3 specification to write identity data to an instance of Microsoft Active Directory.

Generic LDAPv3
A generic LDAPv3 user data store allows identity data to be written to any LDAPv3–compliant database. Note - If the LDAPv3 database you are using does not support Persistent Search, then you cannot use the OpenSSO STS caching feature.

Sun Directory Server With OpenSSO Schema
A Sun Directory Server containing OpenSSO STS Schema resides in a Sun Directory Server instance itself and holds the OpenSSO STS information tree. It is different from the OpenSSO STS Repository Plug-in. A Directory Server with OpenSSO STS Schema contains more configuration attributes and enables you to better customize the user data store.

5.3.1 To Add a New User Data Store

1. On the Access Control tab, click the Data Stores subtab.
2. In the Data Stores section, click New.
3. In the Name field, type the new Data Store name.

   The Data Store name cannot contain spaces.
4. Choose one of the following:
   - Active Directory
   - Generic LDAPv3
   - Sun DS with OpenSSO schema
5. Click Next.
6. Provide values for the User Data Store properties. The following table provides a listing and descriptions of the properties you can configure.

Table 5–8 User Data Store Properties

Property	Description
LDAP Server	The Current Values list displays the name of the LDAP server or servers to which OpenSSO STS will be connected. If more than one LDAP server is listed, OpenSSO STS attempts to connect to the first host in the list. If a connection cannot be made to the first host in the list, then OpenSSO STS tries to access the next host in the list. ■ To add a new LDAP server, in the New Value field enter a server name using the following form: host.domain:portnumber, and then click Add. (Optional) You can append a server identifier and site identifier to the value of the LDAP Server attribute for redundancy. Use the form *host.domain:portnumber*|*serverID*|*siteID*. These identifiers are assigned to the server when they are configured globally. The identifier serverID designates a primary LDAP server. To designate other LDAP servers are as secondary and tertiary fallback servers. If no number is specified, the LDAP server is primary. The identifier siteID is a two-digit number generated internally by OpenSSO STS— for example, 02. To find this value, use an LDAP browser to find the following DN: `ou=accesspoint, ou=site_name, ou=com-sun-identitysites,ou=default, ou=GlobalConfig, ou=iPlanetAMPlatformService, ou=services,root-suffix` Under this DN, see *sunkeyvalue:primary-siteid=site-id* for the site identifier. Do not change the LDAP Server configuration for the OpenSSO STS embedded data store. This could result in unexpected data store behavior. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
LDAP Bind DN	Specify the DN that OpenSSO STS will use to authenticate to the LDAP server to which you are currently connected. The user with the DN used to bind to the LDAP server must have the appropriate privileges for adding, modifying, and deleting operations. These privileges are configured in the LDAPv3 Plugin Supported Types and Operations properties.
LDAP Bind Password	Specify the DN password that OpenSSO STS will use to authenticate to the LDAP server to which you are currently connected.
LDAP Bind Password (confirm)	Type the password again to confirm it.
LDAP Organization DN	Specify the DN to which this data store repository will map. This will be the base DN of all operations performed in this data store.
LDAP SSL	When enabled, OpenSSO STS will connect to the primary server using the HTTPS protocol.
LDAP Connection Pool Minimum Size	Specify the initial number of connections in the connection pool. Using a connection pool avoids having to create a new connection each time.

Table 5–8 (Cont.) User Data Store Properties

Property	Description
LDAP Connection Pool Maximum Size	Specify the maximum number of connections to allow.
Maximum Results Returned from Search	Specify the maximum number of entries returned from a search operation. If this limit is reached, Active Directory returns any entries that match the search request.
Search Timeout	Specify the maximum number of seconds allocated for a search request. If this limit is reached, Active Directory returns any search entries that match the search request.
LDAP Follows Referral	When enabled, referrals to other LDAP servers are followed automatically.
LDAPv3 Repository Plugin Class Name	Specify the location of the class file which implements the LDAPv3 repository.

Attribute Name Mapping	The Current Values list displays common attributes known to the OpenSSO STS framework to be mapped to the native data store. Example: if the framework uses inetUserStatus to determine user status, it is possible that the native data store actually uses userStatus. The attribute definitions are case-sensitive. The defaults are:
	`employeeNumber=distinguishedName`
	`portalAddress=sAMAccountName`
	`uid=sAMAccountName`
	`mail=userPrincipalName`
	`telephonenumber=displayName`
	`iplanet-am-user-alias-list=objectGUID`
	`userPassword=unicodePwd`
	■ To add a new Attribute Name Mapping, in the New Value field enter an attribute name-value pair, and then click Add.
	■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 5–8 (Cont.) User Data Store Properties

Property	Description

LDAPv3 Plug-in Supported Types and Operations	The Current Values list displays operations that are permitted or can be performed on this LDAP server. The default operations are the only operations that are supported by this LDAPv3 repository plug-in. The following are operations supported by LDAPv3 Repository Plugin:
	`agent: read, create, edit, delete`
	`role: read, create, edit, delete`
	`group: read, create, edit, delete`
	`realm: read, create, edit, delete, service`
	`user: read, create, edit, delete, service`
	■ To add a new LDAPv3 plug-in type and operations, in the New Value field, enter a new type:operations string, and then click Add.
	■ To remove an entry from the Current Values list, select the entry and them click Remove.
	You can remove permissions from all operations except for role operations based on your LDAP server settings and the tasks. You cannot add more permissions to any operation.
	If the configured LDAPv3 Repository plug-in is pointing to an instance of Sun Directory Server, permissions for the type role can be added. Otherwise, this permission may not be added because other data stores may not support roles.
	If a user is of supported type, then the read, edit, create, and delete operations allow you to read, edit, create, and delete user entries from the identity repository. The `user=service` operation enables OpenSSO STS to access attributes in user entries. Additionally, the user is allowed to access the dynamic service attributes if the service is assigned to the realm or role to which the user belongs.
	The user is also allowed to manage user attributes for any assigned service. If the user has service as the operation (user=service), then the following service-related operations are supported:
	`assignService`
	`unassignService`
	`getAssignedServices`
	`getServiceAttributes`
	`removeServiceAttributes`
	`modifyService`

LDAPv3 Plug-in Search Scope	Choose the scope to be used to find LDAPv3 plug-in entries. ■ `SCOPE_BASE` searches only the base DN. ■ `SCOPE_ONE` searches only the entries under the base DN. ■ `SCOPE_SUB` (default) searches the base DN and all entries within its subtree.
LDAP Users Search Attribute	Specify the attribute type to use to a search for a user. Example: if the user DN is `uid=user1,ou=people,dc=example,dc=com`, then enter `uid` in this field.
LDAP Users Search Filter	Specify the search filter to be used to find user entries.

Table 5–8 (Cont.) User Data Store Properties

Property	Description
LDAP User Object Class	The Current Values list displays the object classes for a user. When a user is created, this list of user object classes is added to the user's attributes list. ■ To add a new object class to the list, in the New Value field enter an object class name, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
LDAP User Attributes	The Current Values list displays the attributes associated with a user. You cannot read or write user attributes not on this list. The attributes are case-sensitive. The object classes and attribute schema must already be defined before you define the object classes and attribute schema here. ■ To add a new LDAP User Attribute, in the New Value field type an attribute name, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Create User Attribute Mapping	The Current Values list displays the attributes that are required when a user is created. Attributes uses the following syntax:
	■ To add a new user attribute mapping, in the New Values field enter a mapping using the following form:
	`DestinationAttributeName=SourceAttributeName`
	If the source attribute name is missing, the default is the user ID (`uid`). For example:
	`cn sn=givenName`
	Both `cn` and `sn` are required to create a user profile. The attribute `cn` gets the value of the attribute named `uid`, and the attribute `sn` gets the value of the attribute named `givenName`.
	■ To remove an entry from the Current Values list, select the entry and them click Remove.

Attribute Name of User Status	Specify an attribute name that indicates if the user is active or inactive.
User Status Active Value	This field is not displayed for the OpenSSO with Schema Data Store. This attribute value is assigned to the user when the user is created. LDAPv3 uses Active. Note used by Schema. ■ For a user to be active, the Active Directory value is 544. ■ For a user to be inactive, the Active Directory value is 546.
User Status Inactive Value	This field is not displayed for the OpenSSO with Schema Data Store. For Active Directory, this field is not used. LDAPv2 uses Inactive.
LDAP Groups Search Attribute	The Current Values list displays the attribute types to use for conducting a search on a group. The default is cn. ■ To add a new search attribute, in the New Value field enter an LDAP Group attribute. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 5–8 (Cont.) User Data Store Properties

Property	Description
LDAP Group Search Filter	Specify the search filter to be used to find group entries. The default is (`objectclass=groupOfUniqueNames`).
LDAP Groups Container Naming Attribute	Specify the naming attribute for a group container, if groups reside in a container. Otherwise, this attribute is left empty. Example: if a group DN of `cn=group1,ou=groups,dc=iplanet,dc=com` resides in `ou=groups,` then the group container naming attribute is `ou`.
LDAP Groups Container Value	Specify the value for the group container. Example: if a group DN of `cn=group1,ou=groups,dc=iplanet,dc=com` resides in a container named `ou=groups,` then the group container value is `groups`.
LDAP Groups Object Classes	The Current Values list displays object classes for groups. When a group is created, this list of group object classes will be added to the group's attributes list. ■ To add a new object class, in the New Value field type the object class name, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
LDAP Groups Attributes	The Current Values list displays attributes associated with a group. You cannot read or write group attributes that are not on this list. The attributes are case-sensitive. The object classes and attribute schema must be defined before you define the object classes and attribute schema here.
Attribute Name for Group Membership	Specify the name of the attribute whose values are the names of all the groups to which DN belongs. The default is `memberOf`.
Attribute Name of Unique Member	Specify the attribute name whose values is a DN belonging to this group. The default is `uniqueMember`.
Attribute Name of Group Member URL	Specify the name of the attribute whose value is an LDAP URL which resolves to members belonging to this group. The default is `memberUrl`.
Default Group Member's User DN	This field is not displayed for the OpenSSO with Schema Data Store.
LDAP Roles Search Attribute	This field is not displayed for Active Directory or LDAPv3 Data Stores. This field defines the attribute type for which to conduct a search on a role. The default is `cn`.
LDAP Roles Search Filter	This field is not displayed for Active Directory or LDAPv3 Data Stores. Specify the filter used to search for a role. The LDAP Role Search attribute is prepended to this value to form the actual role search filter. Exampe: if the LDAP Role Search Attribute is `CN` and LDAP Role Search Filter is `(objectClass=sunIdentityServerDevice)`, then the actual user search filter is: `(&(cn=*)(objectClass=sunIdentityServ erDevice))`
LDAP Roles Object Class	This field is not displayed for Active Directory or LDAPv3 Data Stores. Specify the object classes for roles. When a role is created, the list of user object classes will be added to the role's attributes list

Table 5–8 (Cont.) User Data Store Properties

Property	Description
LDAP Roles Attributes	This field is not displayed for Active Directory or LDAPv3 Data Stores. The Current Values list displays attributes associated with a role. Reading or writing agent attributes that are not on this list is not allowed. The attributes are case-sensitive. The object classes and attribute schema must be defined in Directory Server before you define the object classes and attribute schema here.
LDAP Filter Roles Search Attribute	This field is not displayed for Active Directory or LDAPv3 Data Stores. Specify the attribute type for which to conduct a search on a filter role. The default is `cn`.
LDAP Filter Roles Search Filter	This field is not displayed for Active Directory or LDAPv3 Data Stores. The Current Values list displays the filter used to search for a filtered role. The LDAP Filter Role Search attribute is prepended to this field to form the actual filtered role search filter.Exampe: if the LDAP Filter Role Search Attribute is CN and LDAP Filter Role Search Filter is `(objectClass=sunIdentityServerDevice)`, then the actual user search filter will be: `(&(cn=*)(objectClass=sunIdentityServ erDevice))`
LDAP Filter Roles Object Class	This field is not displayed for Active Directory or LDAPv3 Data Stores. The Current Values list displays the object classes for filtered roles. When a filtered role is created, the list of user object classes will be added to the filtered role's attributes list
LDAP Filter Roles Attributes	This field is not displayed for Active Directory or LDAPv3 Data Stores. The Current Values list displays attributes associated with a filtered role. Reading or writing agent attributes that are not on this list is not allowed. The attributes are case-sensitive. The object classes and attribute schema must be defined in Directory Server before you define the object classes and attribute schema here.
Attribute Name for Filtered Role Membership	This field is not displayed for Active Directory or LDAPv3 Data Stores.
Attribute Name of Role Membership	This field is not displayed for Active Directory or LDAPv3 Data Stores.
Attribute Name of Filtered Role Filter	This field is not displayed for Active Directory or LDAPv3 Data Stores.
LDAP People Container Naming Attribute	■ If a user resides in a people container, then specify the naming attribute of the people container. ■ If the user does not reside in a people container, then leave this field blank.
LDAP People Container Value	Specify the value of the people container. The default is people. Caution – The entire tree under the baseDN will be searched if the value of this attribute is set to null (empty).
Identity Types That Can be Authenticated	Specify that this data store can authenticate user and/or agent identity types when the authentication module mode for the realm is set to Data Store.

Table 5–8 (Cont.) User Data Store Properties

Property	Description
Authentication Naming Attribute	This value is currently not used.
Persistent Search Base DN	Specify the base DN to use for persistent search. Some LDAPv3 servers only support persistent search at the root suffix level.
Persistent Search Filter	Specify the filter that will return the specific changes to directory server entries. The data store will only receive the changes that match the defined filter.
Persistent Search Scope	Specify the scope to be used in a persistent search. The scope must be one of the following: ▪ `SCOPE_BASE` searches only the base DN. ▪ `SCOPE_ONE` searches only the entries under the base DN. ▪ `SCOPE_SUB` (default) searches the base DN and all entries within its subtree.
Persistent Search Maximum Idle Time Before Restart	Specify the maximum idle time before restarting the persistence search. The value must be great than 1. Values less than or equal to 1 will restart the search regardless of the idle time of the connection. If OpenSSO STS is deployed with a load balancer, some load balancers will time out if it has been idle for a specified amount of time. In this case, you should set the Persistent Search Maximum Idle Time Before Restart to a value less than the specified time for the load balancer.
Maximum Number of Retries After Error Code	Specify the maximum number of retries for the persistent search operation if it encounters the error codes specified in LDAP Exception Error Codes to Retry On.
The Delay Time Between Retries	Specify the time to wait before each retry. This only applies to persistent search connection.
LDAP Exception Error Codes to Retry	Specify the error codes to initiate a retry for the persistent search operation. This attribute is only applicable for the persistent search, and not for all LDAP operations.
Caching	When enabled, OpenSSO STS caches data retrieved from the data store.
Maximum Age of Cached Items	Specify the maximum number of seconds data is stored in the cache before it is removed.
Maximum Size of the Cache	Specify in number of bytes the maximum size of the cache. The larger the value, the more data can be stored, but it will require more memory.

7. Click Finish.

5.3.2 To Delete a User Data Store

1. In the Access Control tab, click the Data Stores subtab.
2. Click to mark the box corresponding to the data store or data stores you want to delete.
3. Click Delete.

5.4 Configuring Global Platform Attributes

1. On the Configuration tab, click the System subtab.
2. On the System Configuration page, in the System Attributes list, click Platform.
3. Provide values for the Global Platform Attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 5–9 Global Platform Attributes

Attribute	Description
Platform Locale	Specify the default language subtype that OpenSSO STS was installed with. The Authentication, Logging and administration services are administered in the language of this value. The default is `en_US`.
Cookie domains	The Current Values list displays domains that will be returned in the cookie header when setting a cookie to the user's browser during authentication. ■ To add a cookie domain to the list, in the New Value field type the domain name, and then click Add. The default value for this field is the domain of the installed OpenSSO STS instance. If the list is empty, no cookie domain will be set. The OpenSSO STS session cookie will be forwarded to only OpenSSO STS itself and to no other servers in the domain. If SSO is required with other servers in the domain, set this attribute with the cookie domain. If you had two interfaces in different domains on one OpenSSO STS instance, then you must set both cookie domains in this attribute. If a load balancer is used, the cookie domain must be that of the load balancer's domain; do not use the cookie domain of the servers behind the load balancer. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Hex Encode Cookie	When set to Yes, hex encoding for cookies is enabled. The default is No.

4. (Optional) To add a new character set, in the Client Character Sets list, click New.

   To delete a character set, in the Client Character Sets section click to mark the box corresponding to the character set you want to remove, and then click Delete.
5. Click OK.

6

Managing the OpenSSO STS Authentication Service

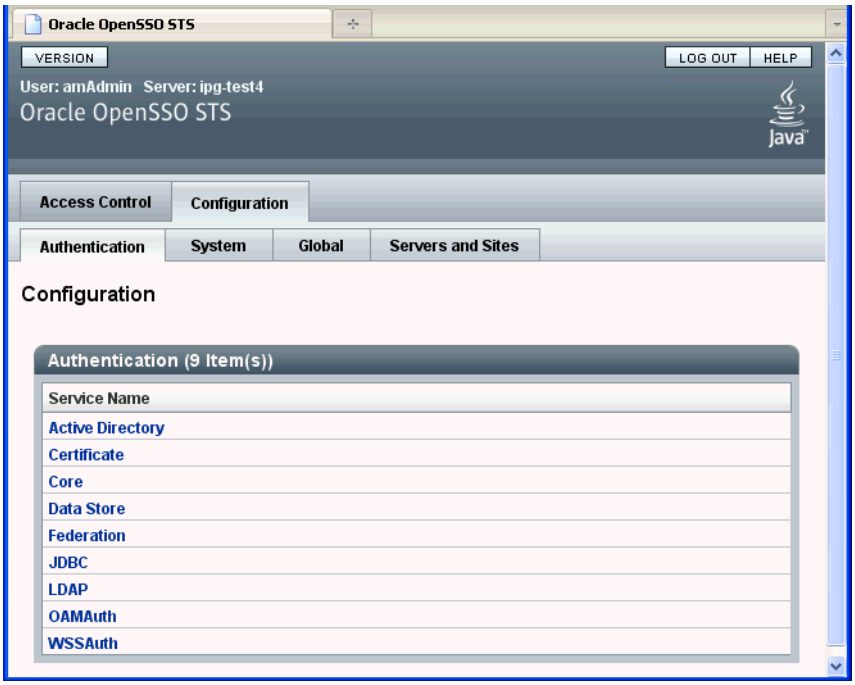
The Oracle OpenSSO Security Token Service (OpenSSO STS) Authentication Service retrieves credentials from an end-user, administrator, or client application and validates the credentials against a configured identity repository. Use the Access Control interface to manage the authentication modules and user data stores that OpenSSO STS uses to authenticate incoming security token requests.

This chapter contains the following sections:

6.1 Configuring Global Authentication Service Properties

The Configuration > Authentication subtab is where you configure global properties for a type of authentication service. The following figures illustrates the subtab in the OpenSSO STS console where you configure global authentication service properties. The figure also lists the types of authentication service supported in OpenSSO STS.

Figure 6–1 Configuration > Authentication Subtab for Configuring Global Authentication Service Properties



- To Configure Active Directory Authentication Service Attributes
- To Configure Certificate Authentication Service Realm Attributes
- To Configure Core Authentication Service Attributes
- To Configure Data Store Authentication Service Attributes
- To Configure Federation Authentication Service Attributes
- To Configure JDBC Authentication Service Realm Attributes
- To Configure LDAP Authentication Service Realm Attributes
- To Configure WSSAuth Authentication Service Attributes
- To Configure the Authentication Realm

6.1.1 To Configure Active Directory Authentication Service Attributes

1. On the Configuration tab, click the Authentication subtab.
2. In the Authentication list, click Active Directory.
3. Provide values for the Active Directory Authentication Service Realm attributes. The following table provides a listing and descriptions of the attributes you can configure.

