Sun OpenSSO Enterprise 8.0 Update 1 Release Notes

Part No: 821–1818
April 13, 2010

ORACLE®

100414@23626

Contents

CHAPTER 1

About OpenSSO Enterprise 8.0 Update 1

This chapter describes **OpenSSO Enterprise 8.0 Update 1**, including:

- "What's New in OpenSSO Enterprise 8.0 Update 1" on page 9
- "Hardware and Software Requirements For OpenSSO Enterprise 8.0 Update 1" on page 16
- "OpenSSO Enterprise 8.0 Update 1 Issues and Workarounds" on page 17
- "OpenSSO Enterprise 8.0 Update 1 Documentation" on page 21
- "OpenSSO Enterprise 8.0 Update 1 Patch Releases" on page 21
- "Additional Information and Resources" on page 28
- "Revision History" on page 30

What's New in OpenSSO Enterprise 8.0 Update 1

OpenSSO Enterprise 8.0 Update 1 also fixes a number of problems, as listed in the README file included with patch 141655-01.

- "OpenDS as a User Data Store" on page 9
- "Simplified OpenSSO WAR File Creation" on page 10
- "Centralized SAMLv2 Error Conditions Page" on page 10
- "Secure Attribute Exchange (SAE) Data Encryption" on page 10
- "FIPS Compliance Mode" on page 10
- "Support for New Web Containers" on page 10
- "OpenDS as a User Data Store" on page 11
- "ASP.NET Fedlet" on page 11
- "Other Enhancements in OpenSSO Enterprise 8.0 Update 1" on page 11

OpenDS as a User Data Store

You can configure an external OpenDS server as the OpenSSO Enterprise 8.0 Update 1 user data store.

You can also store a relatively small number of users in the embedded OpenSSO configuration data store (OpenDS), when scalability is not an important requirement. This option is useful when you want to install OpenSSO Enterprise 8.0 Update 1 quickly for demonstration or evaluation purposes. However, you should not use an embedded OpenDS server as a user data store in a production environment.

See Chapter 9, "Using OpenDS as a User Data Store for OpenSSO Enterprise 8.0 Update 1."

Simplified OpenSSO WAR File Creation

The ability to create a specialized WAR file was present in OpenSSO Enterprise 8.0. In OpenSSO Enterprise 8.0 Update 1, the process has been simplified using the `createwar.sh` or `createwar.bat` script.

See Chapter 4, "Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File."

Centralized SAMLv2 Error Conditions Page

OpenSSO Enterprise 8.0 Update 1 provides a single page where you can view all SAMLv2 error conditions. This page is useful when you are troubleshooting a SAMLv2 configuration.

See Chapter 6, "Centralizing SAML Error Display in OpenSSO Enterprise 8.0 Update 1."

Secure Attribute Exchange (SAE) Data Encryption

OpenSSO Enterprise 8.0 Update 1 supports Secure Attributes Exchange (SAE) data encryption. (SAE is also known as Virtual Federation.)

See Chapter 7, "Encrypting Data in a Secure Attribute Exchange in OpenSSO Enterprise 8.0 Update 1."

FIPS Compliance Mode

OpenSSO Enterprise 8.0 Update 1 supports Federal Information Processing Standards (FIPS) mode.

See Chapter 8, "Configuring OpenSSO Enterprise 8.0 Update 1 in FIPS Mode."

Support for New Web Containers

OpenSSO Enterprise 8.0 Update 1 supports the web containers described in "Web Containers Supported For OpenSSO Enterprise 8.0" in *Sun OpenSSO Enterprise 8.0 Release Notes* and the following new web containers:

- IBM WebSphere Application Server 7.0. See Chapter 5, "Deploying IBM WebSphere Application Server 7.0 as the OpenSSO Enterprise 8.0 Update 1 Web Container."
- Oracle WebLogic Server 10g Release 3 (10.3)
- GlassFish Prelude 3

OpenDS as a User Data Store

OpenSSO Enterprise 8.0 Update 1 supports OpenDS to store user profiles, authentication data, and policies.

See Chapter 9, "Using OpenDS as a User Data Store for OpenSSO Enterprise 8.0 Update 1."

ASP.NET Fedlet

OpenSSO Enterprise 8.0 Update 1 includes the `Fedlet.dll`, template metadata files, and a sample application for implementing the Fedlet with ASP.NET applications. See Chapter 10, "Using the ASP.NET Fedlet with OpenSSO Enterprise 8.0 Update 1."

Other Enhancements in OpenSSO Enterprise 8.0 Update 1

- "CR 6244578: New Property Warns Users if Browser Cookie Support is Disabled or Not Available" on page 12
- "CR 6770231: OpenSSO Enterprise 8.0 Update 1 Validates goto URLs" on page 12
- "CR 6696910: New Property makes Event Notification Cache Configurable" on page 13
- "CR 6740071: New Property Controls Session Cookie for Zero Page Authentication" on page 13
- "CR 6691106: New Properties Prevent Multiple Site Monitor Threads" on page 14
- "CR 6797423: New property configures OpenSSO Enterprise server policy decision cache" on page 14
- "CR 6785321: CRL and OSCP checking support JSS-based logic" on page 14
- "CR 6657112: Redirect callback support is added for Distributed Authentication Server UI" on page 15
- "CR 6657367: CDCServlet removes the JavaScript enabled dependency for user's browser" on page 15
- "CR 6496155: Policy agents send token other than the IP address in cookie hijacking mode" on page 15
- "CR 6697260: New property allows policy agent sessions to time out" on page 15
- "CR 6811036: After upgrading from JES4, in co-existence mode, amadmin authenticates to configuration data store" on page 16
- "CR 6827616: SMS cache is disabled by default for the Client SDK" on page 16

CR 6244578: New Property Warns Users if Browser Cookie Support is Disabled or Not Available

The new `com.sun.identity.am.cookie.check` property indicates whether OpenSSO server should check if cookie support is disabled or not available in the user's browser. A value of true causes OpenSSO server to display an error message if the browser does not support cookies or has not enabled cookies.

Previously, if cookie support was disabled or not available on the user's browser and OpenSSO server was not in cookieless mode, authentication for a user failed without any errors. (Actually, authentication was done successfully, but OpenSSO server could not redirect the user to the OpenSSO protected web site.)

To Set the Property

1. Log in to the OpenSSO Administation Console.
2. Click Configuration, Servers and Sites, *opensso-instance-name*, and then Advanced.
3. Click Add and then specify:
   - Property Name: `com.sun.identity.am.cookie.check`
   - Property Value: true or false
4. Click Save.
5. Restart the OpenSSO server instance.

Note - If OpenSSO server is expected to support cookieless mode for authentication, set this property to false (which is the default).

CR 6770231: OpenSSO Enterprise 8.0 Update 1 Validates goto URLs

OpenSSO Enterprise 8.0 Update 1 can validate a goto URL after a user logs in to prevent a hacker from sending the user to an imposter site in order to steal the user's personal information.

To Set Valid goto URLs:

1. Install OpenSSO Enterprise 8.0 Update 1. If you are patching OpenSSO Enterprise 8.0, make sure you run the `updateschmema.sh` or `updateschema.bat` script and restart the OpenSSO Enterprise web container.
2. Log in to the Admin Console.
3. Click Configuration, Authentication, and then Core.
4. Under Valid goto URL domains, add each valid goto domain name, as follows:
   - A domain name starting with a dot (.) such as `.example.com` allows all hosts in the `example.com` domain to be used in a success redirect URL.