Table 6–1 Active Directory Authentication Service Realm Attributes

Attribute	Description
Primary Active Directory Server	The Current Values list displays the host name and port number of the primary Active Directory server specified during OpenSSO STS installation. This is the first server contacted for Active Directory authentication. The format is hostname:port. The default port number is 389. ■ To add a new Active Directory server to the list, then click Add. If you have OpenSSO STS deployed with multiple domains, you can specify the communication link between specific instances of OpenSSO STS and Directory Server in using the form LocalServerName | Server:PortNumber. For multiple entries, each entry must be prefixed with a local server name. Example: *local_servername* | *server:port local_servername2* | *server2:port2* ... For example, if you have two OpenSSO STS instances deployed in different locations (*L1-machine1-IS and L2-machine2-IS*) communicating with different instances of Directory Server (*L1-machine1-DS and L2-machine2-DS*), use the form: *L1-machine1-IS.example.com* | *L1-machine1-DS* `.example.com:389` *L2-machine2-IS.example.com* | *L2-machine2-DS* `.example.com:389` ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Secondary Active Directory Server	The Current Values list displays the host name and port number of a secondary Active Directory server available to the OpenSSO STS platform. If the primary Active Directory server does not respond to a request for authentication, then this server is contacted. If the primary server is up, OpenSSO STS will switch back to the primary server. ■ To add an Active Directory server to the list, in the New Value field Type the name of the new server, and then click Add. Use the form hostname:port. Multiple entries must be prefixed by the local server name. Caution – When authenticating users from a Directory Server that is remote from the OpenSSO STS server, both the Primary and Secondary LDAP Server Ports must have values. The value for one Directory Server location can be used for both fields. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–1 (Cont.) Active Directory Authentication Service Realm Attributes

Attribute	Description
DN to Start User Search	The Current Values list displays the DN of the node where the search for a user starts. ■ To add a new base DN to the list, in the New Value field Type the new DN, and then click Add. Use the form servername|searchDN. For performance reasons, this DN should be as specific as possible. The default value is the root of the directory tree. Any valid DN will be recognized. If OBJECT is selected in the Search Scope attribute, the DN should specify one level above the level in which the profile exists. Multiple entries must be prefixed by the local server name. Example: *servername1*|*searchDN* *servername2*|*searchDN* *servername3*|*searchDN...* If multiple entries exist under the root organization with the same user ID, then this parameter should be set so that only one entry can be searched for or found in order to be authenticated. For example, in the case where the agent ID and user ID are under the same root org, this parameter should be ou=Agents for the root organization to authenticate using Agent ID and ou=People, for the root organization to authenticate using User ID. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
DN for Root User Bind	Specify the DN of the user that will be used to bind to the Directory Server specified in the Primary LDAP Server and Port fields as administrator. The authentication service must bind as this DN in order to search for a matching user DN based on the user login ID. The default is amldapuser. Any valid DN will be recognized. Make sure that password is correct before you logout. If it is incorrect, you will be locked out. If this should occur, you can login with the super user DN. By default, this the amAdmin account with which you would normally log in, although you will use the full DN. For example: `uid_amAdmin,ou=People,` *OpenSSO-deploy-base*
Password for Root User Bind	Type the password for the administrator profile specified in the DN for Root User Bind field. There is no default value. Only the administrator's valid Active Directory password is recognized.
Password for Root User Bind (confirm)	Type the Root User Bind password again to confirm it.
Attribute Used to Retrieve User Profile	Specify the attribute used for the user entry naming convention. By default, OpenSSO STS assumes that user entries are identified by the uid attribute. If your Directory Server uses a different attribute such as givenname, specify the attribute name in this field.

Table 6–1 (Cont.) Active Directory Authentication Service Realm Attributes

Attribute	Description
Attributes Used to Search for a User to be Authenticated	The Current Values list displays the attributes to be used to form the search filter for a user that is to be authenticated, and that allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to uid, employeenumber, and mail, then the user could authenticate with any of these names. ■ To add an attribute to the list, in the New Value field Type the attribute, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
User Search Filter	Displays the attributes to be used to find the user based on the value in the DN to Start User Search field. The filter works with the User Naming Attribute. There is no default value. Any valid user entry attribute will be recognized.
Search Scope	Choose the number of levels in the Directory Server that will be searched for a matching user profile. The search begins from the node specified in DN to Start User Search field. The default value is SUBTREE. Choose one of the following: ■ OBJECT searches only the specified node. ■ ONELEVEL searches at the level of the specified node and one level down. ■ SUBTREE searches all entries at and below the specified node.
SSL Access to Active Directory Server	When enabled, OpenSSO STS uses the SSL protocol to access the Directory Server specified in the Primary and Secondary Server and Port fields. By default, the box is not checked and the SSL protocol is not used to access the Directory Server. If the Active Directory server is running with SSL enabled (LDAPS), you must make sure that OpenSSO STS is configured with proper SSL trusted certificates. Otherwise OpenSSO STS cannot connect to Directory Server using the LDAPS protocol.
Return User DN to Authenticate	When enabled, the Active Directory authentication module instance returns the DN instead of the User ID, and no search is necessary. Normally, an authentication module instance returns only the User ID, and the authentication service searches for the user in the local OpenSSO STS instance. If the OpenSSO STS directory is the same as the directory configured for Active Directory, this option may be enabled. If an external Active Directory is used, this option is typically not enabled.
Active Directory Server Check Interval	Specify the number of minutes per interval in which a thread will "sleep" before verifying that the primary Active Directory server is running. This attribute is used for Active Directory Server failback.

Table 6–1 (Cont.) Active Directory Authentication Service Realm Attributes

Attribute	Description
User Creation Attributes	The Current Values list displays attributes used by the Active Directory authentication module instance when the Active Directory server is configured as an external Active Directory server. It contains a mapping of attributes between a local and an external Directory Server. The attribute uses the following form: *attr1*|*externalattr1 attr2*|*externalattr2* ▪ To add a new attribute, in the New Value field Type the attribute and then click Add. Use the form: *attr1*|*externalattr1 attr2*|*externalattr2* When this attribute is populated, the values of the external attributes are read from the external Directory Server, and are set for the internal Directory Server attributes. The values of the external attributes are set in the internal attributes only when the User Profileattribute (in the Core Authentication module type) is set to Dynamically Created and the user does not exist in local Directory Server instance. The newly created user will contain the values for internal attributes, as specified in User Creation Attributes List, with the external attribute values to which they map. ▪ To remove an entry from the Current Values list, select the entry and them click Remove.
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*`://`*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6.1.2 To Configure Certificate Authentication Service Realm Attributes

1. On the Configuration tab, click the Authentication subtab.
2. In the Authentication list, click Certificate.

3. Provide values for the Certificate Authentication Realm attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–2 Certificate Authentication Service Realm Attributes

Attribute	Description
Match Certificate in LDAP	When enabled, the Certificate Authentication Module determines whether a user certificate presented at login is stored in the LDAP Server specified in the "LDAP Server Where Certificates are Stored" field below. If no match is found, then the user is denied access. If a match is found and no other validation is required, the user is granted access. By default, this option is disabled and the Certificate Authentication Module does not check for the user certificate. Note – A certificate stored in the Directory Server is not necessarily valid. It may be on the certificate revocation list. See Match Certificate to CRL. However, the web container may check the validity of the user certificate presented at login.
Subject DN Attribute Used to Search LDAP for Certificates	Specify the attribute of the certificate's SubjectDN value that will be used to search LDAP for certificates. This attribute must uniquely identify a user entry. The actual value will be used for the search. The default is cn.
Match Certificate to CRL	When enabled, the Certificate Authentication Module compares the user certificate against the Certificate Revocation List (CRL) in the LDAP Server. The CRL is located based on one of the attribute names in the issuer's SubjectDN. If the certificate is on the CRL, then the user is denied access. If the certificate is not on the CRL, then the user is allowed to proceed. By default this option is disabled. Certificates should be revoked when the owner of the certificate has changed status and no longer has the right to use the certificate or when the private key of a certificate owner has been compromised.
Issuer DN Attribute Used to Search LDAP for CRLs	Specify the attribute of the `subjectDN` for the certificate's issuer. The subjectDN value will be used to search LDAP for CRLs. This field is used only when the Match Certificate to CRL attribute is enabled. The actual value will be used for the search. The default is cn.
HTTP Parameters for CRL Update	Specify the HTTP parameters for obtaining a CRL from a servlet for a CRL update. Contact the administrator of your CA for these parameters.

Table 6–2 (Cont.) Certificate Authentication Service Realm Attributes

Attribute	Description
OCSP Validation	When enabled, Online Certificate Status Protocol (OCSP) validation is performed by contacting the appropriate OCSP responder. The OCSP responder is determined during runtime based on the following settings: ■ If this value is set to true, and the OCSP responder is set in the Responder URL attribute, then the value of the attribute will be used as the OCSP responder. ■ If Online Certificate Status Protocol Check is enabled and if the value of this attribute is not set, then the OCSP responder presented in your client certificate is used as the OCSP responder. ■ If Online Certificate Status Protocol Check is not enabled, or if Online Certificate Status Protocol Check is enabled but an OCSP responder can not be found, then no OCSP validation will be performed. These settings can be configured on the Servers and Sites tab. Before enabling OCSP Validation, make sure that the time of day settings for the OpenSSO STS host and the OCSP responder host are synchronized as closely as possible. Also, the time of day setting for the OpenSSO STS host must be ahead of the time of day setting for the OCSP responder. For example, if the OCSP responder host is set at 12:00:00 PM, then the OpenSSO STS host could be set at 12:00:30 PM.
LDAP Server Where Certificates are Stored	The Current Values list displays the name and port number of the LDAP server where the certificates are stored. The default value is the host name and port specified when OpenSSO STS was installed. ■ To add a new LDAP server, in the New Value field type the server identifier, and then click Add. Use the form hostname:port. You can specify any LDAP server where the certificates are stored. When entering multiple entries, each entry must be prefixed with a local server name. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–2 (Cont.) Certificate Authentication Service Realm Attributes

Attribute	Description
LDAP Search Start DN	The Current Values list displays the DN of the node where the search for the user's certificate should start. ■ To add a DN to the list, in the New Value field type the new DN, and then click Add. Use the format *servername*|*searchDN*. There is no default value. You can enter any valid DN. Multiple entries must be prefixed by the local server name. Example: *servername1*|*searchDN servername2*|*searchDN servername3*|*searchDN* If multiple entries exist under the root organization with the same user ID, then this parameter should be set so that the only one entry can be searched for or found in order to be authenticated. For example, in the case where the agent ID and user ID is same under root org, this parameter should be `ou=Agents` for the root organization to authenticate using `Agent ID` and `ou=People`, for the root organization to authenticate using User ID. ■ To remove an entry from the Current Values list, select the entry and them click Remove. Use the format *servername*|*searchDN*. There is no default value. You can enter any valid DN. Multiple entries must be prefixed by the local server name. Example: *servername1*|*searchDN servername2*|*searchDN servername3*|*searchDN*
LDAP Server Principal User	Specify the DN of the principal user for the LDAP server where the certificates are stored. There is no default value. You can use any valid DN. The principal user must be authorized to read, and search certificate information stored in the Directory Server.
LDAP Server Principal Password	Specify the LDAP password associated with the user specified in the LDAP Server Principal User field above. There is no default value. You can use any valid LDAP password for the specified principal user. This value is stored as readable text in the directory.
LDAP Server Principal Password (confirm	Type the password again to confirm it.
Use SSL for LDAP Access	Specifies whether to use SSL to access the LDAP server. The default is that the Certificate Authentication service does not use SSL for LDAP access.

Certificate Field Used to Access User Profile	From the following, choose the field in the certificate's Subject DN to be used to search for a matching user profile:
	`email address`
	`none`
	`other`
	`subject CN`
	`subject DN`
	`subject UID`
	For example, if you choose email address, the Certificate Authentication service searches for the user profile that matches the attribute milder in the user certificate. The user logging in then uses the matched profile. The default field is subject CN.

Table 6–2 (Cont.) Certificate Authentication Service Realm Attributes

Attribute	Description
Other Certificate Field Used to Access User Profile	This attribute is recognized only if 'other' is selected in the 'Certificate Field Used to Access User Profile' attribute above. Specify the attribute that will be selected from the received certificate's subjectDN value. The Certificate Authentication service will then search the user profile that matches the value of that attribute.
SubjectAltNameExt Value Type to Access User Profile	RFC822Name - Electronic email address UPN - User Principal Name none When 'none' is selected, the 'Certificate Field Used to Access User Profile' or 'Other Certificate Field Used to Access User Profile' attribute is used to access the User Profile.
Trusted Remote Hosts	The Current Values list displays hosts that can be trusted to send certificates to OpenSSO STS. OpenSSO STS must verify whether the certificate came from one of these hosts. This attribute is used for the Portal Server gateway, for a load balancer with SSL termination and for Distributed Authentication. By default, this attribute is set to 'none,' which disables certificate issuer host verification. ■ To add a host to this list, in the New Value field type one of the following, and then click Add. none - Disables certificate issuer host verification. This is set by default. all - Accepts Portal Server Gateway-style certificate authentication from any client IP address. IP ADDR -Lists the IP addresses from which to accept Portal Server Gateway-style certificate authentication requests (the IP Address of the Gateway(s)). The attribute is configurable on an realm basis. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
SSL Port Number	Specify the port number for the secure socket layer (SSL). Currently, this attribute is only used by the Gateway servlet. Before you add or change an SSL Port Number, see the "Policy-Based Resource Management" section in the OpenSSO STS Administration Guide.
HTTP Header Name for Client Certificate	This attribute is used only when the Trusted Remote Hosts attribute is set to all' or has a specific host name defined. Specify the HTTP header name for the client certificate that is inserted by the load balancer or Secure Remote Access component.

Table 6–2 (Cont.) Certificate Authentication Service Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6.1.3 To Configure Core Authentication Service Attributes

1. On the Configuration tab, click the Authentication subtab.
2. In the Authentication list, click Core.
3. On the Core page, provide values for the Core Authentication Service global attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–3 Core Authentication Service Global Attributes

Attribute	Description
Pluggable Authentication Module Classes	The Current Values list displays the Java classes of the available authentication modules. ■ To add a Java class to the list, in the New Value field type a Java class name, and then click Add. Use a text string specifying the full class name, including the package name, of the authentication module. If you write a custom authentication module by implementing the OpenSSO STS AMLoginModule or the Java Authentication and Authorization Service [JAAS] LoginModule service provider interfaces, the new class value must be added to this property. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–3 (Cont.) Core Authentication Service Global Attributes

Attribute	Description
Supported Authentication Modules for Clients	The Current Values list displays authentication modules supported for a specific client. Formatted as: ■ To add an authentication module to the list, in the New Value field type the client type and module name, and then click Add. Use the form *clientType* | *module1,module2,module3* ■ To remove an entry from the Current Values list, select the entry and them click Remove.
LDAP Connection Pool Size	The Current Values list displays the minimum and maximum connection pool size to be used on a specific LDAP server and port. Formatted as: *host:port:min:max*This attribute is for LDAP and Membership authentication services only. ■ To add an entry to the list, in the New Value field type the entry, and then click Add. Use the form *host:port:min:max*. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Default LDAP Connection Pool Size	Specify the default minimum and maximum connection pool to be used with all LDAP authentication module configurations. Use the format *min:max*. This value is superseded by a value defined for a specific host and port in the LDAP Connection Pool Size property.
Remote Auto Security	When enabled, OpenSSO STS validates the identity of the calling application, and all remote authentication requests require the calling application's `SSOToken`. This allows the Authentication Service to obtain the username and password associated with the application.
Keep Post Process Objects for Logout Processing	When enabled, the remote Auth Client must send the application `SSOToken` with each request.
Keep Authentication Module Objects for Logout Processing	When enabled, the user session continues to hold the instances of authentication modules after authentication is complete. This may be required for special logout processing.

4. In the Realm Attributes section, values for the Core Authentication Service Realm attributes. The following table provides a listing and descriptions of the attributes you can configure.

Table 6–4 Core Authentication Service Realm Attributes

Property	Desription
User Profile	Choose a profile status for a successfully authenticated user. ■ Dynamic On successful authentication the Authentication Service will create a user profile if one does not already exist. The SSOToken will then be issued. The user profile is created in the realm's configured user data store. ■ Dynamic with User Alias On successful authentication the Authentication Service will create a user profile that contains the User Alias List attribute which defines one or more aliases that for mapping a user's multiple profiles. ■ Ignored A user profile is not required for the Authentication Service to issue an SSOToken after a successful authentication. ■ Required On successful authentication the user must have a user profile in the realm's configured user data store in order for the Authentication Service to issue an SSOToken.
Administrator Authentication Configuration	Specify the authentication configuration that is invoked when the user accesses /openssosts/console directly. ■ [empty] ■ ldapService
User Profile Dynamic Creation Default Roles	The Current Values list displays DNs of roles to be assigned to a new user whose profile is created when Dynamic or Dynamic with Alias is selected under the User Profile attribute. There are no default values. A role can be either an OpenSSO STS or LDAP role, but it cannot be a filtered role. ■ To add a new role, in the New Value field type a role name, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Persistent Cookie Mode	Determines whether users can return to their authenticated session after restarting the browser. When enabled, a user session will not expire until its persistent cookie expires (as specified by the value of the Persistent Cookie Maximum Time attribute), or the user explicitly logs out. By default, the Authentication Service uses only memory cookies (expires when the browser is closed). The client must explicitly request a persistent cookie by appending the iPSPCCookie=yes parameter to the login URL.
Persistent Cookie Maximum Time	Specify the number of seconds after which a persistent cookie expires. The interval begins when the user session is successfully authentication. Persistent cookie mode must be enabled. The field will accept any integer value less than the maximum 214748647.

Table 6–4 (Cont.) Core Authentication Service Realm Attributes

Property	Desription
Alias Search Attribute Name	The Current Values list displays secondary LDAP attributes to use to search for a user profile when a search using the primary LDAP attribute has failed. This attribute is typically used when the user identification returned from an authentication module is not the same as that specified in the User Naming Attribute. ■ To add a new attribute name, in the New Values field enter the new attribute name, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove. For example, a Certificate server might return abc1234, but the username is abc. There is no default value for this attribute. The field takes any valid LDAP attribute.
Default Authentication Locale	Specify the default language subtype to be used by the Authentication Service. The default value is en_US. To use a difference locale, a directory containing authentication templates for that locale must already exist.
Organization Authentication Configuration	Choose the default authentication chain used the users in the realm. ■ [empty] - No authentication chain is configured. ■ ldapService - Default authentication chain name for the LDAP authentication module.
Login Failure Lockout Mode	When enabled, the user is locked out or prevented from authenticating after repeated unsuccessful login attempt within a specified interval. Lockout criteria are defined in the Login Failure Lockout Count and Login Failure Lockout Interval attributes below.
Login Failure Lockout Count	Specify the number of times a user can attempt to authenticate within the interval defined in the Login Failure Lockout Interval property. When the user exceeds this number, the user is locked out or prevented from further authentication attempts.
Login Failure Lockout Interval	Specify in minutes the interval during which failed login attempts are counted. The lockout interval begins when a user first attempts to authenticate. The lockout count begins after two consecutive failed logins. The user is locked out if the number of attempts reaches the number specified in the Login Failure Lockout Count. If the user successfully authenticates within the Login Failure Lockout Interval, the lockout count is reset.
Email Address to Send Lockout Notification	Specify an email address or multiple email addresses to which notification will be sent if a user lockout occurs. For multiple addresses, separate each address with a space. For non-English locales, use the following format: *email_address* | *locale* | *charset*
Warn User After N Failures	Specify the number of authentication failures that can occur before OpenSSO STS displays a warning message to the user that the user will be locked out.
Login Failure Lockout Duration	Specify in minutes how long a user must wait after a lockout before attempting to authenticate again. If you enter a value greater than 0, then memory lockout is enabled and physical lockout is disabled. When memory lockout is enabled, the user account is locked in memory for the number of minutes you specified. The account is unlocked after that time has elapsed.

Table 6–4 (Cont.) Core Authentication Service Realm Attributes

Property	Desription
Lockout Duration Multiplier	Specify a value used to multiply the Login Failure Lockout Duration value for each successive lockout. The Lockout Duration is incrementally increased based on the number of times the user has been locked out. For example, if the Login Failure Lockout Duration is set to 3 minutes, and the Lockout Duration Multiplier is to 2, then the user will be locked out of the account for 6 minutes. After the 6 minutes has elapsed, if the user again provides the wrong credentials, the lockout duration is now 12 minutes.
Lockout Attribute Name	Specify the LDAP attribute to be used for physical lockout. The default value is inetuserstatus even when the field is empty. The Lockout Attribute Value field must also contain an appropriate value.
Lockout Attribute Value	Specify the action to taken on the attribute defined in the Lockout Attribute Name. The default value is inactive even if the field is empty. The Lockout Attribute Name field must also contain an appropriate value.
Default Success login URL	The Current Values list displays values that specify where users are directed after successful authentication. ■ To add a new URL, in the New Value field type the URL, and then click Add. Use the form client-type|URL. The only value you can specify at this time is a URL which assumes the type HTML. The default value is /opensso/console. Values that don't specify HTTP or HTTP(S) are appended to the deployment URL. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Default Failure Login URL	The Current Values list displays where users are directed after a failed authentication attempt. ■ To add a new URL, in the New Value field type the URL, and then click Add. Use the form *client-type*|*URL*. The only value you can specify at this time is a URL which assumes the type HTML. The default value is `/opensso/console`. Values that don't specify HTTP or HTTP(S) are appended to the deployment URL. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Authentication Post Processing Class	The Current Values list displays a Java class or multiple Java classes to be used for customizing post-authentication processes for either successful or unsuccessful logins. ■ To add a new class, in the New Value field type the class name, and then click Add. Example: `com.abc.authentication.PostProcessClass` The Java class must implement the `com.sun.identity.authentication.spi.AMPostAuthProcessInterfaceOpenSSOEnterprise` interface. Additionally, a JAR containing the post-processing class must be added to the classpath of the web container instance on which OpenSSO STS is configured. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–4 (Cont.) Core Authentication Service Realm Attributes

Property	Desription
Generate UserID Mode	When enabled, if the user identifier entered by a user during the self-registration process is not valid or already existing, the Membership module will generate a list of alternate user identifiers. The user identifiers are generated by the class specified in the Pluggable User Name Generator Class property.
Pluggable User Name Generator Class	Specify the name of the class to be used for generating alternate user identifiers when Generate UserID Mode is enabled. The default value is `com.sun.identity.authentication.spi.DefaultUserIDGenerator.`
Identity Types	Click a box to mark the type or types of identities for which OpenSSO STS will search.
Pluggable User Status Event Classes.	The Current Values list displays the Java classes or Java classes used to provide a callback mechanism for user status changes during the authentication process. ■ To add a new class, in the New Value field type the class name, and then click Add. Example: `com.abc.authentication.PostProcessClass` The Java class must implement the OpenSSO STS interface `com.sun.identity.authentication.spi.AMAuthCallBack.` Account lockout and password changes are supported. Password changes are supported through the LDAP authentication module. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Store Invalid Attempts in Data Store	When enabled, information regarding failed authentication attempts is stored as the value of the sunAMAuthInvalidAttemptsData attribute in the user data store. To store data in this attribute, the OpenSSO STS schema must be loaded. Information stored includes the number of invalid attempts, time of last failed attempt, lockout time, and lockout duration. Storing this information in the identity repository allows the information to be shared among multiple instances of OpenSSO STS.
Module Based Authentication	When enabled, users authenticate using module-based authentication. When disabled, all attempts at authentication using the *module=module-instance-name* login parameter will fail.
Use Attribute Mapping to Session Attribute	The Current Values list displays user identity attributes that are mapped as session attributes in the user's SSOToken. ■ To add a new attribute mapping, in the New Value field type a new attribute-value pair, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove. Use the form *User-Profile-Attribute* | *Session-Attribute-Name*. If *Session-Attribute-Name* is not specified, the value of User-Profile-Attribute is used. All session attributes contain the am.protected prefix to ensure that they cannot be edited by the Client SDK.

Table 6–4 (Cont.) Core Authentication Service Realm Attributes

Property	Desription
Default Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*`://`*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

5. Click Save.

6.1.4 To Configure Data Store Authentication Service Attributes

1. On the Configuration tab, click the Authentication subtab.
2. In the Authentication list, click Data Store
3. On the Data Store Realm Attributes page, provide the Authentication Level value. The following table provides information about the Authentication Level attribute.

Table 6–5 Data Store Authentication Service Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*`://`*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

4. Click Save.

6.1.5 To Configure Federation Authentication Service Attributes

1. On the Configuration tab, click the Authentication subtab.
2. In the Authentication list, click Federation.
3. On the Data Store Realm Attributes page, provide the Authentication Level value. The following table provides information about the Authentication Level attribute.

Table 6–6 Data Store Authentication Service Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

4. Click Save.

6.1.6 To Configure JDBC Authentication Service Realm Attributes

1. On the Configuration tab, click the Authentication subtab.
2. In the Authentication list, click JDBC
3. On the JDBC Realm Attributes page, provide values for the JDBC Authentication Service Realm attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–7 JDBC Authentication Service Realm Attributes

Attribute	Description
Connection Type	Choose the type of connection to be made to the SQL database. ■ Connection pool is retrieved via JNDI The Java Naming and Directory Interface (JNDI) connection pool uses the configuration from the underlying web container. ■ Non-persistent JDBC connection. The Java Database Connectivity (JDBC) API provides a call-level API for SQL-based database access.

Table 6–7 (Cont.) JDBC Authentication Service Realm Attributes

Attribute	Description
Connection Pool JNDI Name	If JNDI is selected in Connection Type, this field specifies the connection pool name. Because JDBC authentication uses the JNDI connection pool provided by the web container, the setup of JNDI connection pool may not be consistent among other web containers. See the OpenSSO STS Administration Guide for examples
JDBC Driver	If JDBC is selected in Connection Type, this field specifies the JDBC driver provided by Oracle Database. Example: oracle.jdbc.driver.OracleDriver. The class specified by JDBC Driver must be accessible to the web container instance on which OpenSSO has been deployed and configured. Include the JAR file that contains the JDBC driver class in the OpenSSO-deploy-base/WEB-INF/lib directory.
JDBC URL	Specify the database URL if JDBC is the selected Connection Type. Example: the URL for Oracle Database is *jdbc:oracle:thin:@hostname:1521/databaseName*.
Connect This User to Database	Specify the username from whom the database connection is made for the JDBC connection.
Password for Connecting to Database	Type the password for the User to Connect to Database.
Password for Connecting to Database (confirm)	Type the password again to confirm it.
Password Column String	Specify the password column name in the SQL database.
Prepared Statement	Specify the SQL statement that retrieves the password of the user that is logging in. For example: select Password from Employees where `USERNAME =?`
Class to Transform Password Syntax	Specify the class name that transforms the password entered by the user for comparison to the password retrieved from the database. This class must implement the `JDBCPasswordSyntaxTranform` interface By default, the value of the attribute is `com.sun.identity.authentication.modules.jdbc.ClearTextTranform` which expects the password to be in clear text.

Table 6–7 (Cont.) JDBC Authentication Service Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

4. Click Save.

6.1.7 To Configure LDAP Authentication Service Realm Attributes

1. On the Configuration tab, click the Authentication subtab.
2. In the Authentication list, click LDAP.
3. Provide values for the LDAP Realm Attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–8 LDAP Authentication Service Realm Attributes

Attributes	Description
Primary LDAP Server	The Current Values list displays the host name and port number of the primary LDAP server specified during OpenSSO STS installation. This is the first server contacted for authentication. If there is no port number, the default value is 389. ■ To add an LDAP server to the list, in the New Value field type the server identifier, and then click Add. Use the form hostname:port. If you have OpenSSO STS deployed with multiple domains, you can specify the communication link between specific instances of OpenSSO STS and Directory Server. Multiple entries must be prefixed by the local server name. Example: *local_servername*|*server:port local_servername2*|*server2:port2 ...* For example, if you have two OpenSSO STS instances deployed in different locations (*L1-machine1-IS* and *L2-machine2-IS*) communicating with different instances of Directory Server (*L1-machine1-DS* and *L2-machine2-DS*), type the following:`L1-machine1-IS.example.com|L1-machine1-DS.example.com:389L2-machine2-IS.example.com|L2-machine2-DS.example.com:389` ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Secondary LDAP Server	The Current Values list displays the host name and port number of a secondary LDAP server available to the OpenSSO STS platform. If the primary LDAP server does not respond to a request for authentication, this secondary server is contacted. When the primary server is up, OpenSSO STS will switch back to the primary server. ■ To add an LDAP server to the list, in the New Value field type the server identifier, and then click Add. Use the format *hostname:port*. Multiple entries must be prefixed by the local server name. Caution – When authenticating users from a Directory Server that is remote from the OpenSSO STS, it is important that both the Primary and Secondary LDAP Server Ports have values. The value for one Directory Server location can be used for both fields. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–8 (Cont.) LDAP Authentication Service Realm Attributes

Attributes	Description
DN to Start User Search	The Current Values lists displays the DN of the node where the search for a user would start. The default value is the root of the directory tree. ■ To add a DN to the list, in the New Value field type the DN, and then click Add. For best performance, use the most specific DN possible. If OBJECT is selected in the Search Scope attribute, then the DN should specify one level above the level in which the profile exists. You can use any valid DN. Multiple entries must be prefixed by the local server name. Example: *servername1* | *search dn servername2* | *search dn servername3* | *search dn...* If multiple entries exist under the root organization with the same user ID, then this parameter should be set so that the only one entry can be searched for or found in order to be authenticated. For example the agent ID and user ID are under the same root org, this parameter should be `ou=Agents` for the root organization to authenticate using `AgentID` and `ou=People`, for the root organization to authenticate using User ID. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
DN for Root User Bind	Specify the DN of the user that will bind as administrator to the Directory Server specified in the Primary LDAP Server and Port field. The authentication service must bind as this DN in order to search for a matching user DN based on the user login ID. The default is amldapuser. You can enter any valid DN.
Password for Root User Bind	Type the password for the administrator profile specified in the DN for Root User Bind field. There is no default value. Only the administrator's valid LDAP password will be recognized.
Password for Root User Bind (confirm	Type the password again to confirm it.
Attribute Used to Retrieve User Profile	Specify the attribute used for the naming convention of user entries. By default, OpenSSO STS identifies user entries by the uid attribute. If your Directory Server uses a different attribute, such as givenname for example, type the attribute name in this field.
Attributes Used to Search for a User to be Authenticated	The Current Values list displays the attributes to be used to form the search filter for finding a user to be authenticated, and allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to `uid`, `employeenumber`, and `mail`, the user could authenticate with any of these attributes. These attributes must be set separately. ■ To add an attribute to the list, in the New Value field type the new attribute, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
User Search Filter	Specify an attribute to use for finding the user under the 'DN to Start User Search' field. This attribute works with the User Naming Attribute. There is no default value. You can enter any valid user entry attribute.

Table 6–8 (Cont.) LDAP Authentication Service Realm Attributes

Attributes	Description
Search Scope	Specify the number of levels in the Directory Server to search for finding a matching user profile. The search begins from the node specified in the 'DN to Start User Search' attribute. The default value is `SUBTREE`. Choose one of the following: `OBJECT` - Searches only the specified node. `ONELEVEL`- Searches the level of the specified node and one level down. `SUBTREE` - Searches all entries at and below the specified node.
SSL Access to LDAP Server	When the OpenSSO STS directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the User ID, and no search is necessary. Normally, an authentication module returns only the User ID, and the authentication service searches for the user in the local OpenSSO STS LDAP. If an external LDAP directory is used, this option is typically not enabled.
Return User DN to Authenticate	When the OpenSSO STS directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the User ID, and no search is necessary. Normally, an authentication module returns only the User ID, and the authentication service searches for the user in the local OpenSSO STS LDAP. If an external LDAP directory is used, this option is typically not enabled
LDAP Server Check Interval	This attribute is used for LDAP Server failback. It defines the number of minutes in which a thread will "sleep" before verifying that the LDAP primary server is running.
User Creation Attributes	The Current Values list displays the attribute-pair used by the LDAP authentication module when the LDAP server is configured as an external LDAP server. ■ To add an attribute-pair to the list, in the New Value field type a string that maps a local Directory Server to an external Directory Server, and then click Add. Use the format *attr1*|*externalattr1* The values of the external attributes are read from the external Directory Server and are set for the internal Directory Server attributes. The values of the external attributes are set in the internal attributes only when the User Profile attribute is set to "Dynamically Created" in the Core Authentication module, and the user does not exist in local Directory Server instance. The newly created user will contain the values for internal attributes, as specified in User Creation Attributes List, with the external attribute values to which they map. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Minimum Password Length	The minimum password length is a value which comes into play when the directory server instance which is being used by the authentication module instance has a password policy to allow the user to reset their password. If the directory server instance returns an LDAP code that the user should reset their password, the new password entered by the user should be equal to or greater than the value of Minimum Password Length.

Table 6–8 (Cont.) LDAP Authentication Service Realm Attributes

Attributes	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

4. Click Save.

6.1.8 To Configure OAMAuth Authentication Service Realm Attributes

1. On the Configuration tab, click the Authentication subtab.
2. In the Authentication list, click OAMAuth.
3. Provide values for the OAMAuth Authentication Service Realm attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–9 OAMAuth Authentication Service Realm Attributes

Attribute	Description
Remote User Header Name	Specify the name of the HTTP header used for an authenticated user. Example `OAM_REMOTE_USER`
Allowed Users Values	The Current Values list displays administrative users who are allowed to access the OpenSSO STS console. ■ To add a user to the list, in the New Value field type a username, and then click Add. ■ To remove an entry from the Current Values list, select the value and then click Remove.

Table 6–9 (Cont.) OAMAuth Authentication Service Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*`://`*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

4. Click Save.

6.1.9 To Configure WSSAuth Authentication Service Attributes

1. On the Configuration tab, click the Authentication subtab.
2. In the Authentication list, click WSSAuth.
3. Provide values for the WSSAuth Authentication Service Realm attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–10 WSSAuth Authentication Service Realm Attributes

Attribute	Description
User search attribute	Specify the user attribute that is used to search for a user. Examples: uid or cn
User realm	Specify the realm that the user belongs to. For OpenSSO STS it is always root realm indicated by a forward slash (/).
User password attribute	Specify the password equivalent for the user. The default could be `userpassword`, it could as well be `employeenumber`, or `mail`.

Table 6–10 (Cont.) WSSAuth Authentication Service Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

4. Click Save.

6.2 Configuring the Authentication Service Realm

A realm is the administrative unit for OpenSSO STS. After OpenSSO STS is deployed and configured, a single top-level realm is created. The Top Level Realm contains all configuration data for the OpenSSO STS instance except for bootstrapping information configured during installation. The Top Level Realm cannot contain subrealms.

Use the OpenSSO STS Authentication tab to specify a supported authentication process, and to instantiate an authentication module used for the entire realm. You can also establish an authentication chain. When you configure multiple authentication module instances to form a chain, a user must pass authentication credentials to all of the module instances specified for the realm.

6.2.1 To Configure the Authentication Realm

1. On the Access Control tab, click the Authentication subtab.
2. In the General section, provide values for the basic Realm properties. The following table provides a listing and descriptions of the properties you can configure.

Table 6–11 Basic Realm Properties

Property	Description
Administrator Authentication Chain	Specify the authentication chain used by administrators when the process must be different from the authentication chain defined for end-users.
Default Success Login URL	The Currents Values list displays URLs that the user will be redirected to upon successful authentication to the realm. ■ To add a new Success Login URL, type a name in the New Value field, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Default Authentication Chain	Specify the default authentication chain used by the realm's users.
Administrator Authentication Chain	Specify the authentication chain used by administrators when the process must be different from the authentication chain defined for end-users.

3. Click Save.
4. To configure additional realm attributes, click Advanced Properties.
5. Provide values for the advanced Realm properties. The following table provides a listing and descriptions of the properties you can configure.

Table 6–12 Advanced Realm Properties

Property	Description
User Profile	Choose a profile status for a successfully authenticated user. ■ Dynamic After successful authentication, the Authentication Service creates a user profile if one does not already exist. The SSOToken will then be issued. The user profile is created in the realm's configured user data store. ■ Dynamic with User Alias After successful authentication, the Authentication Service creates a user profile that contains the User Alias List attribute. This attribute defines one or more aliases for mapping a user's multiple profiles. ■ Ignored A user profile is not required for the Authentication Service to issue an SSOToken after a successful authentication. ■ Required After successful authentication, the user must have a user profile in the realm's configured user data store for the Authentication Service to issue an SSOToken.
Administrator Authentication Configuration	Specify the authentication configuration that is invoked when the user accesses /openssosts/console directly. ■ [empty] ■ ldapService

Table 6–12 (Cont.) Advanced Realm Properties

Property	Description
User Profile Dynamic Creation Default Roles	The Current Values list displays DNs of roles to be assigned to a new user whose profile is created when Dynamic or Dynamic with Alias is selected under the User Profile attribute. There are no default values. A role can be either an OpenSSO STS or LDAP role, but it cannot be a filtered role. ■ To add a new role, in the New Value field type a role name, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Persistent Cookie Mode	Determines whether users can return to their authenticated session after restarting the browser. When enabled, a user session will not expire until its persistent cookie expires as specified by the value of the Persistent Cookie Maximum Time attribute, or the user explicitly logs out. By default, the Authentication Service uses only memory cookies so the session expires when the browser is closed. The client must explicitly request a persistent cookie by appending the iPSPCCookie=yes parameter to the login URL.
Persistent Cookie Maximum Time	Specify the number of seconds after which a persistent cookie expires. The interval begins when the user session is successfully authentication. Persistent cookie mode must be enabled. The field will accept any integer value less than the maximum 214748647.
Alias Search Attribute Name	The Current Values list displays secondary LDAP attributes to use to search for a user profile when a search using the primary LDAP attribute has failed. This attribute is typically used when the user identification returned from an authentication module is not the same as that specified in the User Naming Attribute. ■ To add a new attribute, in the New Values field, type the new attribute name, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove. For example, a Certificate server might return abc1234, but the username is abc. There is no default value for this attribute. The field takes any valid LDAP attribute.
Default Authentication Locale	Specify the default language subtype to be used by the Authentication Service. The default value is `en_US`. To use a difference locale, a directory containing authentication templates for that locale must already exist.
Organization Authentication Configuration	Choose the authentication configuration that is invoked when the user accesses /openssosts/UI/Login. ■ [empty] ■ ldapService
Login Failure Lockout Mode	When enabled, the user is locked out or prevented from authenticating after repeated unsuccessful login attempts within a specified interval. Lockout criteria are defined in the Login Failure Lockout Count and Login Failure Lockout Interval properties below.
Login Failure Lockout Count	Specify the number of times a user can attempt to authenticate within the interval defined in the Login Failure Lockout Interval property. When the user exceeds this number, the user is locked out or prevented from further authentication attempts.

Table 6–12 (Cont.) Advanced Realm Properties

Property	Description
Login Failure Lockout Interval	Specify in minutes the interval during which failed login attempts are counted. The lockout interval begins when a user first attempts to authenticate. The lockout count begins after two consecutive failed logins. The user is locked out if the number of attempts reaches the number specified in the Login Failure Lockout Count. If the user successfully authenticates within the Login Failure Lockout Interval, the lockout count is reset.
Email Address to Send Lockout Notification	Specify an email address or multiple email addresses to which notification will be sent if a user lockout occurs. For multiple addresses, separate each address with a space. For non-English locales, use the following format: *email_address* | *locale* | *charset*
Warn User After N Failures	Specify the number of authentication failures that can occur before OpenSSO STS displays a warning message to the user that the user will be locked out.
Login Failure Lockout Duration	Specify in minutes how long a user must wait after a lockout before attempting to authenticate again. If you enter a value greater than 0, then memory lockout is enabled and physical lockout is disabled. When memory lockout is enabled, the user account is locked in memory for the number of minutes you specified. The account is unlocked after that time has elapsed.
Lockout Duration Multiplier	Specify a value used to multiply the Login Failure Lockout Duration value for each successive lockout. The Lockout Duration is increased incrementally based on the number of times the user has been locked out. For example, if the Login Failure Lockout Duration is set to 3 minutes, and the Lockout Duration Multiplier is to 2, then the user will be locked out of the account for 6 minutes. After the 6 minutes has elapsed, if the user again provides the wrong credentials, the lockout duration is now 12 minutes.
Lockout Attribute Name	Specify the LDAP attribute to be used for physical lockout. The default value is inetuserstatus even when the field is empty. The Lockout Attribute Value field must also contain an appropriate value.
Lockout Attribute Value	Specify the action to be taken on the attribute defined in the Lockout Attribute Name. The default value is inactive even if the field is empty. The Lockout Attribute Name field must also contain an appropriate value.
Default Success login URL	The Current Values list displays URLs where users are directed after successful authentication. ■ To add a new URL, in the New Value field type the URL, and then click Add. Use the form client-type | URL. The only value you can specify at this time is a URL which assumes the type HTML. The default value is /opensso/console. Values that don't specify HTTP or HTTP(S) are appended to the deployment URL. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–12 (Cont.) Advanced Realm Properties

Property	Description
Default Failure Login URL	The Current Values list displays URLs where users are directed after a failed authentication attempt. ■ To add a new URL, in the New Value field type the URL, and then click Add. Use the form client-type|URL. The only value you can specify at this time is a URL which assumes the type HTML. The default value is /opensso/console. Values that don't specify HTTP or HTTP(S) are appended to the deployment URL. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Authentication Post Processing Class	The Current Values list displays a Java class or multiple Java classes to be used for customizing post-authentication processes for either successful or unsuccessful logins. ■ To add a new Java class, in the New Value field type the class name, and then click Add. Example: com.abc.authentication.PostProcessClass The Java class must implement the interface com.sun.identity.authentication.spi. AMPostAuthProcessInterfaceOpenSSOEnterprise. Additionally, a JAR containing the post-processing class must be added to the classpath of the web container instance on which OpenSSO STS is configured. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Generate UserID Mode	When enabled, if the user identifier entered by a user during the self-registration process is not valid or already exists, the Membership module generates a list of alternate user identifiers. The user identifiers are generated by the class specified in the Pluggable User Name Generator Class property.
Pluggable User Name Generator Class	Specify the name of the class to be used for generating alternate user identifiers when Generate UserID Mode is enabled. The default value is `com.sun.identity.authentication.spi.DefaultUserIDGenerator.`
Identity Types	Click a box to mark the type of identity or types of identities for which OpenSSO STS will search.
Pluggable User Status Event Classes.	The Current Values list displays the Java class or Java classes used to provide a callback mechanism for user status changes during the authentication process. ■ To add a Java new class, in the New Value field type the Java class name, and then click Add. Example: `com.abc.authentication.PostProcessClass` The Java class must implement the OpenSSO STS interface com.sun.identity.authentication.spi.AMAuthCallBack. Account lockout and password changes are supported. Password changes are supported through the LDAP authentication module. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–12 (Cont.) Advanced Realm Properties

Property	Description
Store Invalid Attempts in Data Store	When enabled, information regarding failed authentication attempts is stored as the value of the sunAMAuthInvalidAttemptsData attribute in the user data store. To store data in this attribute, the OpenSSO STS schema must be loaded. Information stored includes the number of invalid attempts, time of last failed attempt, lockout time, and lockout duration. Storing this information in the identity repository allows the information to be shared among multiple instances of OpenSSO STS.
Module Based Authentication	When enabled, users authenticate using module-based authentication. When disabled, all attempts at authentication using the `module=module-instance-name` login parameter will fail.
Use Attribute Mapping to Session Attribute	The Current Values list displays user identity attributes that are mapped as session properties in the user's SSOToken. ■ To add a new attribute mapping, in the New Value field type a new attribute-value pair, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove. Use the form *User-Profile-Attribute*|*Session-Attribute-Name*. If *Session-Attribute-Name* is not specified, the value of *User-Profile-Attribute* is used. All session attributes contain the am.protected prefix to ensure that they cannot be edited by the Client SDK.
Default Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*/`opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*/`UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6. Click Save.