- A domain name that does not start with a dot (.) such as `example.com` allows the host `example.com` to be used in a success redirect URL. For example, *http://example.com* would be valid, but *http://host.example.com* would not be valid.
- If you don't add the entire domain to the list, you must add each individual agent host name being used.
- You do not need to add domains for agents in CDSSO mode, because they are protected automatically.

5. Click Save.
6. Restart the OpenSSO Enterprise web container.

   If you subsequently want to disable the goto URL validation, remove all entries from the Valid goto URL domains list.

Additional Information - If a goto URL is found to be invalid, the user will be redirected to the default success login URL (`/opensso/console`).

CR 6696910: New Property makes Event Notification Cache Configurable

The new `com.sun.am.event.notification.expire.time` property allows you to configure or disable the event notification cache in order to improve performance.

To disable the cache, set this property to 0 (zero). The default is 30 minutes.

After you set this property, restart the OpenSSO Enterprise 8.0 web container for the new value to take effect.

CR 6740071: New Property Controls Session Cookie for Zero Page Authentication

The new `com.sun.identity.appendSessionCookieInURL` property determines whether OpenSSO Enterprise 8.0 Update 1 ppends the session cookie to the URL for zero page authentication.

Set this property to false to prevent OpenSSO Enterprise 8.0 Update 1 from appending the session cookie to the URL. For example, if an application is filtering incoming URLs for special characters for security reasons and a cookie contains a special character, then access is denied. The default value is true (cookie is appended).

To set the new `com.sun.identity.appendSessionCookieInURL` property:

1. Log in to the OpenSSO Enterprise 8.0 Update 1 Admin Console.
2. Click Configuration, Servers and Sites, Default Server Settings, and then Advanced.
3. Add the property with a value of `true`.
4. Click Save.

The `com.sun.identity.appendSessionCookieInURL property` is hotswappable, which means that you don't have to restart the OpenSSO Enterprise 8.0 web container for a new value to take effect.

CR 6691106: New Properties Prevent Multiple Site Monitor Threads

The `amNaming` log sometimes indicates multiple Site Monitor threads running for checking the same site. To prevent this problem, OpenSSO Enterprise 8.0 Update 1 provides improved synchronization to prevent the creation of the multiple Site Monitor threads for the same site. OpenSSO Enterprise 8.0 also includes these new properties:

- `com.sun.identity.urlchecker.retry.interval` specifies the time interval in milliseconds between retries for a URL connection. Default is 500 milliseconds (0.5 seconds).
- `com.sun.identity.urlchecker.retry.limit` specifies the maximum number of retries for the URL connection if a connection failure occurs. Default is 3 retries.

After you set these properties, restart the OpenSSO Enterprise 8.0 web container for the new values to take effect.

The fix for this problem also uses the following property:

- `com.sun.identity.urlchecker.sleep.interval` specifies the time interval in milliseconds that the site status check should sleep. Default is 30000 milliseconds (30 seconds).

CR 6797423: New property configures OpenSSO Enterprise server policy decision cache

The new `com.sun.identity.policy.resultsCacheMaxSize` property allows you to configure the policy decision cache for OpenSSO Enterprise 8.0 Update 1 server.

For example, a value of 1000 causes policy decisions to be cached for maximum of 1000 sessions, irrespective of the actual number of concurrent sessions on the server.

CR 6785321: CRL and OSCP checking support JSS-based logic

Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP) checking now support the Network Security Services for Java (JSS) library, enabling FIPS mode when OpenSSO Enterprise 8.0 Update 1 is deployed on the Sun Java System Web Server 7.0 Update 3 or later web container.

Note - FIPS compliance mode depends on JSS, but using JSS does not necessitate FIPS compliance mode.

CR 6657112: Redirect callback support is added for Distributed Authentication Server UI

Redirect callback support (`RedirectCallback`), which is used to redirect users to an external website as part of the authentication process, now works when the login is through a Distributed Authentication Server UI.

CR 6657367: CDCServlet removes the JavaScript enabled dependency for user's browser

If cross-domain single sign-on (CDSSO) is enabled for a policy agent, the `CDCServlet` can now redirect assertions (`CDCRedirectServlet`) for the agent, even if JavaScript is disabled for the user's browser.

CR 6496155: Policy agents send token other than the IP address in cookie hijacking mode

Previously, in cookie hijacking mode, policy agents sent the IP address of the server where they were installed to the OpenSSO Enterprise server. Now, the policy agent first sends the application SSO token. If the agent cannot obtain the application SSO token, the agent then sends the IP address to the OpenSSO Enterprise server.

If strict DN checking is required for a deployment, OpenSSO Enterprise server includes the new

`iplanet-am-session-dnrestrictiononly` property.

The default value is `false`. If this property is set to `true`, the OpenSSO Enterprise server performs strict DN checking. If the agent sends an IP address, the OpenSSO Enterprise server considers the IP address to be an error.

To set `iplanet-am-session-dnrestrictiononly` for strict DN checking:

1. Add the property with a value of `true` using either the OpenSSO Enterprise Admin Console or the `ssoadm` utility.
2. Restart the OpenSSO Enterprise server web container for the DN checking to take effect.

CR 6697260: New property allows policy agent sessions to time out

The new `com.iplanet.am.session.agentsessionidletime` property sets the maximum idle timeout in minutes for policy agent sessions. The minimum value is 30 minutes. A value greater than 0 and less than 30 will be reset to 30.

The default is 0, which means that the policy agent sessions never time out.

To set `com.iplanet.am.session.agentsessionidletime`:

1. Add the property with the maximum idle timeout value using either the OpenSSO Enterprise Admin Console or the `ssoadm` utility.

2. Restart the OpenSSO server web container for the idle timeout value to take effect.

CR 6811036: After upgrading from JES4, in co-existence mode, amadmin authenticates to configuration data store

Due to the fix for security issue 3924 in OpenSSO 8.0 Enterprise 8.0, the `amadmin` user was prevented from logging in to any authentication module other than the DataStore and Application authentication modules.

This new fix for CR 6811036 removes this restriction, but at the same time re-implements the original security fix to protect the authentication as the `amadmin` user, which is considered as the OpenSSO Enterprise internal or special user, in following manner:

- `amadmin` can authenticate only to or or the Top-Level Realm.
- `amadmin` and its password will first be authenticated against the configuration data store. That is, this user and its password should match the `amadmin` user and its password in the OpenSSO Enterprise configuration data store. Then, this user will be authenticated against the required authentication store (authentication module) with the same credentials. Finally, this user will be retrieved (searched) in the OpenSSO Enterprise user data store (based on the user profile option selected in the Authentication service configuration).

  The actual authentication module store and/or user data store and configuration data store could be different, as long as the above is successful. If all three stores are the same, the above would be automatically successful.

CR 6827616: SMS cache is disabled by default for the Client SDK

After a Client SDK installation, the service management service (SMS) cache is disabled by default, which can cause performance issues.

Workaround: To enable the cache for SMS and the Identity Repository (IdRepo), set or add the following properties in the `AMClient.properties` file:

```
com.iplanet.am.sdk.caching.enabled=true
com.sun.identity.idm.cache.enabled=true
com.sun.identity.sm.cache.enabled=true
```

Hardware and Software Requirements For OpenSSO Enterprise 8.0 Update 1

Note - The hardware and software requirements for OpenSSO Enterprise 8.0 Update 1 represent the only environments in which it can be deployed with full support from Oracle. No support is provided for environments that do not meet the stated requirements.

Oracle assumes no responsibility or liability for any environments that don't adhere to supported hardware and software requirements for OpenSSO Enterprise 8.0 Update 1 as documented. Oracle strongly recommends that you involve the Professional Services organization before you begin the installation and deployment process. This may require additional expense on your part.