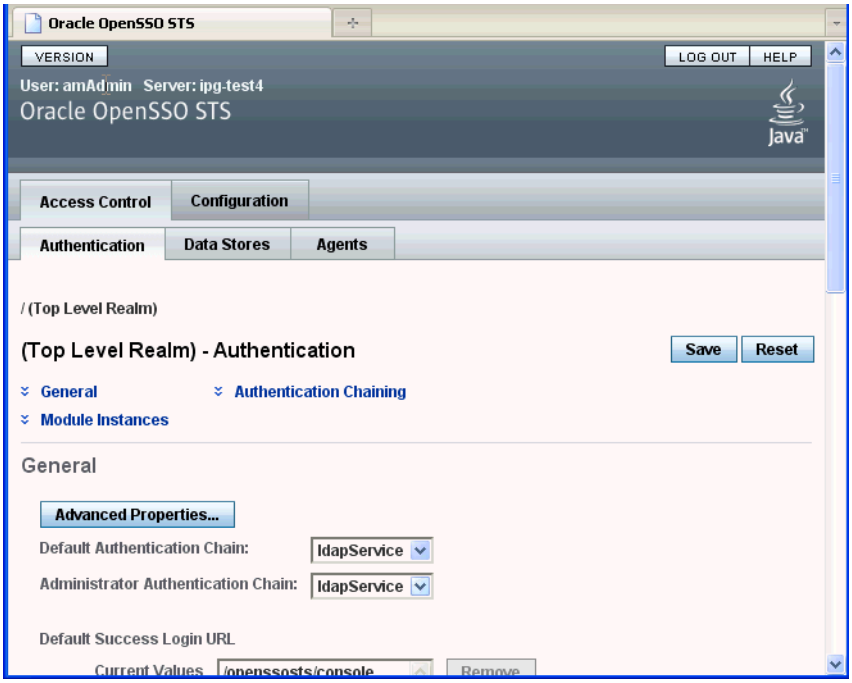
6.3 Managing Authentication Module Instances

OpenSSO STS is installed with a set of default authentication module instance types. An authentication module instance is a plug-in that collects user information such as a user ID and password, checks the information against entries in a database, and allows or denies access to the user. You can create new authentication module instances. You

can also create multiple authentication instances of the same type, which you can configure separately.

The following figure illustrates the Access > Authentication subtab the OpenSSO STS console where you create and configure new authentication module instances.

Figure 6–2 Access Control > Authentication Subtab for Configuring Individual Authentication Modules



The following authentication module types (Module Instances) are supported in OpenSSO STS.

Active Directory Authentication Module

An Active Directory authentication module works similarly to the LDAP authentication module type, but uses the Microsoft Active Directory instead of an LDAP directory. Using this module type makes it possible to have both LDAP and Active Directory coexist under the same realm. See Section 6.3.1, "To Add a New Active Directory Module Instance" and Section 6.3.2, "To Configure an Active Directory Authentication Module Instance."

Certificate Authentication Module

A Certificate authentication module enables a user to log in through a personal digital certificate (PDC). The module instance can require the use of the Online Certificate Status Protocol (OCSP) to determine the state of a certificate. Use of the OCSP is optional. The user is granted or denied access to a resource based on whether or not the certificate is valid. See Section 6.3.3, "To Add a New Certificate Authentication Module Instance" and Section 6.3.4, "To Configure a Certificate Authentication Module Instance."

Data Store Authentication Module

A Data Store authentication modules enables OpenSSO STS to authenticate users against the Identity Repository. Using the Data Store module removes the requirement to write an authentication plug- in module, load, and then configure the authentication

module if you need to authenticate against the same data store repository. Additionally, you do not need to write a custom authentication module where flat-file authentication is needed for the corresponding repository in that realm. See Section 6.3.5, "To Add a New Data Store Authentication Module Instance" and Section 6.3.6, "To Configure a Data Store Authentication Module Instance."

Federation Authentication Module
The Federation authentication module authenticates and validates Federation protocols. For example, when an Identity Provider sends a SAML2 artifact profile or post profile to a Service Provider, the Federation authentication module retrieves the SAML2 assertion and validates the assertion against the Identity Provider server. See Section 6.3.7, "To Add and Configure a New Federation Authentication Module Instance."

JDBC Authentication Module
A Java Database Connectivity (JDBC) authentication module enables OpenSSO STS to authenticate users through any Structured Query Language (SQL) databases that provide JDBC-enabled drivers. The connection to the SQL database can be either directly through a JDBC driver or through a JNDI connection pool. The JDBC attributes are realm attributes. See Section 6.3.8, "To Add a New JDBC Authentication Module Instance" and Section 6.3.9, "To Configure a JDBC Authentication Module Instance."

LDAP Authentication Module
An LDAP authentication module enables OpenSSO STS to authenticate users using LDAP bind, a Directory Server operation which associates a user ID password with a particular LDAP entry. OpenSSO validates the UserName with a cleartext password contained in a web service request to a web service provider. You can define multiple LDAP authentication configurations for a realm. See Section 6.3.10, "To Add an New LDAP Authentication Module Instance" and Section 6.3.11, "To Configure an LDAP Authentication Module Instance."x

Oracle Authentication Module
The Oracle authentication module enables OpenSSO STS to authenticate and single sign-on an administrator, who previously authenticated to Oracle Access Manager, to OpenSSO STS. The administrator does not have to provide credentials OpenSSO STS.

See Section 6.3.12, "To Add a New Oracle Authentication Module Instance" and Section 6.3.13, "To Configure an Oracle Authentication Module Instance".

Web Service Security Authentication Module
The Web Service Security authentication module enables OpenSSO STS to validate the UserName with a digest password received as an authentication token and contained in a service request from the web service client to a web service provider.

See Section 6.3.16, "To Delete an Authentication Module Instance" and Section 6.3.15, "To Configure a WSSAuth Authentication Module Instance."

6.3.1 To Add a New Active Directory Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click New.
3. In the Name field, type a name for this Active Directory module instance.

   The name cannot contain spaces.
4. For Type, choose Active Directory.

5. Click OK.
6. Configure the Active Directory Authentication Module Instance. See Section 6.3.2, "To Configure an Active Directory Authentication Module Instance."

6.3.2 To Configure an Active Directory Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click name of the Active Directory Authentication module instance you want to configure.
3. Provide values for the Active Directory Authentication Module Instance Realm attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–13 Active Directory Authentication Module Instance Realm Attributes

Attribute	Description

Primary Active Directory Server	The Current Values list displays the host name and port number of the primary Active Directory server specified during OpenSSO STS installation. This is the first server contacted for Active Directory authentication. The format is hostname:port. The default port number is 389.
	■ To add a new Active Directory server to the list, then click Add.
	If you have OpenSSO STS deployed with multiple domains, you can specify the communication link between specific instances of OpenSSO STS and Directory Server in using the form *LocalServerName* | *Server:PortNumber*. For multiple entries, each entry must be prefixed with a local server name. Example:
	local_servername | *server:port local_servername2* | *server2:port2...*
	For example, if you have two OpenSSO STS instances deployed in different locations (*L1-machine1-IS* and *L2-machine2-IS*) communicating with different instances of Directory Server (*L1-machine1-DS* and *L2-machine2-DS*), use the form:
	`L1-machine1-IS.example.com|L1-machine1-DS.example.com:389`
	`L2-machine2-IS.example.com|L2-machine2-DS.example.com:389`
	■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–13 (Cont.) Active Directory Authentication Module Instance Realm Attributes

Attribute	Description
Secondary Active Directory Server	The Current Values list displays the host name and port number of a secondary Active Directory server available to the OpenSSO STS platform. If the primary Active Directory server does not respond to a request for authentication, then this server is contacted. If the primary server is up, OpenSSO STS will switch back to the primary server. ■ To add an Active Directory server to the list, in the New Value field Type the name of the new server, and then click Add. Use the form hostname:port. Multiple entries must be prefixed by the local server name. Caution – When authenticating users from a Directory Server that is remote from the OpenSSO STS server, both the Primary and Secondary LDAP Server Ports must have values. The value for one Directory Server location can be used for both fields. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
DN to Start User Search	The Current Values list displays the DN of the node where the search for a user starts. ■ To add a new base DN to the list, in the New Value field Type the new DN, and then click Add. Use the form servername | searchDN. For performance reasons, this DN should be as specific as possible. The default value is the root of the directory tree. Any valid DN will be recognized. If OBJECT is selected in the Search Scope attribute, the DN should specify one level above the level in which the profile exists. Multiple entries must be prefixed by the local server name. Example: *servername1* | *searchDN* *servername2* | *searchDN* *servername3* | *searchDN...* If multiple entries exist under the root organization with the same user ID, then this parameter should be set so that only one entry can be searched for or found in order to be authenticated. For example, in the case where the agent ID and user ID are under the same root org, this parameter should be `ou=Agents` for the root organization to authenticate using `AgentID` and `ou=People,` for the root organization to authenticate using User ID. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
DN for Root User Bind	Specify the DN of the user that will be used to bind to the Directory Server specified in the Primary LDAP Server and Port fields as administrator. The authentication service must bind as this DN in order to search for a matching user DN based on the user login ID. The default is amldapuser. Any valid DN will be recognized. Make sure that password is correct before you logout. If it is incorrect, you will be locked out. If this should occur, you can login with the super user DN. By default, this the amAdmin account with which you would normally log in, although you will use the full DN. For example: `uid_amAdmin,ou=People, OpenSSO-deploy-base`

Table 6–13 (Cont.) Active Directory Authentication Module Instance Realm Attributes

Attribute	Description
Password for Root User Bind	Type the password for the administrator profile specified in the DN for Root User Bind field. There is no default value. Only the administrator's valid Active Directory password is recognized.
Password for Root User Bind (confirm)	Type the Root User Bind password again to confirm it.
Attribute Used to Retrieve User Profile	Specify the attribute used for the user entry naming convention. By default, OpenSSO STS assumes that user entries are identified by the uid attribute. If your Directory Server uses a different attribute such as givenname, specify the attribute name in this field.
Attributes Used to Search for a User to be Authenticated	The Current Values list displays the attributes to be used to form the search filter for a user that is to be authenticated, and that allows the user to authenticate with more than one attribute in the user's entry. For example, if this field is set to uid, employeenumber, and mail, then the user could authenticate with any of these names. ■ To add an attribute to the list, in the New Value field Type the attribute, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
User Search Filter	Displays the attributes to be used to find the user based on the value in the DN to Start User Search field. The filter works with the User Naming Attribute. There is no default value. Any valid user entry attribute will be recognized.
Search Scope	Choose the number of levels in the Directory Server that will be searched for a matching user profile. The search begins from the node specified in DN to Start User Search field. The default value is `SUBTREE`. Choose one of the following: ■ `OBJECT` searches only the specified node. ■ `ONELEVEL` searches at the level of the specified node and one level down. ■ `SUBTREE` searches all entries at and below the specified node.
SSL Access to Active Directory Server	When enabled, OpenSSO STS uses the SSL protocol to access the Directory Server specified in the Primary and Secondary Server and Port fields. By default, the box is not checked and the SSL protocol is not used to access the Directory Server. If the Active Directory server is running with SSL enabled (LDAPS), you must make sure that OpenSSO STS is configured with proper SSL trusted certificates. Otherwise OpenSSO STS cannot connect to Directory Server using the LDAPS protocol.
Return User DN to Authenticate	When enabled, the Active Directory authentication module instance returns the DN instead of the User ID, and no search is necessary. Normally, an authentication module instance returns only the User ID, and the authentication service searches for the user in the local OpenSSO STS instance. If the OpenSSO STS directory is the same as the directory configured for Active Directory, this option may be enabled. If an external Active Directory is used, this option is typically not enabled.

Table 6–13 (Cont.) Active Directory Authentication Module Instance Realm Attributes

Attribute	Description
Active Directory Server Check Interval	Specify the number of minutes per interval in which a thread will "sleep" before verifying that the primary Active Directory server is running. This attribute is used for Active Directory Server failback.
User Creation Attributes	The Current Values list displays attributes used by the Active Directory authentication module instance when the Active Directory server is configured as an external Active Directory server. It contains a mapping of attributes between a local and an external Directory Server. The attribute uses the following form: *attr1*|*externalattr1 attr2*|*externalattr2* ■ To add a new attribute, in the New Value field Type the attribute and then click Add. Use the form: *attr1*|*externalattr1 attr2*|*externalattr2* When this attribute is populated, the values of the external attributes are read from the external Directory Server, and are set for the internal Directory Server attributes. The values of the external attributes are set in the internal attributes only when the User Profileattribute (in the Core Authentication module type) is set to Dynamically Created and the user does not exist in local Directory Server instance. The newly created user will contain the values for internal attributes, as specified in User Creation Attributes List, with the external attribute values to which they map. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Minimum Password Length	The minimum password length is a value which comes into play when the directory server instance which is being used by the authentication module instance has a password policy to allow the user to reset their password. If the directory server instance returns an LDAP code that the user should reset their password, the new password entered by the user should be equal to or greater than the value of Minimum Password Length.

Table 6–13 (Cont.) Active Directory Authentication Module Instance Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6.3.3 To Add a New Certificate Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click New.
3. In the Name field, type a name for this Certificate authentication module instance.
4. For Type, choose Certificate.
5. Click OK.
6. Configure the Certificate Authentication Module Instance. See Section 6.3.4, "To Configure a Certificate Authentication Module Instance."

6.3.4 To Configure a Certificate Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click name of the Certificate authentication module instance you want to configure.
3. Provide values for the Certificate Authentication Module Instance Realm attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–14 Certificate Authentication Module Instance Realm Attributes

Attribute	Description
Match Certificate in LDAP	When enabled, the Certificate Authentication Module determines whether a user certificate presented at login is stored in the LDAP Server specified in the "LDAP Server Where Certificates are Stored" field below. If no match is found, then the user is denied access. If a match is found and no other validation is required, the user is granted access. By default, this option is disabled and the Certificate Authentication Module does not check for the user certificate. Note – A certificate stored in the Directory Server is not necessarily valid. It may be on the certificate revocation list. See Match Certificate to CRL. However, the web container may check the validity of the user certificate presented at login.
Subject DN Attribute Used to Search LDAP for Certificates	Specify the attribute of the certificate's SubjectDN value that will be used to search LDAP for certificates. This attribute must uniquely identify a user entry. The actual value will be used for the search. The default is `cn`.
Match Certificate to CRL	When enabled, the Certificate Authentication Module compares the user certificate against the Certificate Revocation List (CRL) in the LDAP Server. The CRL is located based on one of the attribute names in the issuer's `SubjectDN`. If the certificate is on the CRL, then the user is denied access. If the certificate is not on the CRL, then the user is allowed to proceed. By default this option is disabled. Certificates should be revoked when the owner of the certificate has changed status and no longer has the right to use the certificate or when the private key of a certificate owner has been compromised.
Issuer DN Attribute Used to Search LDAP for CRLs	Specify the attribute of the `subjectDN` for the certificate's issuer. The subjectDN value will be used to search LDAP for CRLs. This field is used only when the Match Certificate to CRL attribute is enabled. The actual value will be used for the search. The default is cn.
HTTP Parameters for CRL Update	Specify the HTTP parameters for obtaining a CRL from a servlet for a CRL update. Contact the administrator of your CA for these parameters.
Match CA Certificate to CRL	When enabled, the Certificate Authentication Module compares the Certificate Authority-issued certificate against the Certificate Revocation List (CRL) in the LDAP Server. The CRL is located based on one of the attribute names in the issuer's SubjectDN. If the certificate is on the CRL, then the user is denied access. If the certificate is not on the CRL, then the user is allowed to proceed. By default this option is disabled. Certificates should be revoked when the owner of the certificate has changed status and no longer has the right to use the certificate or when the private key of a certificate owner has been compromised.

Table 6–14 (Cont.) Certificate Authentication Module Instance Realm Attributes

Attribute	Description
OCSP Validation	When enabled, Online Certificate Status Protocol (OCSP) validation is performed by contacting the appropriate OCSP responder. The OCSP responder is determined during runtime based on the following settings: ■ If this value is set to true, and the OCSP responder is set in the Responder URL attribute, then the value of the attribute will be used as the OCSP responder. ■ If Online Certificate Status Protocol Check is enabled and if the value of this attribute is not set, then the OCSP responder presented in your client certificate is used as the OCSP responder. ■ If Online Certificate Status Protocol Check is not enabled, or if Online Certificate Status Protocol Check is enabled but an OCSP responder can not be found, then no OCSP validation will be performed. These settings can be configured on the Servers and Sites tab. Before enabling OCSP Validation, make sure that the time of day settings for the OpenSSO STS host and the OCSP responder host are synchronized as closely as possible. Also, the time of day setting for the OpenSSO STS host must be ahead of the time of day setting for the OCSP responder. For example, if the OCSP responder host is set at 12:00:00 PM, then the OpenSSO STS host could be set at 12:00:30 PM.
LDAP Server Where Certificates are Stored	The Current Values list displays the name and port number of the LDAP server where the certificates are stored. The default value is the host name and port specified when OpenSSO STS was installed. ■ To add a new LDAP server, in the New Value field type the server identifier, and then click Add. Use the form hostname:port. You can specify any LDAP server where the certificates are stored. When entering multiple entries, each entry must be prefixed with a local server name. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–14 (Cont.) Certificate Authentication Module Instance Realm Attributes

Attribute	Description
LDAP Search Start DN	The Current Values list displays the DN of the node where the search for the user's certificate should start. ■ To add a DN to the list, in the New Value field type the new DN, and then click Add. Use the format *servername*|*searchDN*. There is no default value. You can enter any valid DN. Multiple entries must be prefixed by the local server name. Example: *servername1*|*searchDN servername2*|*searchDN servername3*|*searchDN* If multiple entries exist under the root organization with the same user ID, then this parameter should be set so that the only one entry can be searched for or found in order to be authenticated. For example, in the case where the agent ID and user ID is same under root org, this parameter should be `ou=Agents` for the root organization to authenticate using `AgentID` and `ou=People,` for the root organization to authenticate using User ID. ■ To remove an entry from the Current Values list, select the entry and them click Remove. Use the format *servername*|*searchDN*. There is no default value. You can enter any valid DN. Multiple entries must be prefixed by the local server name. Example: *servername1*|*searchDN servername2*|*searchDN servername3*|*searchDN*
LDAP Server Principal User	Specify the DN of the principal user for the LDAP server where the certificates are stored. There is no default value. You can use any valid DN. The principal user must be authorized to read, and search certificate information stored in the Directory Server.
LDAP Server Principal Password	Specify the LDAP password associated with the user specified in the LDAP Server Principal User field above. There is no default value. You can use any valid LDAP password for the specified principal user. This value is stored as readable text in the directory.
LDAP Server Principal Password (confirm	Type the password again to confirm it.
Use SSL for LDAP Access	Specifies whether to use SSL to access the LDAP server. The default is that the Certificate Authentication service does not use SSL for LDAP access.

Certificate Field Used to Access User Profile	From the following, choose the field in the certificate's Subject DN to be used to search for a matching user profile:
	`email address`
	`none`
	`other`
	`subject CN`
	`subject DN`
	`subject UID`
	For example, if you choose email address, the Certificate Authentication service searches for the user profile that matches the attribute `emailAddr` in the user certificate. The user logging in then uses the matched profile. The default field is subject CN.

Table 6–14 (Cont.) Certificate Authentication Module Instance Realm Attributes

Attribute	Description
Other Certificate Field Used to Access User Profile	This attribute is recognized only if 'other' is selected in the 'Certificate Field Used to Access User Profile' attribute above. Specify the attribute that will be selected from the received certificate's subjectDN value. The Certificate Authentication service will then search the user profile that matches the value of that attribute.
SubjectAltNameExt Value Type to Access User Profile	RFC822Name - Electronic email address UPN - User Principal Name none When 'none' is selected, the 'Certificate Field Used to Access User Profile' or 'Other Certificate Field Used to Access User Profile' attribute is used to access the User Profile.
Trusted Remote Hosts	The Current Values list displays hosts that can be trusted to send certificates to OpenSSO STS. OpenSSO STS must verify whether the certificate came from one of these hosts. This attribute is used for the Portal Server gateway, for a load balancer with SSL termination and for Distributed Authentication. By default, this attribute is set to 'none,' which disables certificate issuer host verification. ■ To add a host to this list, in the New Value field type one of the following, and then click Add. none - Disables certificate issuer host verification. This is set by default. all - Accepts Portal Server Gateway-style certificate authentication from any client IP address. IP ADDR -Lists the IP addresses from which to accept Portal Server Gateway-style certificate authentication requests (the IP Address of the Gateway(s)). The attribute is configurable on an realm basis. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
SSL Port Number	Specify the port number for the secure socket layer (SSL). Currently, this attribute is only used by the Gateway servlet. Before you add or change an SSL Port Number, see the "Policy-Based Resource Management" section in the OpenSSO STS Administration Guide.
HTTP Header Name for Client Certificate	This attribute is used only when the Trusted Remote Hosts attribute is set to all' or has a specific host name defined. Specify the HTTP header name for the client certificate that is inserted by the load balancer or Secure Remote Access component.

Table 6–14 (Cont.) Certificate Authentication Module Instance Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6.3.5 To Add a New Data Store Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click New.
3. In the Name field, type a name for this Data Store authentication module instance.
4. For Type, choose Data Store.
5. Click OK.
6. Configure the Data Store authentication module instance. Section 6.3.6, "To Configure a Data Store Authentication Module Instance."

6.3.6 To Configure a Data Store Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click name of the Data Store authentication module instance you want to configure.
3. Provide values for the Data Store Authentication Module Instance Realm attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–15 Data Store Authentication Module Instance Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6.3.7 To Add and Configure a New Federation Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click New.
3. In the Name field, type a name for this Federation Authentication module instance.
4. For Type, choose Federation
5. Click OK.
6. Configure the Federation authentication module instance.
   a. On the Access > Authentication subtab, in the Module Instances section, select the Federation instance you want to configure.
   b. On the Federation Realm Attributes page, type a value in the Authentication Level field.