Policy Agent Support in OpenSSO Enterprise 8.0 Update 1

Policy Agent Version	OpenSSO Enterprise 8.0 Update 1 Support
3.0	Version 3.0 Java EE (formerly called J2EE) and web policy agents are supported, including new version 3.0 features. For more information, including the available version 3.0 agents, see `http://docs.sun.com/coll/1767.1`.
2.2	Version 2.2 Java EE and web policy agents are supported. However, a version 2.2 policy agent must continue to use version 2.2 features. For example, the OpenSSO Enterprise centralized agent configuration is not supported, and the 2.2 agent must store its configuration data locally in its `AMAgent.properties` file. For more information, including the available version 2.2 agents, see `http://docs.sun.com/coll/1322.1`.
2.1	Version 2.1 policy agents are **not** supported.

OpenSSO Enterprise 8.0 Update 1 Issues and Workarounds

- “CR 6828741: Configuring OpenSSO Enterprise 8.0 Update 1 as site throws exception in debug logs” on page 21
- “CR 6833362: SAMLv2 returns error on WebLogic Server 10 with SOAP binding” on page 21

CR 6830298: OpenSSO Enterprise Admin Tools Must be Re-installed

If you patch OpenSSO Enterprise 8.0 with Update 1, you must re-install the admin tools in Update 1 before you run the `updateschema.sh` or `updateschema.bat` script, because the script requires the Update 1 version of the `ssoadm` command-line utility.

Workaround. Before you run the `updateschema.sh` or `updateschema.bat` script, install the Update 1 admin tools, as described in Chapter 3, “Installing the OpenSSO Enterprise 8.0 Update 1 Admin Tools.”

CR 6823779: `ssoadm` cannot be used with Secure WebSphere Application Server 7.0

If the admin tools (`ssoAdminTools.zip`) are configured to use the IBM JVM with a secure (SSL-enabled) WebSphere Application Server 7.0 instance, the `ssoadm` returns a fatal error.

Workaround. To configure `ssoadm`, see Chapter 5, “Deploying IBM WebSphere Application Server 7.0 as the OpenSSO Enterprise 8.0 Update 1 Web Container.”

CR 6824420: Configuration fails for WebSphere Application Server 7.0 with Java 2 security enabled

If OpenSSO Enterprise 8.0 Update 1 is deployed with IBM WebSphere Application Server 7.0 and Java 2 security is enabled, the configuration fails.

Workaround. Add the required permissions to the WebSphere Application Server 7.0 `server.policy`. For more information see Chapter 5, “Deploying IBM WebSphere Application Server 7.0 as the OpenSSO Enterprise 8.0 Update 1 Web Container.”

CR 6836470: Hotfix Required to Use KDCs Hosted on Windows Server 2008

OpenSSO Enterprise 8.0 Update 1 has added support for using KDCs hosted on Windows Server 2008. To use this new feature, however, you must install a Microsoft hotfix to KTpass on the Windows Server 2008 KDC before using the KDC for Windows Desktop SSO authentication.

For more information and to download this hotfix, see `http://support.microsoft.com/kb/951191`.

CR 6825011: Windows Desktop SSO Authentication fails with Login Exception on WebSphere Application Server 7.0

Workaround. If OpenSSO Enterprise 8.0 Update 1 is deployed on IBM WebSphere Application Server 7.0 on Windows:

1. Prefix the Keytab File Name property of the Windows Desktop SSO authentication module instance with `file:///`. For example:

   ```
   file:///C:/keytabs/ssohost-4100-04.HTTP.keytab
   ```

2. Set the new `com.sun.identity.authentication.module.WindowsDesktopSSO.Krb5LoginModule` property to `com.ibm.security.auth.module.Krb5LoginModule`.

Set this new property using `ssoadm` or in the OpenSSO Enterprise Admin Console under Configuration, Sites and Server, *opensso-instance-name*, and Advanced. Then, restart the WebSphere Application Server 7.0 instance for the value to take effect.

CR 6831600: Configurator buttons are not visible using Safari on a Mac

When running the Configurator using Safari on a Mac, the Next and Cancel buttons are not visible, which gives the impression that the configuration cannot continue.

Workaround. Maximize the Safari browser to the fullest extent and scroll down to see the buttons.

CR 6819848: Berkeley DB client does not failover to secondary Message Queue broker

In a session failover configuration, the Berkeley DB client does not failover to the secondary Message Queue broker. OpenSSO Enterprise server, however, does failover

to the secondary broker, which causes the queue on that broker to quickly fill up. Then, the broker blocks the producer from sending any more messages, which in turn blocks messages from OpenSSO Enterprise server.

CR 6834714: Permissions need updating for WebSphere Application Server 6.1

If you are using IBM WebSphere Application Server 6.1 as the web container and the Java Security Manager is enabled, the securing permissions need to be updated.

Workaround. For the correct permissions, see the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.

CR 6835816: After you enable FIPS mode, bootstrap file cannot be decrypted

Workaround. Before you enable FIPS mode, backup the bootstap file. Then, after you enable FIPS mode, replace the bootstrap file with the backup copy.

For more information, see Chapter 8, "Configuring OpenSSO Enterprise 8.0 Update 1 in FIPS Mode."

CR 6831687: SAML2 post profile fails on the Service Provider (SP)

Using JDK 1.6.x, when a Service Provider (SP) tries to verify a signed SAML2 response/assertion, the Identity Provider (IDP)throws a Null Pointer Exception.

Workaround. This problem occurs because JDK 1.6.x includes an older version of the XML security library. To fix this problem:

1. Create an endorsed directory in JDK 1.6.x. For example:

   JDK_1.6_HOME_DIR`/jre/lib/endorsed`
2. Copy the `xmlsec.jar` file from the *OpenSSO_WAR_extracted_dir*`/WEB-INF/lib` directory to the `endorsed` directory.

3. Restart the OpenSSO Enterprise 8.0 web container.

CR 6828741: Configuring OpenSSO Enterprise 8.0 Update 1 as site throws exception in debug logs

When you configure OpenSSO Enterprise 8.0 Update 1 using the console, if you provide the site details such as the load balancer and server instances, the configuration finishes successfully and you can log in. However, the debug logs contain an exception.

Workaround. None. You can ignore the exception.

CR 6833362: SAMLv2 returns error on WebLogic Server 10 with SOAP binding

If you deploy OpenSSO Enterprise 8.0 Update 1 on WebLogic Server 10 for both the SP and IDP, configure the meta for SP and IDP for signing and encryption using the default keystore, and then terminate with SOAP binding, an error is returned.

Workaround. Remove last two lines from `idpArtifactResolution.jsp`, `idpMNISOAP.jsp`, and `spMNISOAP.jsp`. Also, remove any empty spaces between `%>` and `<%`.

OpenSSO Enterprise 8.0 Update 1 Documentation

In addition to this document, additional OpenSSO Enterprise 8.0 documentation is available in the following collection:

`http://docs.sun.com/coll/1767.1`

OpenSSO Enterprise 8.0 Update 1 Patch Releases

- "OpenSSO Enterprise 8.0 Update 1 Patch IDs" on page 21
- "OpenSSO Enterprise 8.0 Update 1 Patch 3 (Patch ID 141655-04)" on page 22
- "OpenSSO Enterprise 8.0 Update 1 Patch 2 (141655-03)" on page 27

OpenSSO Enterprise 8.0 Update 1 Patch IDs

Oracle periodically releases patches for OpenSSO Enterprise 8.0 on `http://sunsolve.sun.com/`. The following table shows the patch IDs for OpenSSO Enterprise 8.0 Update 1 and subsequent patch releases.