      Specify a value that indicates how much to trust the Federation Authentication module instance.

      Once a user has authenticated, this value is stored in the SSO token for the session. When the SSO token is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSO token does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. The default value is 0.

0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*/opensso/UI/Loin?authlevel=0, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*/UI/Loin?authlevel=50, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed.

If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

c. Click Save.

7. Click Back to Authentication.

6.3.8 To Add a New JDBC Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click New.
3. In the Name field, type a name for this JDBC authentication module instance.
4. For Type, choose JDBC.
5. Click OK.
6. Configure the JDBC authentication module instance. Section 6.3.9, "To Configure a JDBC Authentication Module Instance."

6.3.9 To Configure a JDBC Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click name of the JDBC authentication module instance you want to configure.
3. Provide values for the JDBC Authentication Module Instance Realm attributes. The following table provides a listing and descriptions of the attributes you can configure.

Table 6–16 JDBC Authentication Module Instance Realm Attributes

Attribute	Description
Connection Type	Choose the type of connection to be made to the SQL database. ▪ Connection pool is retrieved via JNDI The Java Naming and Directory Interface (JNDI) connection pool uses the configuration from the underlying web container. ▪ Non-persistent JDBC connection. The Java Database Connectivity (JDBC) API provides a call-level API for SQL-based database access.

Table 6–16 (Cont.) JDBC Authentication Module Instance Realm Attributes

Attribute	Description
Connection Pool JNDI Name	If JNDI is selected in Connection Type, this field specifies the connection pool name. Because JDBC authentication uses the JNDI connection pool provided by the web container, the setup of JNDI connection pool may not be consistent among other web containers. See the OpenSSO STS Administration Guide for examples
JDBC Driver	If JDBC is selected in Connection Type, this field specifies the Oracle driver provided by the Oracle Database. Example: oracle.jdbc.driver.OracleDriver. The class specified by Oracle Driver must be accessible to the web container instance on which OpenSSO has been deployed and configured. Include the JAR file that contains the Oracle driver class in the OpenSSO-deploy-base/WEB-INF/lib directory.
JDBC URL	Specify the database URL if JDBC is the selected Connection Type. Example: the URL for Oracle Database is *jdbc:oracle:thin:@hostname:1521/databaseName*.
Connect This User to Database	Specify the username from whom the database connection is made for the JDBC connection.
Password for Connecting to Database	Type the password for the User to Connect to Database.
Password for Connecting to Database (confirm)	Type the password again to confirm it.
Password Column String	Specify the password column name in the SQL database.
Prepared Statement	Specify the SQL statement that retrieves the password of the user that is logging in. For example: select Password from Employees where `USERNAME =?`
Class to Transform Password Syntax	Specify the class that transforms the password entered by the user for comparison to the password retrieved from the database. This class must implement the `JDBCPasswordSyntaxTranform interface` By default, the value of the attribute is com.sun.identity.authentication.modules.jdbc.ClearTextTranform which expects the password to be in clear text.

Table 6–16 (Cont.) JDBC Authentication Module Instance Realm Attributes

Attribute	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*`://`*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6.3.10 To Add an New LDAP Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click New.
3. In the Name field, type a name for this LDAP authentication module instance.
4. For Type, choose LDAP.
5. Click OK.
6. Configure the LDAP authentication module instance. Section 6.3.11, "To Configure an LDAP Authentication Module Instance."

6.3.11 To Configure an LDAP Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click name of the LDAP authentication module instance you want to configure.
3. Provide values for the LDAP Authentication Module Instance Realm attributes. The following table provides a listing and descriptions of the properties you can configure.

Table 6–17 LDAP Authentication Module Instance Realm Attributes

Attributes	Description
Primary LDAP Server	The Current Values list displays the host name and port number of the primary LDAP server specified during OpenSSO STS installation. This is the first server contacted for authentication. If there is no port number, the default value is 389. ■ To add an LDAP server to the list, in the New Value field type the server identifier, and then click Add. Use the form hostname:port. If you have OpenSSO STS deployed with multiple domains, you can specify the communication link between specific instances of OpenSSO STS and Directory Server. Multiple entries must be prefixed by the local server name. Example: *local_servername*|*server:port local_servername2*|*server2:port2* Example: if you have two OpenSSO STS instances deployed in different locations (*L1-machine1-IS* and *L2- machine2-IS)* communicating with different instances of Directory Server (*L1-machine1-DS* and *L2-machine2-DS)*, type the following:`L1-machine1-IS.example.com|L1-machine1-DS.example.com:389L2-machine2-IS.example.com|L2-machine2-DS.example.com:389` ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Secondary LDAP Server	The Current Values list displays the host name and port number of a secondary LDAP server available to the OpenSSO STS platform. If the primary LDAP server does not respond to a request for authentication, this secondary server is contacted. When the primary server is up, OpenSSO STS will switch back to the primary server. ■ To add an LDAP server to the list, in the New Value field type the server identifier, and then click Add. Use the format hostname:port. Multiple entries must be prefixed by the local server name. Caution – When authenticating users from a Directory Server that is remote from the OpenSSO STS, it is important that both the Primary and Secondary LDAP Server Ports have values. The value for one Directory Server location can be used for both fields. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

Table 6–17 (Cont.) LDAP Authentication Module Instance Realm Attributes

Attributes	Description
DN to Start User Search	The Current Values lists displays the DN of the node where the search for a user would start. The default value is the root of the directory tree. ■ To add a DN to the list, in the New Value field type the DN, and then click Add. For best performance, use the most specific DN possible. If OBJECT is selected in the Search Scope attribute, then the DN should specify one level above the level in which the profile exists. You can use any valid DN. Multiple entries must be prefixed by the local server name. Example: *servername1* | *search dn servername2* | *search dn servername3* | *search dn...* If multiple entries exist under the root organization with the same user ID, then this parameter should be set so that the only one entry can be searched for or found in order to be authenticated. For example the agent ID and user ID are under the same root org, this parameter should be `ou=Agents` for the root organization to authenticate using `AgentID` and `ou=People`, for the root organization to authenticate using User ID. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
DN for Root User Bind	Specify the DN of the user that will bind as administrator to the Directory Server specified in the Primary LDAP Server and Port field. The authentication service must bind as this DN in order to search for a matching user DN based on the user login ID. The default is amldapuser. You can enter any valid DN.
Password for Root User Bind	Type the password for the administrator profile specified in the DN for Root User Bind field. There is no default value. Only the administrator's valid LDAP password will be recognized.
Password for Root User Bind (confirm	Type the password again to confirm it.
Attribute Used to Retrieve User Profile	Specify the attribute used for the naming convention of user entries. By default, OpenSSO STS identifies user entries by the uid attribute. If your Directory Server uses a different attribute, such as givenname for example, type the attribute name in this field.
Attributes Used to Search for a User to be Authenticated	The Current Values list displays the attributes to be used to form the search filter for finding a user to be authenticated, and allows the user to authenticate with more than one attribute in the user's entry. Example: if this field is set to uid, employeenumber, and mail, the user could authenticate with any of these attributes. These attributes must be set separately. ■ To add an attribute to the list, in the New Value field type the new attribute, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
User Search Filter	Specify an attribute to use for finding the user under the 'DN to Start User Search' field. This attribute works with the User Naming Attribute. There is no default value. You can enter any valid user entry attribute.

Table 6–17 (Cont.) LDAP Authentication Module Instance Realm Attributes

Attributes	Description
Search Scope	Specify the number of levels in the Directory Server to search for finding a matching user profile. The search begins from the node specified in the 'DN to Start User Search' attribute. The default value is `SUBTREE`. Choose one of the following: `OBJECT` - Searches only the specified node. `ONELEVEL`- Searches the level of the specified node and one level down. `SUBTREE` - Searches all entries at and below the specified node.
SSL Access to LDAP Server	When the OpenSSO STS directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the User ID, and no search is necessary. Normally, an authentication module returns only the User ID, and the authentication service searches for the user in the local OpenSSO STS LDAP. If an external LDAP directory is used, this option is typically not enabled.
Return User DN to Authenticate	When the OpenSSO STS directory is the same as the directory configured for LDAP, this option may be enabled. If enabled, this option allows the LDAP authentication module to return the DN instead of the User ID, and no search is necessary. Normally, an authentication module returns only the User ID, and the authentication service searches for the user in the local OpenSSO STS LDAP. If an external LDAP directory is used, this option is typically not enabled
LDAP Server Check Interval	This attribute is used for LDAP Server failback. It defines the number of minutes in which a thread will "sleep" before verifying that the LDAP primary server is running.
User Creation Attributes	The Current Values list displays the attribute-pair used by the LDAP authentication module when the LDAP server is configured as an external LDAP server. ■ To add an attribute-pair to the list, in the New Value field type a string that maps a local Directory Server to an external Directory Server, and then click Add. Use the format *attr1*|*externalattr1* The values of the external attributes are read from the external Directory Server and are set for the internal Directory Server attributes. The values of the external attributes are set in the internal attributes only when the User Profile attribute is set to "Dynamically Created" in the Core Authentication module, and the user does not exist in local Directory Server instance. The newly created user will contain the values for internal attributes, as specified in User Creation Attributes List, with the external attribute values to which they map. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Minimum Password Length	The minimum password length is a value which comes into play when the directory server instance which is being used by the authentication module instance has a password policy to allow the user to reset their password. If the directory server instance returns an LDAP code that the user should reset their password, the new password entered by the user should be equal to or greater than the value of Minimum Password Length.

Table 6–17 (Cont.) LDAP Authentication Module Instance Realm Attributes

Attributes	Description
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6.3.12 To Add a New Oracle Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click New.
3. In the Name field, type a name for this Oracle authentication module instance.
4. For Type, choose OAMAuth.
5. Click OK.
6. Configure the OAMAuth authentication module instance. Section 6.3.13, "To Configure an Oracle Authentication Module Instance."

6.3.13 To Configure an Oracle Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click name of the OAMAuth authentication module instance you want to configure.
3. Provide values for the Oracle Authentication Module Instance Realm attributes. The following table provides a listing and descriptions of the attributes you can configure.

Table 6–18 Oracle Authentication Module Instance Realm Attributes

Attribute	Description
Remote User Header Name	Specify the name of the HTTP header used for an authenticated user. Example `OAM_REMOTE_USER`

Table 6–18 (Cont.) Oracle Authentication Module Instance Realm Attributes

Attribute	Description
Allowed Users Values	The Current Values list displays administrative users who are allowed to access the OpenSSO STS console. ■ To add a user to the list, in the New Value field type a username, and then click Add. ■ To remove an entry from the Current Values list, select the value and then click Remove.
Authentication level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*://*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6.3.14 To Add a New Web Service Security Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click New.
3. In the Name field, type a name for this WSSAuth authentication module instance.
4. For Type, choose WSSAuth.
5. Click OK.
6. Configure the WSSAuth authentication module instance.

6.3.15 To Configure a WSSAuth Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, click name of the WSSAuth authentication module instance you want to configure.
3. Provide values for the WSSAuth Authentication Module Instance Realm attributes. The following table provides a listing and descriptions of the attributes you can configure.

Table 6–19 WSSAuth Authentication Module Instance Realm Attributes

Attribute	Description
User search attribute	Specify a user attribute that to be used to search for a user. Examples: `uid, cn`
User realm	Specify the realm the user belongs to. For OpenSSO STS it is always root realm, indicated by a forward slash (`/`) .
User password attribute	Specify a password attribute (password equivalent) for the user. The default could be `userpassword`, it could as well be `employeenumber` or `mail`.
Authentication Level	Specify a value that indicates how much to trust an authentication mechanism. The default value is 0. The authentication level is set separately for each method of authentication. Once a user has authenticated, this value is stored in the `SSOToken` for the session. When the SSOToken is presented to an application the user wants to access, the application uses the stored value to determine whether the level is sufficient to grant the user access. If the authentication level stored in an SSOToken does not meet the minimum value required, the application can prompt the user to authenticate again through a service with a higher authentication level. 0 is a low value. For example, if the user accesses the URL *protocol*`://`*openssoServer:openssoPort*`/opensso/UI/Loin?authlevel=0`, a selection menu is displayed containing all authentication module instances with an authentication level of 0 or greater, or all authentication module instances. Similarly if the user accesses the URL *protocol://openssoServer:port/opensso*`/UI/Loin?authlevel=50`, a selection menu is displayed containing authentication module instances with an authentication level of 50 or greater. Or if only one authentication module instance meets that constraint, a login screen for that authentication module instance is displayed. If no authentication level is specified, the SSO token stores the value specified in the Core Authentication attribute Default Authentication Level.

6.3.16 To Delete an Authentication Module Instance

1. In the Access Control tab, click the Authentication subtab.
2. In the Module Instances section, select the name of the module instance to be deleted.
3. Click Delete.

6.4 Managing Authentication Chains

You can configure multiple authentication modules instance so that a user must pass authentication credentials to all of them. This is known as authentication chaining.

When you configure an authentication chain, the user interacts with each of the authentication module instances in the chain, from the top down, in order to pass the authentication process. A criteria flag is assigned to each instance in the chain. The criteria flag defines how authentication proceeds through the ordered group of modules if, for example, authentication to one of the modules fails. Once authentication to all modules in the chain has been successfully achieved, the

Authentication Service validates that all user identifiers in the chain are mapped to the same user. After validation occurs, a session token is issued for the user and the policy evaluation process begins. Authentication chains can be assigned to a realm, a user, a role, or a service.

6.4.1 To Create a New Authentication Chain

1. In the Access Control tab, click the Authentication subtab.
2. To add a new chain, in the Authentication Chains section, click New.
3. In the Name field, type a name for the new authentication chain.
4. Click OK.
5. To add an authentication module to the authentication chain, in the ChainName Properties page, in the Items section, click Add.

   The authentication module instance name is displayed in the Items list.
6. To configure chaining for an authentication module in the Items list, first mark the box that corresponds to the authentication module you want to configure, then provide values for the Required Authentication Module Instance Chaining properties.

   The following table provides a listing and descriptions of the properties you can configure.

Table 6–20 Required Authentication Module Instance Chaining Properties

Property	Description
Instance	In the Instance column, a drop down list displays the authentication modules instantiated in the realm. Choose the authentication module instance to be configured.

Table 6–20 (Cont.) Required Authentication Module Instance Chaining Properties

Property	Description
Criteria	In the Criteria column, a dropdown list displays the supported options. Choose the criteria to be used for this authentication module instance. ■ REQUIRED Successful authentication to this module instance is required for the authentication process to succeed. The authentication process will continue through the authentication chain whether authentication to the REQUIRED module instance succeeds or fails. However, if authentication to any REQUIRED module instances defined in a chain fails, authentication will ultimately fail and the user is not authenticated. ■ REQUISITE Successful authentication to this module instance is required to proceed through the authentication chain. If authentication is successful, the authentication process moves to the next module instance in the authentication chain. If authentication fails, the chain is broken, control returns to the Authentication Service, and the user is not authenticated. ■ SUFFICIENT Successful authentication to this module is not required but, if authentication does succeed, the user is authenticated and the authentication process will not continue through the authentication chain. If authentication to a SUFFICIENT module instance fails, the authentication process continues through the module instances in the authentication chain. ■ OPTIONAL Successful authentication to this module instance is not required but, whether it succeeds or fails, the authentication process continues through the module instances in the authentication chain.
Options	In the Options column, you can define additional options for the authentication module instance. Type a key=value pair. Example: if the authentication module supports debugging, type `debug=true`. Multiple options must be separated by a space. More information can be found in the javax.security.auth.login.Configuration class document.

7. To reorder the authentications modules in the list, click Reorder.

The authentication module instances will be invoked in the order listed the authentication configuration. For example, if authentication module instance A which is moved below authentication module instance B, then B will be invoked before A.

a. In the Reorder Authentication Chains page, click a module instance name and then click Move Up, Move Down, Move to Top, or Move to Bottom until the instance name is in the proper place in the list.

b. Click OK.

8. (Optional) Provide values for the optional Authentication Chaining Post-Processing properties. The following table provides a listing and descriptions of the properties you can configure.

Table 6–21 Optional Authentication Chaining Post-Processing Properties

Property	Description
Successful Login URL	The Current Values list displays URLs that the user will be redirected to upon successful authentication. ■ To add a URL to the list, in the New Value field type the URL, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Failed Login URL	The Current Values list displays URLs that the user will be redirected to upon failed authentication. ■ To add a URL to the list, in the New Value field type the URL, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.
Post Authentication Processing Class	The Current Values list displays the name of a Java class to be used for customizing any post-authentication processes regardless of whether authentication succeeds or fails. ■ To add a Java class to the list, in the New Value field type the Java class name, and then click Add. ■ To remove an entry from the Current Values list, select the entry and them click Remove.

9. In the ChainName Properties page, click Save.

6.4.2 To Delete an Authentication Chain

Caution: Do not delete ldapService. Deleting ldapService may cause problems with logging into OpenSSO STS administration console.

1. In the Access Control tab, click the Authentication subtab.
2. In the Authentication Chains section, select the name of the authentication chain to be deleted.
3. Click Delete.

7

Using the Logging Service

The Oracle OpenSSO Security Token Service (OpenSSO STS) logging service records information such as access denials and approvals, authentication events, and authorization violations. Administrators can use the logs to track user actions, analyze traffic patterns, audit system usage, review authorization violations, and troubleshoot. The logged information is recorded in one centralized directory. This chapter contains the following topics:

7.1 About the Logging Service

The log files record a number of events for each of the OpenSSO STS services. These files should be checked by the administrator on a regular basis. The default directory for the log files is ConfigurationDirectory/uri/log/, where ConfigurationDirectory is the configuration directory, and uri is the OpenSSO deployment URI specified during OpenSSO configuration and deployment time. These tags are interpreted at run time. Each deployed OpenSSO instance has its own logging directory. This is particularly useful when there are multiple OpenSSO instances per system. The log file directory can be configured in the logging service by using the OpenSSO STS console or ssoadm command-line utility. An absolute path may also be configured as the log file directory. For detailed configuration steps, see "Configuring Global Logging Attributes" on page 7-4.

7.1.1 Log Records

Log records are created using the com.sun.identity.log.LogRecord class, and then logged by authenticated and authorized entities using the com.sun.identity.log.Logger class. Log records can be logged by:

- Other components of the OpenSSO STS server.
- Utilities installed on the OpenSSO STS server system.
- Other OpenSSO STS servers using a second instance of OpenSSO STS acting as the log server.
- Remote client applications (for example, policy agents) using the OpenSSO STS Logging Service.

The following table summarizes the events logged by default in the LogRecord, and a brief description of each event.

Table 7–1 Events Recorded in LogRecord

Event	Description
Time	The date (YYYY-MM-DD) and time (HH:MM:SS) at which the log message was recorded. This field is not configurable.
Data	Variable data pertaining to the log records's MESSAGE ID. This field is not configurable.
ModuleName	Name of the OpenSSO STS service or application being logged.
Domain	OpenSSO STS domain to which the user (whom the log record is regarding) belongs. This information is taken from the session token passed in the LogRecord(level,msg,token) call.
LogLevel	The Java 2 Platform, Standard Edition (J2SE) version 1.4 log level of the log record.
LoginID	The identifier of the user (taken from the session token) as the subject of the log record.
IPAddress	User who writes the log record. The information is taken from the session token passed during logger.log (`logRecord`, `ssoToken`).
HostName	Host name associated with the IP address above. This is present if the Log Record Resolve Host Name attribute is enabled. If not, the IP address is printed.
MESSAGEID	Non-internationalized message identifier for this log record's message.
ContextID	Session identifier associated with a particular login session. The session identifier is for the entity about whom the log record is regarding.

7.1.2 Error Logs and Access Logs

Access log files and error log files are the two types of log files used in OpenSSO STS. Access log files record general auditing information concerning the OpenSSO STS deployment. An access log may contain a single record for an event (such as a successful authentication), or multiple records for the same event. For example, when an administrator uses the console to change an attribute value, the logging service logs the attempt to change in one record but, it also logs the results of the execution of the change in a second record. Error log files record errors that occur within the application. While an operation error is recorded in the error log, the operation attempt is recorded in the access log file.

> **Tip:** The period (.) separator in a log filename is converted to an underscore (_) in database formats. Also in databases, table names may be converted to all upper case. For example, amConsole.access may be converted to AMCONSOLE_ACCESS, or it may be converted to amConsole_access.

7.1.3 Log File Formats

Log records generated for one event are entered as two separate records. The first log record records the attempt to perform an action. The second log record records the result of the attempt. The following example illustrates this two-record approach.

Example 7–1 Log Record Example

```
Data: agroupSubscription1|group|/
MessageID:CONSOLE-1
```

and

```
Data:agroupSubscription1|group|/
MessageID:CONSOLE-2
```

In this example, CONSOLE-1 indicates an attempt to create an identity object, and CONSOLE-2 indicates that the attempt to create the identity object was successful. The root organization is noted by a forward slash (/). The variable parts of the messages (agroupSubscription1, group, and /) are separated by a pipe character (|) and continue to go into the Data field of each log record. The MessageID string is not internationalized in order to facilitate machine-readable analysis of the log records in any locale.

OpenSSO STS can record events in either flat file format or relational database format.

Flat log files are appended with the .error or .access extension. Database column names end with _ERROR or _ACCESS for an Oracle database, or _error or _access for MySQL databases. For example, a flat file logging console events is named amConsole.access, while a database column logging the same events is named AMCONSOLE_ACCESS.