Release	Patch ID
OpenSSO Enterprise 8.0 Update 1 Patch 3	141655-04
OpenSSO Enterprise 8.0 Update 1 Patch 2	141655-03
OpenSSO Enterprise 8.0 Update 1 Patch 1	141655-02
OpenSSO Enterprise 8.0 Update 1	141655-01

To download the latest patch, click Download Latest Patch 141655.

To determine if you should install a patch, check this document and the README file available with the patch.

OpenSSO Enterprise 8.0 Update 1 Patch 3 (Patch ID 141655-04)

New Features in OpenSSO Enterprise 8.0 Update 1 Patch 3

Message Queue is upgraded from 4.3 to 4.4 (CR 6900482)

In Patch 3, Message Queue 4.3 has been upgraded to GlassFish Message Queue 4.4. This upgrade improves OpenSSO Enterprise performance and addresses several issues with session failover deployments.

For the Message Queue documentation, see `http://docs.sun.com/coll/1307.7`.

OpenSSO Enterprise session cookies can be marked as HTTPOnly (CR 6843487)

Patch 3 includes the new `com.sun.identity.cookie.httponly` property to allow OpenSSO Enterprise session cookies to be marked as HTTPOnly, in order to prevent scripts or third-party programs from accessing the cookies. Specifically, session cookies marked as HTTPOnly can help to prevent cross-site scripting (XSS) attacks.

By default, the value for `com.sun.identity.cookie.httponly` is `false`. To set this new property, use the OpenSSO Administration Console:

1. Log in to the OpenSSO Administration Console.
2. Click Configuration, Servers and Sites, *opensso-instance-name*, and then Advanced.
3. Add `com.sun.identity.cookie.httponly` with a value of `true`.
4. Click Save and log out of the Console.
5. Restart the OpenSSO Enterprise web container.

You also need to set this property on the client side. For example, for a Distributed Authentication UI server deployment, set it to `true` in the `AMDistAuthConfig.properties` file.

Support is added for module-based, realm-based, and service-based authentication (CR 6893507)

In Patch 3, the OpenSSO REST-based authentication web service now supports module-based, realm-based, or service-based authentication. You can pass module, realm, and service as query parameters. For example, here are some sample REST commands:

```
http://host.example.com/opensso/identity/authenticate?username=user1
ANDAMPpassword=changeit
http://host.example.com/opensso/identity/authenticate?username=user1
ANDAMPpassword=changeitANDAMPuri=realm%3Dsun
http://host.example.com/opensso/identity/authenticate?username=user1
ANDAMPpassword=changeitANDAMPuri=module%3DDataStore
http://host.example.com/opensso/identity/authenticate?username=user1
ANDAMPpassword=changeitANDAMPuri=service%3DldapService
http://host.example.com/opensso/identity/authenticate?username=user1
ANDAMPpassword=changeitANDAMPuri=realm%3D/sun%26module%3DDataStore
http://host.example.com/opensso/identity/authenticate?username=user1
ANDAMPpassword=passwordANDAMPuri=realm%3D/iplanet%26module%3DdataStore
```

`AMLoginModule` class includes new method to determine user?s current session quota level (CR 6667760)

In Patch 3, the `AMLoginModule` class includes the new `isSessionQuotaReached()` method to determine a user?s current session quota level:

```
public boolean isSessionQuotaReached(String userName)
```

This new method checks if the sessionCount is greater than or equal to the sessionQuota and returns true or false, depending the result.

Thus, a custom authentication module can check a user?s current session quota level and then if the user is about to exceed the session quota, ask whether that user wants to continue the session. This feature is normally be more useful when session constraints are enabled.

OpenSSO provides new property to specify client configuration folder (CR 6903279)

If a new administrator user logs into OpenSSO Enterprise server and tries to access the OpenSSO client website (for example, as deployed from the opensso-client-jdk15.war file), the new administrator user is asked to perform the client reconfiguration even though the configuration has already been done by the previous administrator.

Patch 3 provides the new openssoclient.config.folder property as a JVM argument in the container's configuration file (server.xml or domain.xml) to specify the configuration folder. For example:

```
<jvm-options>-Dopenssoclient.config.folder=C:/Sun/opensso-client-config</jvm-options>
```

If this argument is not specified, the configuration folder is user.home by default.

OpenSSO Console checks for minimum password length of 8 characters (CR 6888785)

In Patch 3, the OpenSSO Console checks for a minimum password length of 8 characters for new users and for existing users who are changing a password.

OpenSSO Diagnostic Tool is available (CR 6900820)

Patch 3 includes the OpenSSO Diagnostic Tool, which allows you to run a number of diagnostic tests to verify configuration settings and to identify potential installation or deployment problems. For information, see the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.

Known Issues and Limitations in OpenSSO Enterprise 8.0 Update 1 Patch 3

- "OpenSSO ssoadm utility is not producing audit logs (CR 6928588)" on page 25
- "STS client samples deployed on WebLogic Server and Jetty are not working for the valid keystore (CR 6928433)" on page 25
- "Distributed Authentication UI deployments are not receiving session notifications (CR 6919698)" on page 25
- " updateschema.sh script does not modify idRepoService to include minimum password length validation (CR 6919321)" on page 25

- “Fedlet SSO HTTP POST link returns a blank page (CR 6927350)” on page 25

OpenSSO ssoadm utility is not producing audit logs (CR 6928588)

In Patch 3, the ssoadm utility does not produce audit logs to record which sub-commands have been executed. For example, the ssoadm list-realms sub-command should produce four audit log records (AMCLI-1, AMCLI-2, AMCLI-3020, and AMCLI-3021), but the log records are not produced.

STS client samples deployed on WebLogic Server and Jetty are not working for the valid keystore (CR 6928433)

In Patch 3, when the Security Token Server (STS) client samples are deployed on WebLogic Server and Jetty, the samples do not obtain the token that the server is deployed on WebLogic Server, and an uninitialized keystore error is thrown.

Distributed Authentication UI deployments are not receiving session notifications (CR 6919698)

After installing OpenSSO Enterprise 8.0 Patch 3, Distributed Authentication UI deployments are not receiving notifications from the server.

Workaround. The notification URL property com.iplanet.am.notification.url has been renamed to com.sun.identity.client.notification.url. Update the AMDistAuthConfig.properties configuration file for the Distributed Authentication UI server (and other clients) with the new com.sun.identity.client.notification.url property.

updateschema.sh script does not modify idRepoService to include minimum password length validation (CR 6919321)

Workaround.

After you apply Patch 3, the default minimum password length is 8 characters. However, to specify a different length for a different realm, run the following command:

```
./ssoadm set-realm-svc-attrs -u amadmin -f password-file
-s sunIdentityRepositoryService -e realm-name
-a sunIdRepoAttributeValidator=
class=com.sun.identity.idm.server.IdRepoAttributeValidatorImpl
sunIdRepoAttributeValidator=minimumPasswordLength=password-minimum-length
```

Fedlet SSO HTTP POST link returns a blank page (CR 6927350)

In Patch 3, the Fedlet SSO HTTP POST link randomly returns a blank page. This problem occurs when a user is logged in on the IDP side and a session is created with SSO. The problem also occurs with SAMLv2.

Workaround. None

Documentation Updates for OpenSSO Enterprise 8.0 Update 1 Patch 3

Upgrading to OpenSSO Enterprise 8.0 Update 1 Patch 3 (CR 6887525)

Always run the latest versions of the `ssopatch` or `ssopatch.bat` utility and the corresponding `updateschema.sh` or `updateschema.bat` script from the Patch 3 release.

If you are patching OpenSSO Enterprise 8.0 with Patch 3:

1. Run the `ssopatch` or `ssopatch.bat` utility from Patch 3.
2. Run the `updateschema` or `updateschema.bat` script from Patch 3.