7.1.3.1 Flat File Format

The default flat file format is the W3C Extended Log Format (ELF). OpenSSO STS uses this format to record the default fields in each log record. See Table 7.1.1, "Log Records" for a list of default fields. The following example illustrates an authentication log record formatted for a flat file. The fields are in this order: Time, Data, ModuleName, MessageID, Domain, ContextID, LogLevel, LoginID, IPAddr, LoggedBy, and HostName.

Example 7–2 Flat File Log Record Example

```
"2005-08-01 16:20:28"   "Login Success" LDAP    AUTHENTICATION-100
   dc=example,dc=com       e7aac4e717dda1bd01      INFO
uid=amAdmin,ou=People,dc=example,dc=com 192.18.187.152
"cn=exampleuser,ou=Example Users,dc=example,dc=com" exampleHost
```

7.1.3.2 Relational Database Format

When OpenSSO STS uses a relational database to log messages, the messages are stored in a database table. OpenSSO STS uses Java Database Connectivity (JDBC), which provides connectivity to a wide range of databases. (Oracle® and MySQL databases are currently supported.) The following table summarizes the schema for a relational database.

Table 7–2 Relational Database Log Format

Event	Format	Description
TIME	Date (Oracle) DateTime (MySQL)	The format is YYYY-MM-DD HH24:MI:SS (Oracle) or %Y-%m-%d %H:%i:%s (MySQL). The formats are specified in the Logging Service attributes.
DATA	CLOB (Oracle) LONGTEXT (MySQL)	The data type is specified in the Logging Service attributes.

Table 7–2 (Cont.) Relational Database Log Format

Event	Format	Description
MODULE NAME	VARCHAR(255)	Name of the OpenSSO STS component invoking the log record.
DOMAIN	VARCHAR(255)	OpenSSO STS domain of the user.
LOGLEVELJDK	VARCHAR(255)	JDK 1.4 log level of the log record.
LOGINID	VARCHAR(255)	Login ID of the user who performed the logged operation.
IPADDRESS	VARCHAR(255)	IP Address of the machine from which the logged operation was performed.
LOGGEDBY	VARCHAR(255)	Login ID of the user who writes the log record.
HOSTNAME	VARCHAR(255)	Host name of machine from which the logged operation was performed.
MESSAGEID	VARCHAR(255)	Non-internationalized message identifier for this log record's message.
CONTEXTID	VARCHAR(255)	Identifier associated with a particular login session.

7.2 Configuring Global Logging Attributes

Configuring global attributes in the Logging Service configuration affects logging output. The Log Status can be set to Inactive to disable all logging output. The Logging Level can be set to one of the java.util.logging.Level values other than the default INFO to get more or less detailed logging output.

7.2.1 To Configure Global Logging Attributes

1. On the Configuration tab, click the System subtab.
2. On the System Configuration page, in the System Attributes list, click Logging.
3. Provide values for the Global Logging attributes. The following table provides a listing and descriptions of the attributes you can configure.

Table 7–3 Global Logging Attributes

Attribute	Description
Maximum Log Size	Specify the maximum number of bytes to allow for an OpenSSO STS log file. The default value is 100000000. This attribute applies to only the FILE logging type. When the logging type is set to DB, there are no history files and no size limit explicitly set by OpenSSO STS.
Number of History File	Specify the number of backup log files to be retained for historical analysis. You can enter any integer based on the partition size and available disk space of the local computer system. The default value is 1. Entering a value of 0 is interpreted to be the same as a value of 1. If you specify 0, a history log file will be created. This attribute applies to only the FILE logging type. When the logging type is set to DB, there are no history files and no size limit explicitly set by OpenSSO STS.

Table 7–3 (Cont.) Global Logging Attributes

Attribute	Description
Log File Location	Specify a path to a directory where OpenSSO STS log files can be stored. The default location is: OpenSSO-deploy-base/uri/log You can set this value an explicit path, but the base path should be the value of OpenSSO-deploy-base to avoid permissions problems. OpenSSO-deploy-base/uri/log represents the base configuration directory and the OpenSSO STS deployment URI. These are specified during post-installation configuration. At runtime, the Logging service determines the instance's proper directory for logging. If a non-default directory is specified, OpenSSO STS will create the directory if it does not already exist. You must then set the appropriate permissions for that directory. Example: 0700. When configuring the log location for DB (database) logging such as, Oracle or MySQL, part of the log location is case sensitive. For example, if you are logging in to an Oracle database, the log location is case-sensitive as in the following example: jdbc:oracle:thin:@machine.domain:port:DBName To configure logging to DB, add the JDBC driver files to the web container's JVM classpath. You must manually add JDBC driver files to the classpath of the ssoadm script. Otherwise ssoadm logging can not load the JDBC driver. Changes to logging attributes usually take effect after you save them. This does not require you to restart the server. If you are changing to secure logging, however, you should restart the server.
Log Status	Specify whether logging is turned on (ACTIVE) or off (INACTIVE). Value is set to ACTIVE during installation.
Log Record Resolve Host Name	When set to false, host lookups will not be performed to populate the LogRecord's HostName field.
Logging Type	Specify one of the following: ■ File, - for flat file logging ■ DB - for database logging If the Database User Name or Database User Password is invalid, it will seriously affect OpenSSO STS processing. If OpenSSO STS or the console becomes unstable, set the Log Status attribute to Inactive. After you have set the property, restart the server. You can then log in to the console and reset the logging attribute. Then, change the Log Status property to ACTIVE and restart the server.
Database User Name	Specify the name of the user that will connect to the database when the Logging Type attribute is set to DB.
Database User Password	Specify the database user password when the Logging Type attribute is set to DB.
Database User Password (confirm)	Type the Database User Password again to confirm it.
Database Driver Name	Specify the driver used for the logging implementation class.

Table 7–3 (Cont.) Global Logging Attributes

Attribute	Description
Configurable Log Fields	Specify the fields that are to be logged. By default, all of the following fields are logged: CONTEXTID DOMAIN HOSTNAME IPADDRESS LOGGED BY LOGINID LOGLEVEL MESSAGEID MODULENAME NAMEID At minimum you should log CONTEXTID, DOMAIN, HOSTNAME, LOGINID and MESSAGEID.
Log Verification Frequency	Specify in seconds the how often the server should verify the logs to detect tampering. The default time is 3600 seconds. This parameter applies to secure logging only.
Log Signature Time	Specify in seconds how often that the log will be signed. The default time is 900 seconds. This parameter applies to secure logging only.
Secure Logging	
Secure Logging Signing Algorithm	When enabled, secure logging detects unauthorized changes or tampering of security logs. By default, secure logging disabled. Secure logging can only be used for flat files. This option does not work for Database (DB) logging.
Logging Certificate Store Location	Choose one of the following RSA or DSA encryption signing algorithms. Each has private keys for signing and a public key for verification: MD2 w/RSA MD2 w/RSA SHA1 w/DSA SHA1 w/RSA MD2, MD5 and RSA are one-way hashes. For example, if you select the signing algorithm MD2 w/RSA, the secure logging feature generates a group of messages with MD2 and encrypts the value with the RSA private key. This encrypted value is the signature of the original logged records and will be appended to the last record of the most recent signature. For validation, it well decrypt the signature with the RSA public key and compare the decrypted value to the group of logged records. The secure logging feature will then will detect any modifications to any logged record.
Maximum Number of Records	Specify the maximum number of records that the Java LogReader interfaces should return, regardless of how many records match the read query. By default, it is set to 500. This attribute can be overridden by the caller of the Logging API through the LogQuery class.
Number of Files per Archive	This attribute is applicable to only secure logging. Specify when the log files and keystore must be archived, and the secure keystore regenerated, for subsequent secure logging. The default is five, and means that the log files and keystore are archived after five log files have been created.

Table 7–3 (Cont.) Global Logging Attributes

Attribute	Description
Buffer Size	Specify the maximum number of log records to be buffered in memory before the logging service attempts to write them to the logging repository. The default is one record.
DB Failure Memory Buffer Size	Specify the maximum number of log records held in memory if database (DB) logging fails. This attribute is only applicable when DB logging is specified. When the OpenSSO STS Logging service loses connection to the DB, it buffers up to the number of records specified here. The default value is two times of the value defined in the Buffer Size attribute.
Buffer Time	Specify the number of seconds that the log records will be buffered in memory before they are sent to the Logging service to be written. This attribute applies if Time Buffering is ON. The default is 3600 seconds.
Time Buffering	When ON is selected, OpenSSO STS sets a time limit for log records to be buffered in memory before they are written. The amount of time is set in the Buffer Time attribute.
Logging Level	Specify the degree of detail to be contained in all OpenSSO STS log files. The default is the INFO level. FINE, FINER, FINEST provide more detail and more log records. Use the OFF level to turn off logging, which is essentially the same as setting the Log Status attribute to INACTIVE.

4. Click Save.

7.3 Using OpenSSO STS Component Logs

The log files record a number of events for each of the OpenSSO STS components using the logging service. Administrators typically review these log files on a regular basis. The following table provides a brief description of the log files produced by each OpenSSO STS component.

Table 7–4 OpenSSO STS Component Logs

Component	Log Filename	Description
Session Service	■ amSSO.access	Session management attributes values such as login time, logout time, and time out limits. Also session creations and terminations.
Administration Console	■ amConsole.access ■ amConsole.error	User actions performed through the administration console such as creation, deletion and modification of identity-related objects, realms, and policies. amConsole.access logs successful console events while amConsole.error logs error events.
Authentication Service	■ amAuthentication.access ■ amAuthentication.error	User logins and log outs, both successful and failed.

Table 7–4 (Cont.) OpenSSO STS Component Logs

Component	Log Filename	Description
Federation Services	■ amFederation.access ■ amFederation.error ■ amLiberty.access ■ amLiberty.error	Federation-related events such as the creation of an authentication domain or the creation of a hosted provider entity.
Policy Service (Authorization)	■ amPolicy.access ■ amPolicy.error ■ amAuthLog	Events related to authorization such as policy creation, deletion, or modification, and policy evaluation. amPolicy.access logs policy allows, amPolicy.error logs policy error events, and amAuthLog logs policy denies.
Policy Agents	amAgent	Exceptions regarding resources that were either accessed by a user or denied access to a user. amAgent logs reside on the server where the policy agent is installed. Agent events are logged on the OpenSSO STS machine in the Authentication logs.
SAML v1.x	■ SAML.access ■ SAML.error	SAML v1.x-related events such as assertion and artifact creation or removal, response and request details, and SOAP errors.
SAML v2	■ SAML2.access ■ SAML2.error	SAML v2-related events such as assertion and artifact creation or removal, response and request details, and SOAP errors.
Command Line	■ amAdmin.access ■ amAdmin.error	Event successes and errors that occur during operations using the command line tools. Loading a service schema, creating policy, and deleting users are some examples of command line operations.
Password Reset	amPasswordReset.access	Password reset events.
Web Services Security	WebServiceSecurity.access	Event successes that occur during operations using the command-line tools.

7.4 Using Secure Logging

Secure logging can only be used for flat files. Secure logging does not work for Database (DB) logging.

Secure logging adds an extra measure of security to the logging service. When secure logging is enabled, the logging service can detect unauthorized changes to the security logs. No special coding is required to leverage this feature. However, secure logging uses a certificate that you must create and install in the container that runs OpenSSO STS. When secure logging is enabled, a Manifest Analysis and Certification (MAC) is generated and stored for every log record, and a special signature record is periodically inserted in the log. The signature record represents the signature for the contents of the log written up to that point. The combination of the certificate and the signature record ensures that the logs have not been tampered.

There are two methods to enable secure logging; through a through a Java Cryptography Extension (JCE) provider and through a Java Security Server (JSS) provider.

7.4.1 To Enable Secure Logging through a JSS Provider

1. Create a certificate with the name Logger and install it in the key store specified by the Logging Service configuration's Logging Certificate Store Location.

   The key store's password is expected to be the same as the top-level administrator password. The default location set during OpenSSO STS configuration is ConfigurationDirectory/uri/Logger.jks/, where ConfigurationDirectory is the configuration directory, and uri is the OpenSSO deployment URI specified during OpenSSO configuration. These tags are interpreted at run time. Each deployed OpenSSO instance has its own key store. It is particularly useful when there are multiple OpenSSO instances per system.

2. Turn on Secure Logging in the logging service configuration using the OpenSSO STS administration console and save the change. See Section 7.2, "Configuring Global Logging Attributes."The administrator can also modify the default values for the other logging service attributes.

   If the logging directory is changed from the default /log directory, make sure that the directory is writable by the user ID and that the OpenSSO STS's web application is running. Also set the directory's permissions to 0700, as the logging service will create the directory, if it does not exist, with permissions set to 0755.

3. Verify Secure Log Archives.

   To detect unauthorized changes or tampering of the secure logs, look for error messages that are written by the Logging Service's periodic verification process to ConfigurationDirectory/uri/debug/amLog. To manually check for tampering, run the amverifyarchive command-line utility, which is included in the ssoAdminTools.zip file.

4. Changing from a JCE Provider to a JSS Provider

   The default secure log helper provider is the JCE provider, com.sun.identity.log.secure.impl.SecureLogHelperJCEImpl, as specified by the iplanet-am-logging-secure-log-helper attribute in the iPlanetAMLoggingService's schema. Refer to the opensso/xml/amLogging.xml file from the opensso.zip file.

7.4.2 To Change from a JCE Provider to a JSS Provider

The default secure log helper provider is the JCE provider, com.sun.identity.log.secure.impl.SecureLogHelperJCEImpl, as specified by the iplanet-am-logging-secure-log-helper attribute in the iPlanetAMLoggingService's schema. Refer to the opensso/xml/amLogging.xml file from the opensso.zip file.

1. Execute the following ssoadm command:

```
./ssoadm set-attr-defs --servicename iPlanetAMLoggingService --schematype
global --attributevalues iplanet-am-logging-secure-log-helper-class-name=
com.sun.identity.log.secure.SecureLogHelperJSSImpl --adminid amadmin
--password-file amadminpass
```

2. Verify the change:

```
./ssoadm get-attr-defs --servicename iPlanetAMLoggingService --attributenames
iplanet-am-logging-secure-log-helper-class-name --schematype global --adminid
amadmin --password-file amadminpass
```

7.5 Using Database Logging

This feature provides logging to Oracle or MySQL databases. No special coding is required to enable this feature.

The DB Failure Memory Buffer Size specifies how many records per table to buffer if the connection to the database fails. If more records are queued before the connection is reestablished, older records will be discarded.

The ssoadm command line interface cannot log to the database directly. In addition to adding the JDBC driver to the web application's classpath, remove `-D"com.sun.identity.log.dir=`*the_specified_log_dir*.

7.5.1 To Enable Database Logging

1. On the Configuration tab, click the System subtab.
2. On the System Configuration page, in the System Attributes list, click Logging.
3. Set the Logging Type to DB.
4. Set the Database User Name, Database User Password, and Database Driver Name.
   - For Oracle, the default driver name set is oracle.jdbc.driver.OracleDriver.
   - For MySQL, it is typically `com.mysql.jdbc.Driver`.
5. Specify values for other fields.

   See Section 7.2.1, "To Configure Global Logging Attributes."

Be sure to put the JDBC driver's .zip or .jar file in the OpenSSO STS web application's classpath (for example, WEB-INF/lib or jre/lib/ext).

8

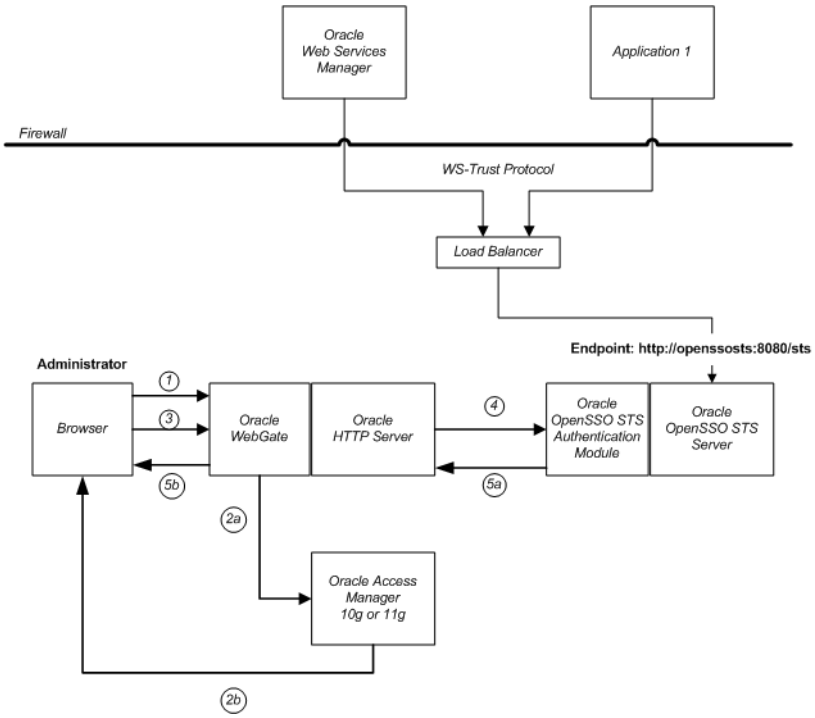
Deploying OpenSSO STS with Other Oracle Products

The following topics are contained in this chapter:

8.1 Configuring Administrator Single Sign-On with Oracle Access Manager

You can deploy OpenSSO STS in an environment where Oracle Access Manager already exists. When OpenSSO STS and Oracle Access Manager are configured for single sign-on, an administrator who has authenticated to Oracle Access Manager can access OpenSSO STS without having to present credentials a second time. The administrator single sign-on use case requires that Oracle HTTP Server is deployed in the environment to act as a proxy for OpenSSO STS, and OpenSSO STS must be configured with the Oracle Access Manager authentication module. The following diagram illustrates a typical single sign-on topology.

Figure 8–1 Single Sign-On Using Oracle Access Manager and OpenSSO STS



In this topology, Oracle WebGate is installed on the Oracle HTTP Server. The HTTP Server must be configured in proxy mode for OpenSSO STS, and OpenSSO STS must be deployed on a supported web container.

When an administrator attempts to access OpenSSO STS, Oracle WebGate intercepts the request. Then Oracle Access Manager presents a login page. The administrator presents credentials, which are then authenticated to Oracle Access Manager. Once the administrator has been authenticated, the access request is redirected to the browser, and then to OpenSSO STS. OpenSSO STS is configured with the Oracle Access Manager authentication module, enabling Oracle Access Module to validate the administrator based on a list of allowed users. If the administrator is on the list of allowed users, then the administrator can access the OpenSSO STS console without having to present credentials.

8.1.1 To Configure Administrator Single Sign-On with Oracle Access Manager

The following list summarizes high-level steps you must take to configure administrator single sign-on with Oracle Access Manager. Follow the detailed installation and configuration instructions in the documentation for each Oracle product in your environment. See `http://www.oracle.com/technology/documentation/index.html`.

1. Install OpenSSO STS on Oracle WebLogic Server.
2. Install Oracle Access Manager.
3. Install Oracle HTTP Server and configure it to proxy for OpenSSO STS.
4. Install Oracle Webgate on Oracle HTTP Server.
5. Configure single sign-on between Oracle HTTP Server and Oracle Access Manager to protect the OpenSSO STS login URL:

```
http://HostName.Domain.com:port/openssosts/UI/Login?module=OA
MAuthModule
```

6. Configure Oracle HTTP Server for proxying.

   Edit `$OH_INSTANCE_DIR/config/OHS/ohs1/mod_w1_ohs.conf` to include the following:

   ```
   WebLogicHost:HostName.Domain.com
   WeblogiPort: 7001
   MatchExpression: openssosts
   ```

7. Access the Webgate URL using the OpenSSO STS URI (the proxy URL). Example:

   ```
   http://HostName.Domain.com:port/openssosts/UI/Login?module=OA
   MAuthModule
   ```

   The browser redirects the user request to the Oracle Access Manager console.

8. Log in to Oracle Access Manager using OpenSSO STS administrator credentials.

   By default, Oracle Access Manager sets the remote user as OAM_REMOTE_USER. The OpenSSO STS validates OAM_REMOTE_USER, and provides access to the OpenSSO STS administration console.

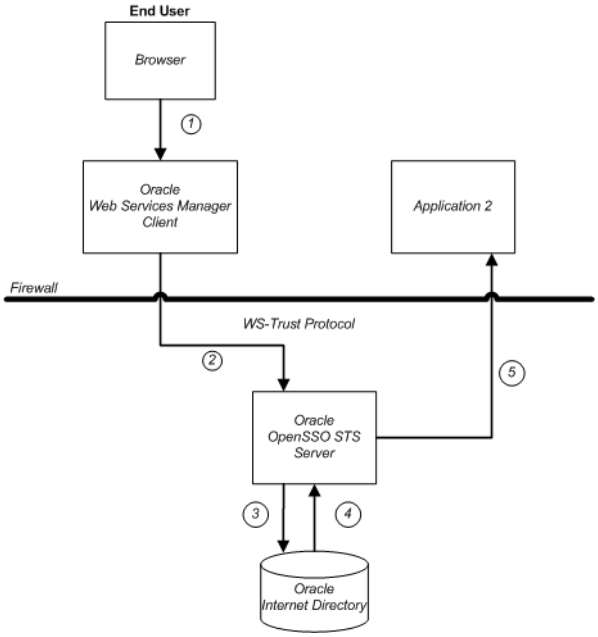
9.