For more information about patching OpenSSO Enterprise, see the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.

If you are moving to Patch 3 from Access Manager 7.1 or Access Manager 7 2005Q4:

1. Execute the `ssoupgrade` or `ssoupgrade.bat` script from Patch 3.
2. Run `updateschema` or `updateschema.bat` script from Patch 3.

For more information about upgrading, see the *Sun OpenSSO Enterprise 8.0 Upgrade Guide*.

Changing Information in the Directory Server `bootstrap` File (CR 6849622)

OpenSSO Enterprise 8.0 stores parameters used to access the directory server in the `/opensso/bootstrap` file. If required by your deployment, you can change some of these parameters using the OpenSSO Adminstration Console. For example, you can change the Directory Manager password.

To Change the Directory Server Parameters in the `bootstrap` File

1. Log in to the OpenSSO Administration Console.
2. Click Configuration, Servers and Sites, *opensso-instance-name*, and then Directory Configuration.
3. Change the following values, as required by your deployment:
   - Bind DN is the privileged directory server administrator.

     The default is `cn=Directory Manager`.
   - Bind Password is the password used by the Bind DN user to access the directory server.
4. You can also change the values for the following parameters, if you wish:
   - Minimum Connection Pool

- Maximum Connection Pool

5. When you have made your changes, click Save.

   The OpenSSO Console updates the responding values in the directory server `bootstrap` file.

OpenSSO Enterprise 8.0 Update 1 Patch 2 (141655-03)

- "Additional Web Container and Platform Support in OpenSSO Enterprise 8.0 Update 1 Patch 2" on page 27
- "Known Issues and Limitations in OpenSSO Enterprise 8.0 Update 1 Patch 2" on page 27

Additional Web Container and Platform Support in OpenSSO Enterprise 8.0 Update 1 Patch 2

Patch 141655-03 includes support for:

- IBM AIX 6.1 platform
- GlassFish Enterprise Server v2.1 web container

Known Issues and Limitations in OpenSSO Enterprise 8.0 Update 1 Patch 2

- "OpenSSO Enterprise cannot create URLStreamHandler for WebLogic Server (CR 6867442)" on page 27
- "Deploying the console.war file in patch 141655-03 generates a malformed `goto` URL (CR 6881715)" on page 28

OpenSSO Enterprise cannot create URLStreamHandler for WebLogic Server (CR 6867442)

The OpenSSO Enterprise `AMURLStreamHandlerFactory` cannot create the `URLStreamHandler` for WebLogic Server, because WebLogic Server has preset the value for the `java.protocol.handler.pkgs` system property to

`weblogic.net|weblogic.utils|weblogic.utils|weblogic.utils`. If you try to access a remote WebLogic Server instance from the Console Session UI, OpenSSO Enterprise dumps an error log in the CoreSystem file.

The fix for CR 6867442 adds the new `opensso.protocol.handler.pkgs` property.

Although this problem occurred on WebLogic Server, the fix affects all web containers. If you have `java.protocol.handler.pkg` in your setup or if you are planning to use `java.protocol.handler.pkg`, add this new property as follows:

1. In the OpenSSO Administration Console, click Configuration, Servers and Sites, *opensso-instance-name*, and then Advanced.

2. Click Add and then enter:
   - Property Name: `opensso.protocol.handler.pkgs`
   - Property Value: `com.sun.identity.protocol`
3. Click Save.

Deploying the console.war file in patch 141655-03 generates a malformed `goto` URL (CR 6881715)

If you deploy and configure the `console.war` file in patch 141655-03, when you access the login page, the `goto` URL page is malformed.

Workaround. Manually enter the `goto` URL as *protocol*`://`*openssohost*`:`*port*`/console` and re-request the login page. For example: `https://openssohost.example.com:8080/console`

Oracle periodically releases patches to OpenSSO Enterprise 8.0 Update 1 on `http://sunsolve.sun.com/`. To find the latest patch for Update 1, search for patch ID 141655. To determine if you should install a patch, check the README file available with the patch.

Each patch release includes an `opensso.war` file that you can deploy as follows:

- Patch an existing OpenSSO Enterprise 8.0 deployment
- Install a new OpenSSO Enterprise 8.0 deployment
- Create or patch one of the following specialized WAR files:
  - OpenSSO Enterprise Administration console only
  - OpenSSO Enterprise server only without the Administration console
  - OpenSSO Enterprise Distributed Authentication UI server
  - OpenSSO Enterprise IDP Discovery Service

For more information see Chapter 2, “Installing OpenSSO Enterprise 8.0 Update 1.”

Additional Information and Resources

- “Deprecation Notifications and Announcements” on page 29
- “How to Report Problems and Provide Feedback” on page 29
- “Accessibility Features for People With Disabilities” on page 30
- “Related Third-Party Web Sites” on page 30

You can also find additional useful information and resources at the following locations:

- Oracle Advanced Customer Services for Systems:

  `http://www.oracle.com/us/support/systems/advanced-customer-services/index.html`
- Software Products: `http://www.sun.com/software/`

- SunSolve: http://sunsolve.sun.com/
- Sun Developer Network (SDN): http://developers.sun.com/
- Sun Developer Services:http://developers.sun.com/services/

Deprecation Notifications and Announcements

- The Service Management Service (SMS) APIs (`com.sun.identity.sm` package) and SMS model will not be included in a future OpenSSO Enterprise release.
- The Unix authentication module and the Unix authentication helper (`amunixd`) will not be included in a future OpenSSO Enterprise release.
- The Sun Java System Access Manager 7.1 Release Notes stated that the Access Manager `com.iplanet.am.sdk` package, commonly known as the Access Manager SDK (AMSDK), and all related APIs and XML templates will not be included in a future OpenSSO Enterprise release.

  Consequently, when the AMSDK is removed, the Legacy Mode option and support will also be removed.

  Migration options are not available now and are not expected to be available in the future. Oracle Identity Manager provides user provisioning solutions that you can use instead of the AMSDK. For more information about Identity Manager, see http://www.oracle.com/products/middleware/identity-management/identity-manager.html.

How to Report Problems and Provide Feedback

If you have questions or issues with OpenSSO Enterprise 8.0 Update 1 or a subsequent patch release, contact Support Resources at http://sunsolve.sun.com/.

This site has links to the Knowledge Base, Online Support Center, and Product Tracker, as well as to maintenance programs and support contact numbers. If you are requesting help for a problem, please include the following information:

- Description of the problem, including when the problem occurs and its impact on your operation
- Machine type, operating system version, web container and version, JDK version, and OpenSSO Enterprise version, including any patches or other software that might be affecting the problem
- Steps to reproduce the problem
- Any error logs or core dumps

Accessibility Features for People With Disabilities

To obtain accessibility features that have been released since the publishing of this media, consult Section 508 product assessments available upon request to determine which versions are best suited for deploying accessible solutions.

For information about Oracle's commitment to accessibility, see `http://www.sun.com/accessibility/index.jsp`.

Related Third-Party Web Sites

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

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

Revision History

TABLE 1–1 Revision History

Part Number	Date	Description
821-1818-10	April 13, 2010	Initial release of converted document from the Wiki version.

CHAPTER 2

Installing OpenSSO Enterprise 8.0 Update 1

OpenSSO Enterprise 8.0 Update 1 Installation Overview

OpenSSO Enterprise 8.0 Update 1 is available as patch **141655-01** on `http://sunsolve.sun.com/`.

Before you install OpenSSO Enterprise 8.0 Update 1 (or subsequent patches), check the information about new features, hardware and software requirements, and issues and workarounds in this document.