8.2 Configuring OpenSSO STS to Work with Oracle Internet Directory and Oracle Virtual Directory

Both Oracle Internet Directory and Oracle Virtual Directory are LDAP-compliant directory services. You can add an LDAP authentication module to OpenSSO STS, and then configure the LDAP authentication module to use either Oracle Internet Directory or Oracle Virtual Directory as a user data store.

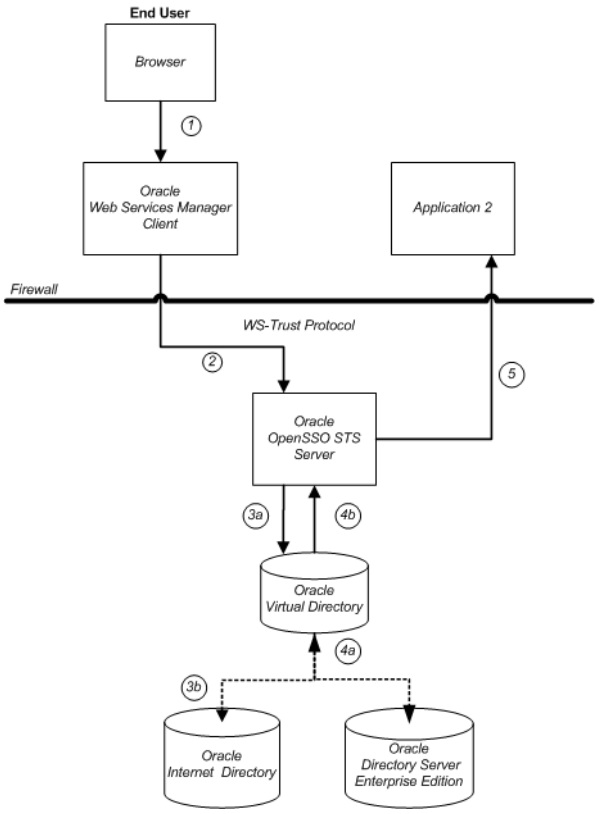
The following figure illustrates a topology in which Oracle Internet Directory is the sole user directory used for authentication and attribute retrieval.

Figure 8–2 OpenSSO STS Integrated with Oracle Internet Directory



Oracle Virtual Directory provides an abstraction layer that enables external directories to connect to OpenSSO STS. When OpenSSO STS receives a WS-Trust request with a UserName token (or UserName On-Behalf-Of token) as either an inbound or outbound token, OpenSSO STS validates the user, though Oracle Virtual Directory, against one of the connected directories. The following figure illustrates a typical topology using Oracle Virtual Directory and two LDAP directories.

Figure 8–3 OpenSSO STS Integrated with Oracle Virtual Directory



An inbound request is sent through Oracle Web Services Manager client in the form of a SOAP request. OpenSSO STS receives the request and authenticates it, through Oracle Virtual Directory, against Oracle Internet Directory. OpenSSO STS reads the attributes from the UserName token, and generates a SAML assertion response.

8.2.1 To Configure Oracle Internet Directory or Oracle Virtual Directory for User Authentication

The following summarizes high-level steps you must take to configure Oracle Internet Directory for user authentication.

1. In the OpenSSO STS console, add and configure an Oracle Internet Directory or Oracle Virtual Directory authentication module instance.

   See Section 6.3.10, "To Add an New LDAP Authentication Module Instance" and Section 6.3.11, "To Configure an LDAP Authentication Module Instance."

2. Create and configure an authentication chain for inbound security tokens.

   See Section 6.4.1, "To Create a New Authentication Chain."

3. Create and configure an authentication chain for outbound security tokens.

   See Section 6.4.1, "To Create a New Authentication Chain."

4. Configure the Security Token Service.

   See Section 4.2, "To Configure the Security Token Service."

The OpenSSO STS authentication service determines the security mechanism registered by the web service provider, and retrieves the appropriate security tokens.

5. Enable both inbound and outbound authentication chains based on the authentication chain security mechanism specified when the web service provider agent profile was created.

   See Section 4.4, "To Register a Web Service Provider to OpenSSO STS."

8.2.2 To Configure SAML Attribute Generation and Retrieval

Each time you add a web service provider to OpenSSO STS, you must be sure that the web service provider is configured for SAML attribute retrieval, and that SAML attribute mapping is defined in OpenSSO STS. The following summarizes high-level steps you must take to use Oracle Internet Directory or Oracle Virtual Directory for SAML attribute generation and retrieval in a web service provider.

1. Add a new LDAP data store and configure it to communicate with Oracle Internet Directory or Oracle Virtual Directory.

   See Section 5.3.1, "To Add a New User Data Store."

2. In the web service provider profile, configure the SAML attribute mapping.

   See Section 5.2, "Managing OpenSSO STS Sites."

Part III

Appendixes

Part III contains the following appendixes:

- Appendix A, "Using the ssoadm Command-Line Interface"
- Appendix B, "Debugging and Troubleshooting OpenSSO STS"

A

Using the ssoadm Command-Line Interface

This chapter provides information about the ssoadm command-line interface. The following topics are contained in this chapter:

- About ssoadm
- Basic ssoadm Usage
- Command-Line Reference

A.1 About ssoadm

The ssoadm interface has two main purposes: to load configuration data into the data store, and to perform batch administrative tasks. You can use ssoadm to load XML service files that use the format defined in the sms.dtd. XML service files are stored in the configuration data store and are referenced only by OpenSSO STS. When ssoadm is executed, the utility automatically checks the OpenSSO STS version. If the version does not match, the ssoadm command fails.

For information about installing the ssoadm utility, see Section 1.6, "Installing the OpenSSO STS Command-Line Utility."

A.2 Basic ssoadm Usage

The ssoadm command contains subcommands to perform specific tasks for OpenSSO STS services and plug-ins. Each subcommand contains a number of options, both required and optional, that are designed to carry out these tasks.

A.2.1 ssoadm Syntax

ssoadm subcommand --options [--global-options]

The following global options are common to all subcommands, but are not required for the command to function:

Table A–1 Global Options for ssoadm

Option	Short Form	Description
--locale	-l	Name of the locale to display the results.
--debug	-d	Run in debug mode. Results sent to the debug file.
--verbose	-v	Run in verbose mode. Results sent to standard output.

A.2.2 Password File

In most ssoadm subcommands, the password file is required. The password file is a simple file that contains the administrator password for the given task.

To create a password file, complete the following steps:

1. Create the password file in a location you will remember. Example:

   # echo "" > /tmp/testpwd

2. Change the permissions to read-only. Example:

   # chmod 400 /tmp/testpwd

A.2.3 ssoadm Usage Example

This example uses the update-agent option to illustrate how to use the ssoadm command with options.

./ssoadm update-agent -e testRealm1 -b testAgent1 -f /tmp/testpwd -a "com.sun.identity.agents.config. notenforced.url[0]=/exampledir/public/*"

When using the ssoadm command, if you include values that contain wildcards (* or -*-), then be sure to enclose the property name/value pair in quotes to avoid substitution by the shell. This applies when you use the -a (--attributevalues) option. The double quotes are not necessary when you list the properties in a data file and access them with the -D option.

A.2.4 Displaying Options for an ssoadm Subcommand

You can display a list of options while using the ssoadm command. On the OpenSSO STS host, in the directory containing the ssoadm utility, issue the ssoadm command with the appropriate subcommand with no options. For example:

```
ssoadm list-wsps --options [--global-options]
    List web service providers.

Usage:
ssoadm list-wsps
    --adminid|-u
    --password-file|-f
    [--filter|-x]

Global Options:
    --locale, -l
        Name of the locale to display the results.

    --debug, -d
        Run in debug mode. Results sent to the debug file.

    --verbose, -v
        Run in verbose mode. Results sent to standard output.


Options:
    --adminid, -u
        Administrator ID of running the command.

    --password-file, -f
        File name that contains password of administrator.
```

```
    --filter, -x
        Filter (Pattern).
```

In this example, since the command is missing required options, the utility lists all the options available for this subcommand. The global options are common to all subcommands.

A.2.5 ssoadm Subcommand Usage

By looking at the usage information of a subcommand, you can determine which options are required and which are optional. You can list an option for the command with either a single letter, such as -u or with an entire word, such as --adminid. The following table lists options and usage information for the list-wsps subcommand.

Option	Short Form	Description
--adminid	-u	Administrator ID of running the command
--password-file	-f	File name that contains password of administrator
--filter	-x	Filter (Pattern)

The options not bounded by square brackets [ ] are required. In this example, adminid, password-file, and filter are required.

For subcommand options that accept multiple values, the values are space-separated and placed within quotation marks. For example, the --attrubutevalues option uses the following format:

-attributevalues "attributename=value" "-attributename =value2"

A.3 Command-Line Reference

The following table lists ssoadm command and brief descriptions. Click a command name to jump to more detailed information about the command.

Table A–2 Summary of ssoadm Commands

Command	Description
add-attrs	Add an attribute schema to an existing service.
add-attr-defs	Add the default attribute values in a schema.
add-auth-cfg-entr	Add an authentication configuration entry.
add-plugin-interface	Add the plug-in interface to a service.
add-site-members	Add members to a site.
add-site-sec-urls	Add site secondary URLs.
add-sub-schema	Add a sub schema.
clone-server	Clone a server instance.
create-agent	Create a new agent configuration.
create-auth-cfg	Create an authentication configuration.
create-auth-instance	Create an authentication instance.
create-boot-url	Create a bootstrap URL that can bootstrap the product web application.
create-datastore	Create a datastore under a realm.

Table A–2 (Cont.) Summary of ssoadm Commands

Command	Description
create-server	Create a server instance.
create-site	Create a site.
create-sub-cfg	Create a new sub configuration.
create-svc	Create a new service in the server.
create-svrcfg-xml	Create the serverconfig.xml file.
create-wsp	Creates a new web service provider.
create-wsp-grp	Create a new web service provider group.
delete-attr	Delete the attribute schemas from a service.
delete-attr-def-values	Delete the attribute schema default values.
delete-auth-cfgs	Delete existing authentication configurations.
delete-auth-instances	Delete existing authentication instances.
delete-datastores	Delete the data stores under a realm.
delete-server	Delete a server instance.
delete-site	Delete a site.
delete-sub-cfg	Delete the sub configuration.
delete-svc	Delete the service from the server.
delete-wsps	Delete web service providers.
delete-wsp-grps	Delete web service provider groups.
do-batch	Do multiple requests in one command.
export-server	Export a server instance
export-svc-cfg	Export the service configuration.
get-attr-defs	Get the default attribute values in a schema.
get-auth-cfg-entr	Get the authentication configuration entries.
get-auth-instance	Get the authentication instance values.
get-revision-number	Get the service schema revision number.
get-svrcfg-xml	Get the server configuration XML from the centralized data store.
import-server	Import a server instance.
import-svc-cfg	Import the service configuration.
list-auth-cfgs	List the authentication configurations.
list-auth-instances	List the authentication instances.
list-datastores	List the data stores under a realm.
list-datastore-types	List the supported data store types.
list-server-cfg	List the server configuration.
list-servers	List all the server instances.
list-sites	List all the sites.
list-wsps	Lists web service providers.

Table A–2 (Cont.) Summary of ssoadm Commands

Command	Description
list-wsp-grps	List web service provider groups.
list-wsp-grp-members	List web service providers in web service provider group.
register-auth-module	Register an authentication module.
remove-attr-choicevals	Remove choice values from the attribute schema.
remove-attr-defs	Remove the default attribute values in a schema.
remove-server-cfg	Remove the server configuration.
remove-site-members	Remove members from a site.
remove-site-sec-urls	Remove the site secondary URLs.
remove-sub-schema	Remove the sub schema.
remove-wsp-from-grp	Remove web service providers from a group.
set-attr-any	Set any member of the attribute schema.
set-attr-bool-values	Set the boolean values of the attribute schema.
set-attr-choicevals	Set choice values for the attribute schema.
set-attr-defs	Set the default attribute values in a schema.
set-attr-end-range	Set the attribute schema end range.
set-attr-i18n-key	Set the i18nkey member of the attribute schema.
set-attr-start-range	Set attribute schema start range.
set-attr-syntax	Set syntax member of attribute schema.
set-attr-type	Set the type member of the attribute schema.
set-attr-ui-type	Set the UI type member of the attribute schema.
set-attr-validator	Set the attribute schema validator.
set-attr-view-bean-url	Set the properties view bean URL member of the attribute schema.
set-inheritance	Set the inheritance value of the sub schema.
set-plugin-viewbean-url	Set the properties view bean URL of the plug-in schema.
set-revision-number	Set the service schema revision number.
set-site-pri-url	Set the primary URL of a site.
set-site-sec-urls	Set the site secondary URLs.
set-sub-cfg	Set the sub configuration.
show-wsp	Shows web service provider.
set-svc-i18n-key	Set the service schema i18n key.
set-svc-view-bean-url	Set the service schema properties view bean URL.
set-svrcfg-xml	Set the server configuration XML to the centralized data store.
show-auth-modules	Show the supported authentication modules in the system.
show-datastore	Show the data store profile.
show-site	Show the site profile.
show-site-members	Display the members of a site.

Table A–2 (Cont.) Summary of ssoadm Commands

Command	Description
show-wsp-grp	show web service provider group profile.
show-wsp-membership	List web service provider's membership.
unregister-auth-module	Unregister the authentication module.
update-auth-cfg-entr	Set the authentication configuration entries.
update-auth-instance	Update the authentication instance values.
update-datastore	Update the datastore profile.
update-server-cfg	Update the server configuration.
update-svc	Update the service.
update-wsp	Update web service provider.
update-wsp-grpd	Update web service provider group configuration.
wsp-remove-propsd	Remove web service provider's properties.

ssoadm Commands

add-attrs

Add an attribute schema to an existing service.

```
ssoadm add-attrs --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschemafile	-F	An XML file containing the attribute schema definition.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

add-attr-defs

Add the default attribute values in a schema.

```
ssoadm add-attr-defs --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--adminid, -u	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--attributevalues]	-a	The attribute values. For example, homeaddress=here.
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.
[--subschemaname]	-c	The name of the sub schema.

add-auth-cfg-entr

Add an authentication configuration entry.

```
ssoadm add-auth-cfg-entr --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--name	-m	The name of the authentication configuration.
--modulename	o	The module name.

Option	Short Form	Description
--criteria	-c	The criteria for this entry. Possible values are REQUIRED, OPTIONAL, SUFFICIENT, and REQUISITE.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--options]	-t	The options for this entry.
[--position]	-p	The position where the new entry is to be added.

add-plugin-interface

Add the plug-in interface to a service.

```
ssoadm add-plugin-interface --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--interfacename	-s	The name of the interface.
--pluginname	-g	The name of the plug-in.
--i18nkey	-g	The name of the plug-in.
--i18nkey	-k	The i18n key plug-in.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

add-site-members

Add members to a site.

```
ssoadm add-site-members --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The name of the site. For example, mysite.
--servernames	-e	The server name. For example, http://www.example.com:8080/openssosts
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

add-site-sec-urls

Add site secondary URLs.

```
ssoadm add-site-sec-urls --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The name of the site. For example, mysite.
--secondaryurls	-a	The secondary URLs.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

add-sub-schema

Add a sub schema.

```
ssoadm add-sub-schema --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--filename	-F	The filename that contains the schema.
--adminid	-u	The administrator ID running the command.
[--subschemaname]	-c	The name of the sub schema.

clone-server

Clone a server instance.

```
ssoadm clone-server --options [--global-options]
```

Option	Short Form	Description
--servername	-a	The server name.
--cloneservername	-o	The clone server name.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

create-agent

Create a new agent configuration.

```
ssoadm create-agent --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--agentname	-b	The type of agent. For example, J2EEAgent or WebAgent.
--adminid	-u	The type of agent. For example, J2EEAgent or WebAgent.
--adminid	-u	The administrator ID running the command.

Option	Short Form	Description
--password-file	-f	The filename that contains the password of the administrator.
[--attributevalues]	-f	The filename that contains the password of the administrator.
[--attributevalues]	-a	The properties. For example, homeaddress=here.
[--datafile]	-a	The properties. For example, homeaddress=here.
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.

create-auth-cfg

Create an authentication configuration.

```
ssoadm create-auth-cfg --options [--global-options]
```

Option	Short Form	Description
-realm	-e	The name of the realm.
-name	-m	The name of the authentication configuration.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

create-auth-instance

Create an authentication instance.

```
ssoadm create-auth-instance --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--name	-m	The name of the authentication instance.
--authtype	-t	The type of authentication instance. For example LDAP or DataStore.
--adminid	-u	The administrator ID running the command.
-password-file	-f	The filename that contains the password of the administrator.

create-boot-url

Create a bootstrap URL that can bootstrap the product web application.

```
ssoadm create-boot-url --options [--global-options]
```

Option	Short Form	Description
--dshost	-t	The Directory Server hostname.
--dsport	-p	The Directory Server port number.

Option	Short Form	Description
--basedn	-p	The Directory Server port number.
--basedn	-b	The Directory Server base distinguished name.
--dsadmin	-a	The Directory Server base distinguished name.
--dspassword-file	-x	The filename that contains the Directory Server administrator password.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--ssl]	s	Set this flag for LDAPS.

create-datastore

Create a datastore under a realm.

```
ssoadm create-datastore --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--name	-m	The name of the datastore.
--datatype	-t	The type of the datastore.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--attributevalues]	-a	The attribute values. For example, sunIdRepoClass=com.sun.identity.idm.plugins.ldapv3 .LDAPv3Repo".
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.

create-server

Create a server instance.

```
ssoadm create-server --options [--global-options]
```

Option	Short Form	Description
--servername	-a	The server name. For example, http://www.example.com:8080/opensso.
--serverconfigxml	-X	The server configuration XML filename.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--attributevalues]	-a	The attribute values. For example, homeaddress=here.

Option	Short Form	Description
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.

create-site

Create a site.

```
ssoadm create-site --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The site name. For example, mysite.
--siteurl	-i	The site's primary URL. For example, http://www.example.com:8080.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--secondaryurls]	-a	The secondary URLs.

create-sub-cfg

Create a new sub configuration.

```
ssoadm create-sub-cfg --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--subconfigname	-g	The name of the sub configuration.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--attributevalues]	-a	The attribute values. For example, homeaddress=here.
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.
[--realm]	-e	The name of the realm. The sub configuration will be added to the global configuration if this option is not selected.
[--subconfigid]	-b	The ID of the parent configuration. The sub configuration will be added to the root configuration if this option is not selected.
[--priority]	-p	The priority of the sub configuration.

create-svc

Create a new service in the server.

```
ssoadm create-svc --options [--global-options]
```

Option	Short Form	Description
--xmlfile	-X	The XML file that contains the schema.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--continue]	-c	Continue adding services if one or more previous services can not be added.

create-svrcfg-xml

Create the serverconfig.xml file.

```
ssoadm create-svrcfg-xml --options [--global-options]
```

Option	Short Form	Description
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--dshost]	-t	The Directory Server hostname.
[--dsport]	-p	The Directory Server port number.
[--basedn]	-b	The Directory Server base distinguished name.
[--dspassword-file]	-x	The filename that contains the Directory Server administrator password.
[--outfile]	-o	The filename where serverconfig.xml is written.

create-wsp

Creates a new web service provider.

```
ssoadm create-wsp --options [--global-options]
```

Example:

```
# ./ssoadm create-wsp -u amadmin -f /tmp/fampass --wspname wsptest  --securitymech
urn:sun:wss:security:null:SAMLToken-HK --endpoint Default --publickeyalias test1
--samlattributemapping "abc=xyz" --nameidmapper nameidmapper.class
--attributenamespace 123  --includememberships true

Web service provider was created.
```

Option	Short Form	Description
--wspname	-b	Name of web service provider.
--securitymech	-y	Security mechanism.
--endpoint	-e	Web service provider's end point
--publickeyalias	-a	Public key alias
--samlattributemapping	-t	SAML Attribute Mapping
--nameidmapper	-i	SAML NameID Mapper Plugin
--attributenamespace	-p	Attribute Namespace

Option	Short Form	Description
--includememberships	-m	Include Memberships. Possible values are true or false.
--adminid	-u	Administrator ID of running the command.
--password-file	-f	File name that contains password of administrator.

create-wsp-grp

Create a new web service provider group.

```
ssoadm create-wsp-grp --options [--global-options]
```

Example:

```
# ./ssoadm create-wsp-grp -u amadmin -f /tmp/fampass --groupname wspgroup
--securitymech urn:sun:wss:security:null:SAMLToken-HK --endpoint Default
--publickeyalias test1 --samlattributemapping "abc=xyz" --nameidmapper
nameidmapper.class --attributenamespace 123  --includememberships false

Group was created.
```

Option	Short Form	Description
--groupname	-b	Name of web service provider group
--securitymech	-y	Security mechanism
--endpoint	-e	Web service provider's end point
--publickeyalias	-a	Public key alias
--samlattributemapping	-t	SAML Attribute Mapping
--nameidmapper	-i	SAML NameID Mapper Plugin
--attributenamespace	-p	Attribute Namespace
--includememberships	-m	false]
--adminid	-u	Administrator ID of running the command
--password-file	-f	File name that contains password of administrator

delete-attr

Delete the attribute schemas from a service.

```
ssoadm delete-attr --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschema	-a	The administrator ID running the command.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

delete-attr-def-values

Delete the attribute schema default values.