OpenSSO Enterprise 8.0 Update 1 includes an `opensso.war` file that you can install using these methods:

- **Patch an existing OpenSSO Enterprise 8.0 deployment**: Use the `ssopatch` utility in Update 1 to patch an existing OpenSSO Enterprise 8.0 deployment, as described in this chapter.

  Note - Oracle supports patching only OpenSSO Enterprise 8.0 releases. For example, patching OpenSSO Enterprise 8.0 with OpenSSO Enterprise 8.0 Update 1 is supported.

- **Install a new OpenSSO Enterprise 8.0 Update 1 deployment:** Install and configure the OpenSSO Enterprise 8.0 Update 1 `opensso.war` file, as described in the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.
- **Create a new specialized WAR file**: Use the `createwar` script to create one of the following new WAR files from the Update 1 `opensso.war` file:
  - OpenSSO Administration console only WAR
  - Distributed Authentication UI server WAR
  - OpenSSO server only WAR, without the Administration Console
  - IDP Discovery Service WAR

  For information, see Chapter 4, "Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File."
- **Patch an existing specialized OpenSSO Enterprise WAR file**: Use the `ssopatch` utility in Update 1 to patch an existing specialized OpenSSO Enterprise 8.0 WAR file, as described inChapter 23, "Patching OpenSSO Enterprise 8.0," in *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*

Note – If you are running Access Manager 7.1 or Access Manager 7 2005Q4 and you want to upgrade to Update 1, follow these steps:

1. Upgrade Access Manager 7.x to OpenSSO Enterprise 8.0, as described in *Sun OpenSSO Enterprise 8.0 Upgrade Guide*.
2. Apply the Update 1 patch, as described in this chapter.

OpenSSO Enterprise 8.0 Update 1 Patches

Sun periodically releases patches for OpenSSO Enterprise 8.0 Update 1. For information about these patches, see "OpenSSO Enterprise 8.0 Update 1 Patch Releases" on page 21.

Planning Your Patch Operation

▼ To Plan Your Patch Operation for OpenSSO Enterprise 8.0

1. **Read the "Overview of the `ssopatch` Utility" on page 33.**
2. **Install the patch utility for your platform, as described in "Installing the `ssopatch` Utility" on page 34.**

3 **Get information about your existing WAR file, to determine if your existing WAR file has been customized or modified, as described in "Comparing an OpenSSO Enterprise WAR File to Its Internal Manifest" on page 36.**

4 **Compare your existing WAR file and the Update 1 WAR file, to return the files customized in the original WAR, files updated in the new WAR file, and files added or deleted between the two WAR versions, as described in "Comparing Two OpenSSO Enterprise WAR Files" on page 37.**

5 **Backup and archive your existing Opensso WAR file, as described in "Backing Up an OpenSSO Enterprise WAR File" on page 35.**

6 **Patch your OpenSSO Enterprise WAR File, as described in "Patching an OpenSSO Enterprise WAR File" on page 37.**

7 **Run the updateschema script, as described in "Running the updateschema Script" on page 41.**

Note - If you are patching a specialized WAR file that you generated from an opensso.war, such as an OpenSSO server only, administration console only, Distributed Authentication UI server, or IDP Discovery Service WAR, see "Patching a Specialized OpenSSO Enterprise WAR" on page 40.

Overview of the ssopatch Utility

The ssopatch utility is a Java command-line utility that is available on Solaris and Linux systems as ssopatch and on Windows as ssopatch.bat.

Note - The syntax for ssopatch in OpenSSO Enterprise 8.0 Update 1 has changed considerably since the OpenSSO Enterprise 8.0 release. For the new syntax, see "Running the updateschema Script" on page 41.

The ssopatch patch utility performs these functions:

- Compares an OpenSSO Enterprise WAR to its original manifest, to determine if the WAR file has been customized or modified
- Compare two OpenSSO Enterprise WAR files, to determine the differences between the two files including any customizations made to the original WAR file and any changes in the new WAR file
- Generates a staging area of the files required to generate a new patched OpenSSO Enterprise WAR file

After you download and unzip the OpenSSO Enterprise 8.0 Update 1 ZIP file (opensso_enterprise_80U1.zip), the patch utilities and related files are available in the ssoPatchTools.zip file, in the *zip-root*/opensso/tools directory, where *zip-root* is where you unzipped opensso_enterprise_80U1.zip.

The ssopatch utility uses a manifest file to determine the contents of a specific OpenSSO Enterprise WAR file. A manifest file is an ASCII text file that contains:

- A string that identifies the specific version of the OpenSSO Enterprise WAR file
- All of the individual files in the OpenSSO Enterprise WAR file, with checksum information for each file

The manifest file is usually named OpenSSO.manifest and is stored in the in the META-INF directory of the OpenSSO Enterprise WAR file.

The ssopatch utility sends its results to the standard output (stdout). If you prefer, you can capture the ssopatch output by redirecting the output to a file. If ssopatch finishes successfully, it returns a zero (0) exit code. If errors occur, ssopatch returns a non-zero exit code.

Installing the ssopatch Utility

Before you install the ssopatch utility:

- Download and unzip the OpenSSO Enterprise 8.0 Update 1 ZIP file (opensso_enterprise_80U1.zip).
- Set your JAVA_HOME environment variable point to JDK 1.5 or later.

To Install the ssopatch Utility

1. Locate the ssoPatchTools.zip file in the *zip-root*/opensso/tools directory, where *zip-root* is where you unzipped opensso_enterprise_80U1.zip.
2. Create a new directory to unzip the ssoPatchTools.zip file. For example: ssopatchtools
3. Unzip the ssoPatchTools.zip file in the new directory.
4. If you want to run the ssopatch utility from a directory other than its current directory without providing the full path, add the utility to your PATH variable.

The following table describes the files in ssoPatchTools.zip.

File or Directory	Description
README	Readme file that describes ssopatch
/lib	Required ssopatch JAR files
/patch	updateschema and updateschema.bat scripts and related XML files

File or Directory	Description
/resources	Required properties files
ssopatch and ssopatch.bat	Utilities for Solaris, Linux, and Windows systems

Backing Up an OpenSSO Enterprise WAR File

Before you begin, backup your existing OpenSSO Enterprise WAR file and configuration data:

- Copy your existing OpenSSO Enterprise WAR file to a safe location. Then, if you need to back out Update 1 for some reason, you can re-deploy your backup copy of the WAR file.
- Backup your configuration data, as described in Chapter 15, "Backing Up and Restoring Configuration Data," in *Sun OpenSSO Enterprise 8.0 Administration Guide*.

Running the ssopatch Utility

To run the ssopatch utility, follow this usage:

```
ssopatch
--help|-?
[--locale|-l]

ssopatch
--war-file|-o
[--manifest|-m]
[--locale|-l]

ssopatch
--war-file|-o
--war-file-compare|-c
[--staging|-s]
[--locale|-l]
[--override|-r]
[--overwrite|-w]
```

where the options are:

- -war-file|-o specifies a path to a WAR file (such as opensso.war) that has previously been deployed.
- -manifest|-m specifies the path to the manifest file you want to create. The manifest file will be generated from the WAR file indicated by -war-file|-o if this option is provided.
- -war-file-compare|-c species a path to a WAR file to compare against against the WAR file indicated by -war-file|-o.

- `-staging|-s` specifies a path to the staging area where the files from an OpenSSO Enterprise WAR will be written.
- `-locale|-l` specifies the locale to be used. If this option is not specified, `ssopatch` uses the default system locale.
- `-override|-r` overrides revision checking for the two WAR files. Revision checking determines the versions of the WAR files and continues only if the versions are compatible. This option allows you to override this check.

  Default is false (revision checking is performed).
- `-overwrite|-w` overwrites the files in the existing staging area. Default is false (files are not overwritten).

Comparing an OpenSSO Enterprise WAR File to Its Internal Manifest

Use this procedure to determine if an OpenSSO Enterprise WAR file has been customized or modified since it was downloaded.

The `ssopatch` utility generates a new internal manifest file and then compares this internal manifest against the manifest stored inside the original OpenSSO Enterprise WAR file in the `META-INF` directory.

To Compare an OpenSSO Enterprise WAR File to Its Internal Manifest

1. Run `ssopatch` to compare the OpenSSO Enterprise WAR file to its internal manifest. For example:

```
./ssopatch -o /zip-root/opensso/deployable-war/opensso.war
Generating Manifest for: /zip-root/opensso/deployable-war/opensso.war
Comparing manifest of Internal (Enterprise 8.0 Build 6(200810311055))
against /zip-root/opensso/deployable-war/opensso.war (generated-200905050855)
File not in original war (images/login-origimage.jpg)
File updated in new war (images/login-backimage.jpg)
File updated in new war (WEB-INF/classes/amConfigurator.properties)
Differences: 3
```

This example shows these changes to the original WAR file:

- `images/login-origimage.jpg` is in `opensso.war` but was not found in the original manifest.
- `images/login-backimage.jpg` has been customized in `opensso.war` from the original manifest.

- WEB-INF/classes/amConfigurator.properties file has been customized in opensso.war from the original manifest.

Comparing Two OpenSSO Enterprise WAR Files

Use this procedure to compare two WAR files, to show the files that have been:

- Customized in an original OpenSSO Enterprise WAR
- Updated in a new OpenSSO Enterprise WAR file
- Added or deleted between the two OpenSSO Enterprise WAR versions

To Compare Two OpenSSO Enterprise WAR Files

1. Run ssopatch to compare the two WAR files. In the example, the -override option is used to override the revision checking between the two WAR files:

```
./ssopatch -o /zip-root/opensso/deployable-war/opensso.war
-c /u1/opensso/deployable-war/opensso.war --override
Generating Manifest for: /zip-root/opensso/deployable-war/opensso.war
Original manifest: Enterprise 8.0 Build 6(200810311055)
New manifest: Enterprise 8.0 Update 1 Build 6.1(200904300525)
Versions are compatible
Generating Manifest for: /u1/opensso/deployable-war/opensso.war
Comparing manifest of /zip-root/opensso/deployable-war/opensso.war
(generated-200905050919) against
    /u1/opensso/deployable-war/opensso.war (generated-200905050920)
File updated in new war(WEB-INF/classes/amClientDetection_en.properties)
File updated in new war(WEB-INF/classes/fmSAMLConfiguration_fr.properties)
...
Differences: 1821
Customizations: 3
```

This example shows the files that have been updated and customized in the new WAR file.

Patching an OpenSSO Enterprise WAR File

Use this procedure to create a new staging area, where an original WAR file is merged with a new WAR file.

This operation compares the manifests for each WAR file and then shows:

- Files customized in the original WAR file
- Files updated in a new WAR file
- Files added or removed between the two WAR file versions

The `ssopatch` then copies the appropriate files to a staging directory, where you must add any customizations before you create and deploy the new patched WAR.

To Create a Staging Area to Patch an OpenSSO Enterprise WAR File

1. Although the `ssopatch` does not modify your original `opensso.war` file, it is recommended that you back up this file, in case you need to back out the patched `opensso.war` file.
2. Run `ssopatch` to create the staging area. For example:

```
./ssopatch -o /zip-root/opensso/deployable-war/opensso.war
  -c /u1/opensso/deployable-war/opensso.war --override -s /tmp/staging
Generating Manifest for: /zip-root/opensso/deployable-war/opensso.war
Original manifest: Enterprise 8.0 Build 6(200810311055)
New manifest: Enterprise 8.0 Update 1 Build 6.1(200904300525)
Versions are compatible
Generating Manifest for: /u1/opensso/deployable-war/opensso.war
Comparing manifest of /zip-root/opensso/deployable-war/opensso.war
    (generated-200905051031) against /u1/opensso/deployable-war/opensso.war
    (generated-200905051032)
File was customized in original, but not found in new war.
Staging area using original war version (samples/saml2/sae/header.jsp)
File was customized in original, but not found in new war.
Staging area using original war version
    (WEB-INF/template/opends/config/upgrade/config.ldif.4517)
File was customized in original, but not found in new war.
Staging area using original war version
    (WEB-INF/template/opends/config/upgrade/schema.ldif.4517)
Differences: 1813
Customizations: 0
```

In this example, `/tmp/staging` is the staging area where `ssopatch` copies the files.

Update the files as needed in the staging-area, using the results of the previous step.

Use the following table to determine the action you might need to take for each file before you generate a new patched WAR file.

`ssopatch` **Results**	**Explanation and Action Required**
`File not in original war` *filename*	The indicated file does not exist in the original WAR file but is in the latest version of the WAR file. **Action**: None

`ssopatch` **Results**	**Explanation and Action Required**
`File updated in new war` *filename*	The indicated file exists in both the original and new WAR files and has been updated in the latest version of the WAR file. No customizations have been done in the original WAR file. **Action**: None
`File customized` *filename*	The indicated file exists in both WAR files, has been customized in the original version of the WAR file, but has not been updated in the latest version of the WAR file. **Action**: None
`May require manual customization` *filename*	The file exists in both WAR files, has been customized in the original version of the WAR file, and has been updated in the latest version of the WAR file. **Action**: If you want your customizations in the file, you must manually add them to the new updated file in the staging directory.
`File was customized in original, but not found in new war`	The file existed in the original WAR file, but is not in the new WAR. **Action**: None.

Next Steps

1. Create a new OpenSSO Enterprise WAR file from the files in the staging area. For example:

   ```
   cd /tmp/staging
   jar cvf /patched/opensso.war *
   ```

   where `/patched/opensso.war` is the new patched OpenSSO Enterprise WAR file
2. Redeploy the `/patched/opensso.war` file to the web container using the original deploy URI. For example, `/opensso`

OpenSSO configuration changes. A new OpenSSO Enterprise WAR file might have configuration changes that were not in your original WAR file. Any configuration changes, if any, will be documented separately for each patch. Check the patch documentation and the *Sun OpenSSO Enterprise 8.0 Release Notes* for more information about any configuration changes. (The version string in the OpenSSO manifest file will change, even if there are no configuration changes in the new WAR file.)

If you need to back out your patched version, undeploy the patched WAR file and then redeploy your original WAR file.

Creating an OpenSSO Enterprise WAR Manifest File

An OpenSSO manifest file is a text file that identifies all of the individual files in a WAR file for a specific release, with checksum information for each file.

Use this procedure to create a manifest file that you can include in a specialized OpenSSO Enterprise WAR, such as an OpenSSO Enterprise server only, administration console only, Distributed Authentication UI server, or IDP Discovery Service WAR

To Create an OpenSSO Enterprise WAR Manifest File

1. Run `ssopatch` to create the OpenSSO manifest file. For example:

   ```
   ./ssopatch -o zip-root/opensso/deployable-war/opensso.war --manifest /tmp/manifest
   ```

   where `opensso.war` is an existing OpenSSO Enterprise WAR file.