```
ssoadm delete-attr-def-values --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--defaultvalues	-e	The default values to be deleted.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname,	-c	The name of the sub schema.

delete-auth-cfgs

Delete existing authentication configurations.

```
ssoadm delete-auth-cfgs --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--names	-m	The names of the authentication configurations.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

delete-auth-instances

Delete existing authentication instances.

```
ssoadm delete-auth-instances --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--names	-m	The names of the authentication instances.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

delete-datastores

Delete the data stores under a realm.

```
ssoadm delete-datastores --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.

Option	Short Form	Description
--names	-m	The names of the data stores.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

delete-server

Delete a server instance.

```
ssoadm delete-server --options [--global-options]
```

Option	Short Form	Description
--servername	-s	The server name. For example, http://www.example.com:8080/openssosts.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

delete-site

Delete a site.

```
ssoadm delete-site --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The site name. For example, mysite.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

delete-sub-cfg

Delete the sub configuration.

```
ssoadm delete-sub-cfg --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--subconfigname	-g	The name of the sub configuration.
-adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
--attributevalues	-a	The attribute values. For example, homeaddress=here.
--datafile	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.

Option	Short Form	Description
--realm	--realm	The name of the realm. The sub configuration will be added to the global configuration if this option is not selected.
--subconfigid	-b	The ID of the parent configuration. The sub configuration will be added to the root configuration if this option is not selected.
--priority	-p	The priority of the sub configuration.

delete-svc

Delete the service from the server.

```
ssoadm delete-svc --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--continue]	-c	Continue deleting services if one or more previous services can not be deleted
[--deletepolicyrule]	-r	Delete the policy rule.

delete-wsps

Delete web service providers.

```
ssoadm delete-wsps --options [--global-options]
```

Example:

```
# ./ssoadm delete-wsps -u amadmin -f /tmp/fampass  --wspnames wsptest

The followings were deleted.
       wsptest
```

Option	Short Form	Description
--wspnames	-s	Names of web service provider.
--adminid	-u	Administrator ID of running the command.
--password-file	-f	File name that contains password of administrator.

delete-wsp-grps

Delete web service provider groups.

```
ssoadm delete-wsp-grps --options [--global-options]
```

Example:

```
# ./ssoadm delete-wsp-grps -u amadmin -f /tmp/fampass --groupnames wspgroup

The following groups were deleted.
```

```
wspgroup
```

Option	Short Form	Description
--groupnames	-s	Names of group
--adminid	-u	Administrator ID of running the command.
--password-file	-f	File name that contains password of administrator.

do-batch

Do multiple requests in one command.

```
ssoadm do-batch --options [--global-options]
```

Option	Short Form	Description
--batchfile	-D	The filename that contains the commands and options.
--adminid	-u	The administrator ID running the command.
-password-file	-f	The filename that contains the password of the administrator.
[--batchstatus	-b	The name of the status file
[--continued	-c	Continue processing the rest of the request when the previous request was erroneous.

export-server

Export a server instance

```
ssoadm export-server --options [--global-options]
```

Option	Short Form	Description
--servername	-s	The server name. For example, http://www.example.com:8080/opensso.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--outfile]	-o	The filename where configuration is written.

export-svc-cfg

Export the service configuration.

```
ssoadm export-svc-cfg --options [--global-options]
```

Option	Short Form	Description
--encryptsecret	-e	The secret key for encrypting a password.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--outfile]	-o	The filename where configuration is written.

get-attr-defs

Get the default attribute values in a schema.

```
Get the default attribute values in a schema.
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.
[--attributenames]	-a	The names of the attribute.

get-auth-cfg-entr

Get the authentication configuration entries.

```
ssoadm get-auth-cfg-entr --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--name	-m	The name of the authentication configuration.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

get-auth-instance

Get the authentication instance values.

```
ssoadm get-auth-instance --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--name	-m	The name of the authentication instance.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

get-revision-number

Get the service schema revision number.

```
ssoadm get-revision-number --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.

Option	Short Form	Description
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

get-svrcfg-xml

Get the server configuration XML from the centralized data store.

```
ssoadm get-svrcfg-xml --options [--global-options]
```

Option	Short Form	Description
--servername	-s	The server name.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--outfile]	-o	The filename where serverconfig.XML is written.

import-server

Import a server instance.

```
ssoadm import-server --options [--global-options]
```

Option	Short Form	Description
-servername	-s	The server name.
--xmlfile	-X	The XML file that contains the configuration.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

import-svc-cfg

Import the service configuration.

```
ssoadm import-svc-cfg --options [--global-options]
```

Option	Short Form	Description
--encryptsecret	-e	The secret key for decrypting the password.
--xmlfile	-X	The XML file that contains the configuration data.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

list-auth-cfgs

List the authentication configurations.

```
ssoadm list-auth-cfgs --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

list-auth-instances

List the authentication instances.

```
ssoadm list-auth-instances --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
TBD	TBD	TBD

list-datastores

List the data stores under a realm.

```
ssoadm list-datastores --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

list-datastore-types

List the supported data store types.

```
ssoadm list-datastore-types --options [--global-options]
```

Option	Short Form	Description
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

list-server-cfg

List the server configuration.

```
ssoadm list-server-cfg --options [--global-options]
```

Option	Short Form	Description
--servername	-s	The server name.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--withdefaults]	-w	Set this flag to get the default configuration.

list-servers

List all the server instances.

```
ssoadm list-servers --options [--global-options]
```

Option	Short Form	Description
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

list-sites

List all the sites.

```
ssoadm list-sites --options [--global-options]
```

Option	Short Form	Description
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

list-wsps

Lists web service providers.

```
ssoadm list-wsps --options [--global-options]
```

Example:

```
# ./ssoadm list-wsps -u amadmin -f /tmp/fampass

 wsptest (id=wsptest,ou=agentonly,dc=opensso,dc=java,dc=net)
 wsp (id=wsp,ou=agentonly,dc=opensso,dc=java,dc=net)
```

Option	Short Form	Description
--adminid	-u	Administrator ID of running the command
--password-file	-f	File name that contains password of administrator
--filter	-x	Filter (Pattern)

list-wsp-grps

List web service provider groups.

```
ssoadm list-wsp-grps --options [--global-options]
```

Example:

```
# ./ssoadm list-wsp-grps -u amadmin -f /tmp/fampass

wspgroup
```

Option	Short Form	Description
--adminid	-u	Administrator ID of running the command
--password-file	-f	File name that contains password of administrator
--filter	-x	Filter (Pattern)

list-wsp-grp-members

List web service providers in web service provider group.

```
ssoadm list-wsp-grp-members --options [--global-options]
```

Example:

```
# ./ssoadm list-wsp-grp-members -u amadmin -f /tmp/fampass --groupname wspgroup

wsptest (id=wsptest,ou=agent,dc=opensso,dc=java,dc=net)
```

Option	Short Form	Description
--groupname	-b	Name of web service provider group
--adminid	-u	Administrator ID of running the command
--password-file	-f	File name that contains password of administrator
--filter	-x	Filter (Pattern)

register-auth-module

Register an authentication module.

```
ssoadm register-auth-module --options [--global-options]
```

Option	Short Form	Description
--authmodule	-a	The Java class name of the authentication module.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

remove-attr-choicevals

Remove choice values from the attribute schema.

```
ssoadm remove-attr-choicevals --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.

Option	Short Form	Description
--schematype	-t	The type of schema.
--attributename	-a	The name of the attribute.
--choicevalues	-k	The choice values. For example, inactive.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

remove-attr-defs

Remove the default attribute values in a schema.

```
ssoadm remove-attr-defs --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributenames	-a	The names of the attribute.
--adminid	-u	The administrator ID running the command
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

remove-server-cfg

Remove the server configuration.

```
ssoadm remove-server-cfg --options [--global-options]
```

Option	Short Form	Description
--servername	-s	The server name. For example, http://www.example.com:8080/opensso.
--propertynames	-a	The names of the properties to be removed.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

remove-site-members

Remove members from a site.

```
ssoadm remove-site-members --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The site name. For example, mysite.

Option	Short Form	Description
--servernames	-e	The server name. For example, http://www.example.com:8080/opensso.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

remove-site-sec-urls

Remove the site secondary URLs.

```
ssoadm remove-site-sec-urls --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The site name. For example, mysite.
--secondaryurls	-a	The secondary URLs.
--adminid	-f	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

remove-sub-schema

Remove the sub schema.

```
ssoadm remove-sub-schema --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--subschemanames	-a	The names of the sub schema to be removed.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
--subschemaname	-c	The name of the parent sub schema.

remove-wsp-from-grp

Remove web service providers from a group.

```
ssoadm remove-wsp-from-grp --options [--global-options]
```

Example:

```
# ./ssoadm remove-wsp-from-grp -u amadmin -f /tmp/fampass  --groupname wspgroup
--wspnames wsptest

Provider was removed from group.
```

Option	Short Form	Description
--groupname	-b	Name of group.

Option	Short Form	Description
--wspnames	-s	Names of web service providers.
--adminid	-u	Administrator ID of running the command.
--password-file	-f	File name that contains password of administrator

set-attr-any

Set any member of the attribute schema.

```
ssoadm set-attr-any --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschema	-a	The name of the attribute schema.
--any	-y	The attribute schema. Any value.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-attr-bool-values

Set the boolean values of the attribute schema.

```
ssoadm set-attr-bool-values --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributename	-a	The name of the attribute.
--truevalue	-e	The value for true.
--truei18nkey	-k	The internationalization key for the true value.
--falsevalue	-z	The value for false.
--falsei18nkey	-j	The internationalization key for the false value.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-attr-choicevals

Set choice values for the attribute schema.

```
ssoadm set-attr-choicevals --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributename	-a	The name of the attribute.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--add]	-p	Set this flag to append the choice values to existing ones.
[--subschemaname]	-c	The name of the sub schema.
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.
[--choicevalues]	-k	The choice values. For example, 0102=Inactive.

set-attr-defs

Set the default attribute values in a schema.

```
ssoadm set-attr-defs --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.
[--attributevalues]	-a	The attribute values. For example, homeaddress=here.
[--datafile]	-D]	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.

set-attr-end-range

Set the attribute schema end range.

```
ssoadm set-attr-end-range --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschema	-a	The name of the attribute schema.
--range	-r	The end range.
--adminid	-u	The administrator ID running the command.

Option	Short Form	Description
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-attr-i18n-key

Set the i18nkey member of the attribute schema.

```
ssoadm set-attr-i18n-key --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschema	-a	The name of the attribute schema.
--i18nkey	-k	The attribute schema i18n key.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-attr-start-range

Set attribute schema start range.

```
ssoadm set-attr-start-range --options [--global-options]
```

Options	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschema	-a	The name of the attribute schema.
--range	-r	The start range.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-attr-syntax

Set syntax member of attribute schema.

```
ssoadm set-attr-syntax --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.

Option	Short Form	Description
--attributeschema	-a	The name of the attribute schema.
--syntax	-x	The attribute schema syntax.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-attr-type

Set the type member of the attribute schema.

```
ssoadm set-attr-type --options [--global-options
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschema	-a	The name of the attribute schema.
--type	-p	The attribute schema type.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-attr-ui-type

Set the UI type member of the attribute schema.

```
ssoadm set-attr-ui-type --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschema	-a	The name of the attribute schema.
--uitype	-p	The attribute schema UI type.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-attr-validator

Set the attribute schema validator.

```
ssoadm set-attr-validator --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschema	-a	The name of the attribute schema.
--validator	-r	The validator class name.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-attr-view-bean-url

Set the properties view bean URL member of the attribute schema.

```
ssoadm set-attr-view-bean-url --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--attributeschema	-a	The name of the attribute schema.
--url	-r	The attribute schema properties view bean URL.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--subschemaname]	-c	The name of the sub schema.

set-inheritance

Set the inheritance value of the sub schema.

```
ssoadm set-inheritance --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--schematype	-t	The type of schema.
--subschemaname	-c	The name of the sub schema.
--inheritance	-r	The value of inheritance.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

set-plugin-viewbean-url

Set the properties view bean URL of the plug-in schema.

```
ssoadm set-plugin-viewbean-url --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--interfacename	-i	The name of the interface.
--pluginname	-g	The name of the plug-in.
--url	-r	The properties view bean URL.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

set-revision-number

Set the service schema revision number.

```
ssoadm set-revision-number --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--revisionnumber	-r	The revision number.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

set-site-pri-url

Set the primary URL of a site.

```
ssoadm set-site-pri-url --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The site name. For example, mysite.
--siteurl	-i	The site's primary URL. For example, http://www.example.com:8080.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

set-site-sec-urls

Set the site secondary URLs.

```
ssoadm set-site-sec-urls --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The site name. For example, mysite.
--secondaryurls	-a	The secondary URLs.
--adminid	-u	The administrator ID running the command.

Option	Short Form	Description
--password-file	-f	The filename that contains the password of the administrator.

set-sub-cfg

Set the sub configuration.

```
ssoadm set-sub-cfg --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--subconfigname	-g	The name of the sub configuration.
--operation	-o	The operation (either add/set/modify) to be performed on the sub configuration.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--attributevalues]	-a	The attribute values. For example, `homeaddress=here`.
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.
[--realm]	-e	The name of the realm. The sub configuration will be added to the global configuration if this option is not selected.

show-wsp

Shows web service provider.

```
ssoadm show-wsp --options [--global-options]
```

Example:

```
# ./ssoadm show-wsp -u amadmin -f /tmp/fampass --wspname wsptest

securitymech=urn:sun:wss:security:null:SAMLToken-HK
publickeyalias=test1
endpoint=Default
includememberships=true
nameidmapper=nameidmapper.class
attributenamespace=123
samlattributemapping=abc=xyz
```

Option	Short Form	Description
--wspname	-b	Name of web service provider
--password-file	-f	File name that contains password of administrator
--outfile	-o	Filename where configuration is written to
--inherit	-i	Set this to inherit properties from parent group

set-svc-i18n-key

Set the service schema i18n key.

```
ssoadm set-svc-i18n-key --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--i18nkey	-k	The i18n key.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

set-svc-view-bean-url

Set the service schema properties view bean URL.

```
ssoadm set-svc-view-bean-url --options [--global-options]
```

Option	Short Form	Description
--servicename	-s	The name of the service.
--url	-r	The service schema properties view bean URL.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

set-svrcfg-xml

Set the server configuration XML to the centralized data store.

```
ssoadm set-svrcfg-xml --options [--global-options]
```

Option	Short Form	Description
--servername	-s	The server name.
--xmlfile	-X	The XML file that contains the configuration.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--outfile]	-o	The filename where serverconfig XML is written.

show-auth-modules

Show the supported authentication modules in the system.

```
ssoadm show-auth-modules --options [--global-options]
```

Option	Short Form	Description
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

show-datastore

Show the data store profile.

```
ssoadm show-datastore --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--name	-m	The name of the datastore.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

show-site

Show the site profile.

```
ssoadm show-site --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The site name. For example, mysite.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

show-site-members

Display the members of a site.

```
ssoadm show-site-members --options [--global-options]
```

Option	Short Form	Description
--sitename	-s	The site name. For example, mysite.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

show-wsp-grp

how web service provider group profile.

```
ssoadm show-wsp-grp --options [--global-options]
```

Example:

```
# ./ssoadm show-wsp-grp -u amadmin -f /tmp/fampass  --groupname wspgroup

securitymech=urn:sun:wss:security:null:SAMLToken-HK
publickeyalias=test1
endpoint=Default
includememberships=false
nameidmapper=nameidmapper.class
attributenamespace=123
samlattributemapping=abc=xyz
```

Option	Short Form	Description
--groupname	-b	Name of web service provider group
--adminid	-u	Administrator ID of running the command
--password-file	-f	File name that contains password of administrator
--outfile	-o	Filename where configuration is written to

show-wsp-membership

List web service provider's membership.

```
ssoadm show-wsp-membership --options [--global-options]
```

Example:

```
# ./ssoadm show-wsp-membership -u amadmin -f /tmp/fampass  --wspname wsptest

This provider belongs to wspgroup
(id=wspgroup,ou=agentgroup,dc=opensso,dc=java,dc=net).
```

Option	Short Form	Description
--wspname	-b	Name of web service provider
--adminid	-u	Administrator ID of running the command
--password-file	-f	File name that contains password of administrator

unregister-auth-module

Unregister the authentication module.

```
ssoadm unregister-auth-module --options [--global-options]
```

Option	Short Form	Description
--authmodule	-a	The Java class name of the authentication module.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

update-auth-cfg-entr

Set the authentication configuration entries.

```
ssoadm update-auth-cfg-entr --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--name	-m	The name of the authentication configuration.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.

Option	Short Form	Description
[--entries]	-a	The formatted authentication configuration entries.
[--datafile]	-D	The filename that contains the formatted authentication configuration entries. Enter one attribute-name=attribute-value per line.

update-auth-instance

Update the authentication instance values.

```
ssoadm update-auth-instance --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--name	-m	The name of the authentication instance.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--attributevalues]	-a	The attribute values. For example, homeaddress=here.
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.

update-datastore

Update the datastore profile.

```
ssoadm update-datastore --options [--global-options]
```

Option	Short Form	Description
--realm	-e	The name of the realm.
--name	-m	The name of the datastore.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--attributevalues]	-a	The attribute values. For example, sunIdRepoClass=com.sun.identity.idm.plugins.files.FilesRepo.
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.

update-server-cfg

Update the server configuration.

```
ssoadm update-server-cfg --options [--global-options]
```

Option	Short Form	Description
--servername	-s	The server name.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--attributevalues]	-a	The attribute values. For example, homeaddress=here.
[--datafile]	-D	Name of file that contains attributes and corresponding values as in attribute-name=attribute-value. Enter one attribute and value per line.

update-svc

Update the service.

```
ssoadm update-svc --options [--global-options]
```

Option	Short Form	Description
--xmlfile	-X	The XML file that contains the schema.
--adminid	-u	The administrator ID running the command.
--password-file	-f	The filename that contains the password of the administrator.
[--continue]	-c	Continue updating services if one or more previous services can not be updated.

update-wsp

Update web service provider.

```
ssoadm update-wsp --options [--global-options]
```

Example:

```
# ./ssoadm update-wsp -u amadmin -f /tmp/fampass --endpoint newendpoint -b wsptest

Web service provider was updated.
```

Option	Short Form	Description
--wspname	-b	Name of web service provider.
--adminid	-u	Administrator ID of running the command
--password-file	-f	File name that contains password of administrator
--securitymech	-y	Security mechanism
--endpoint	-e	Web service provider's end point
--publickeyalias	-a	Public key alias
--samlattributemapping	-t	SAML Attribute Mapping
--nameidmapper	-i	SAML NameID Mapper Plugin
--attributenamespace	-p	Attribute Namespace

Option	Short Form	Description
--includememberships	-m	Include Memberships. Possible values are true or false.
--set	-s	Set this flag to overwrite properties values.

update-wsp-grpd

Update web service provider group configuration.

```
ssoadm update-wsp-grp --options [--global-options]
```

Example:

```
# ./ssoadm update-wsp-grp -u amadmin -f /tmp/fampass --groupname wspgroup
--publickeyalias testtest

Web service provider group configuration was updated.
```

Option	Short Form	Description
--groupname	-b	Name of web service provider group
--adminid	-u	Administrator ID of running the command
--password-file	-f	File name that contains password of administrator
--securitymech	-y	Security mechanism
--endpoint	-e	Web service provider's end point
--publickeyalias	-a	Public key alias
--samlattributemapping	-t	SAML Attribute Mapping
--nameidmapper	-i	SAML NameID Mapper Plugin
--attributenamespace	-p	Attribute Namespace
--includememberships	-m	false]
--set	-s	et this flag to overwrite properties values.

wsp-remove-propsd

Remove web service provider's properties.

```
ssoadm wsp-remove-props --options [--global-options]
```

Example:

```
# ./ssoadm wsp-remove-props -u amadmin -f /tmp/fampass  --wspname wsptest
--attributenames  includememberships

Properties were removed.
```

Option	Short Form	Description
--wspname	-b	Name of web service provider
--attributenames	-a	properties name(s). They are securitymech, endpoint publickeyalias samlattributemapping nameidmapper attributenamespace and includememberships.
--adminid	-u	Administrator ID of running the command

Option	Short Form	Description
--password-file	-f	File name that contains password of administrator

B

Debugging and Troubleshooting OpenSSO STS

This chapter contains the following topics:

- Debugging OpenSSO STS
- Troubleshooting OpenSSO STS Issues

B.1 Debugging OpenSSO STS

Set debug properties when you configure an OpenSSO STS server instance. See Section 5.1.3.1, "To Configure OpenSSO STS Server General Properties."

OpenSSO Security Token Service (OpenSSO STS) debug files are stored in the WebServices file.

B.2 Troubleshooting OpenSSO STS Issues

The following are error conditions or error messages and troubleshooting tips you can try:

Time stamp is invalid.
Make sure that all host systems are in sync. The default skew allowed is 10 seconds. You can reconfigure this setting.

Unsupported security mechanism
The security mechanism identified in the request does not match with one of the configured security mechanisms.

Authentication failed.
Make sure that your credentials are correctly provisioned in OpenSSO STS under User Credential. If configured to authenticate at Oracle Internet Directory or at Oracle Virtual Directory, then make sure the authentication chain is enabled in OpenSSO STS.

Decryption failed, or signing validation failed.
The encryption/decryption settings should be identical among client and server. The following are typical recommendations:

- For asymmetric or symmetric binding, enable request and response signing of both body and header, and enable request decryption and response encyrption.
- For transport-layer binding, disable signature validation when SSL is used; disable encryption when SSL is used.