   The `ssopatch` utility creates a new manifest file named `manifest` in the the `/tmp` directory.
2. To allow the WAR file to be patched, copy this new manifest file to the `META-INF` directory inside the `opensso.war` file. For example:

   ```
   mkdir META-INF
   cp /tmp/manifest META-INF
   jar uf opensso.war META-INF/manifest
   ```

Patching a Specialized OpenSSO Enterprise WAR

If you have previously created a specialized OpenSSO Enterprise WAR, such as an OpenSSO Enterprise server only, administration console only, Distributed Authentication UI server, or IDP Discovery Service WAR, you can patch it by using the `ssopatch` utility.

To Patch a Specialized OpenSSO Enterprise WAR

1. Create a manifest file for your specialized OpenSSO Enterprise WAR, as described in "Creating an OpenSSO Enterprise WAR Manifest File" on page 40.

   Note: Create the manifest file based on the original OpenSSO Enterprise 8.0 `opensso.war`, as delivered from Sun, prior to any customizations you might have done. If the manifest is created after customizations, `ssopatch` might use the files from Update 1, rather than your customizations, so you would need to re-do your customizations after patching.
2. Generate the specialized OpenSSO Enterprise WAR from the OpenSSO Enterprise 8.0 Update 1 `opensso.war` file, as described in Chapter 4, "Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File."

3. Use the ssopatch utility to compare the your old and new WAR files.
4. Generate a staging area for the new specialized WAR file, as described in "To Create a Staging Area to Patch an OpenSSO Enterprise WAR File" on page 38.
5. Redeploy the new specialized WAR file.

Running the updateschema Script

After you run ssopatch, run the updateschema.sh on Solaris or Linux systems or updateschema.bat on Windows. The script updates the OpenSSO Enterprise server version, adds new default server properties, adds new attribute schemas required for bug fixes and enhancements in Update 1. You must run updateschema in order to update the server version.

Before You Begin

The updateschema.sh or updateschema.bat script requires the Update 1 version (or later) of the ssoadm command-line utility. Therefore, before you run this script, install the Update 1 admin tools, as described in Chapter 3, "Installing the OpenSSO Enterprise 8.0 Update 1 Admin Tools."

To Run the updateschema Script

1. Change to the *patch-tools*/patch directory, where *patch-tools* is where you unzipped ssoPatchTools.zip.
2. Run updateschema.sh or updateschema.bat. For example, on Solaris systems:

   ```
   ./updateschema.sh
   ```
3. When the scripts prompts you, provide the following information:

- Full path to the ssoadm utility (excluding ssoadm itself). For example: /opt/ssotools/opensso/bin
- amadmin password

The updateschema.sh or updateschema.bat script writes any messages or errors to the standard output.

Backing Out a Patch Installation

If you need to back out your patch installation, simply redeploy the original `opensso.war` file (or specialized WAR file).

CHAPTER 3

Installing the OpenSSO Enterprise 8.0 Update 1 Admin Tools

The **Sun OpenSSO Enterprise 8.0 Update 1** ZIP file (opensso_enterprise_80U1.zip) includes the admin tools, scripts, libraries, and other supporting files in the ssoAdminTools.zip file, including:

- ssoadm
- amverifyarchive
- ampassword
- amtune tuning scripts

Note – The OpenSSO Enterprise 8.0 ssoadm utility is not compatible with OpenSSO Enterprise 8.0 Update 1 ssopatch utility. Therefore, after you deploy OpenSSO Enterprise 8.0 Update 1, also re-install the Update 1 tools and scripts in the ssoAdminTools.zip file.

- "ssoAdminTools.zip Files" on page 43
- "Using ssoadm With OpenSSO Enterprise Configured as a Site" on page 45

ssoAdminTools.zip Files

The ssoAdminTools.zip file is in the following directory:

zip-root/*deploy_uri*/tools where *zip-root* is where you unzipped the opensso_enterprise_80U1.zip file and *deploy_uri* is the deployment URI.

For example: /opt/opensso/tools

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

File or Directory	Description
README.setup	Description of the ssoAdminTools.zip file.

File or Directory	Description
license.txt	CDDL license agreement.
setup	Script to install the tools on Solaris and Linux systems.
setup.bat and setup.bat	Scripts to install the tools on Solaris, Linux, and Windows systems.
lib	JAR files required to run the scripts.
resources	Properties files required for the scripts for the various locales.
template	Script templates for Solaris, Linux, and Windows systems.

▼ To Install the OpenSSO Enterprise Tools and Scripts

1. **Make sure the the OpenSSO Enterprise 8.0 Update 1 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 `ssoAdminTools.zip` file. For example: `tools-zip-root`.**

4. **Unzip the `ssoAdminTools.zip` file in the new directory.**

5. **In the directory where you unzipped the `ssoAdminTools.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 Enterprise configuration, log, and debug directories. For example: `/opensso`**

Next Steps You can now run the OpenSSO Enterprise CLI tools and utilities from the following directory:

tools-zip-root/*deploy_uri*/bin

where:

- *tools-zip-root* is the directory where you unzipped the ssoAdminTools.zip file.
- *deploy_uri* is the name of the OpenSSO Enterprise deploy URI. For example: opensso

  For example: /opt/ssotools/opensso/bin

Related Information

- CLI utilities: *Sun OpenSSO Enterprise 8.0 Administration Reference*
- Tuning scripts: *Sun OpenSSO Enterprise 8.0 Performance Tuning Guide*

Using `ssoadm` With OpenSSO Enterprise Configured as a Site

In a typical large deployment, OpenSSO Enterprise server instances are configured behind one or load balancers. The HTTP(s) traffic is usually one directional. That is, the traffic goes from one of the load balancers to the servers, but requests from servers are unable to reach the load balancers.

If the above scenario applies to your deployment and you need to use the `ssoadm` utility (Solaris and Linux systems) or `ssoadm.bat` utility (Windows), perform the following procedure.

▼ To Use `ssoadm` With OpenSSO Enterprise Configured as a Site

1 **After you install the tools, edit the `ssoadm` or `ssoadm.bat` utility by adding the following property to the `java` command:**

```
-D"com.iplanet.am.naming.map.site.to.server=
http://lb.example.com:8080/opensso=http://sso1.example.com:8080/opensso"
```

where `lb` is the load balancer, and `sso1` is the OpenSSO Enterprise server where `ssoadm` or `ssoadm.bat` is installed.

2 **Save the `ssoadm` or `ssoadm.bat` utility.**

The utility can now send naming requests to the OpenSSO Enterprise server instance.

CHAPTER 4

Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File

Overview of the `createwar` Script

Sun OpenSSO Enterprise 8.0 Update 1 includes the `createwar` script to simplify the creation of the following specialized OpenSSO WAR files:

- OpenSSO Administration Console only WAR
- Distributed Authentication UI server WAR
- OpenSSO server only WAR, without the Administration Console
- IDP Discovery Service WAR

This script creates specialized OpenSSO WAR files by automatically executing several steps that you previously had to perform using the `jar` command.

The `createwar` script is available for the following platforms:

- Solaris and Linux systems: `createwar.sh`
- Windows: `createwar.bat`

After you unzip `opensso_enterprise_80U1.zip` for OpenSSO Enterprise 8.0 Update 1, the script is in the following directory:

zip-root/*deploy_uri*/`deployable-war`

where:

- *zip-root* is the directory where you unzipped opensso_enterprise_80U1.zip for OpenSSO Enterprise 8.0 Update 1.
- *deploy_uri* is the OpenSSO deploy URI. The default is opensso.

For example: /downloads/opensso/deployable-war

The following table shows the contents of the *zip-root*/opensso/deployable-war directory, where *zip-root* is where you unzip the opensso_enterprise_80U1.zip file.

File or Directory	Description
README	README file
createwar.sh	Shell script to create WAR files on Solaris and Linux systems.
createwar.bat	Script to create WAR files on Windows.
opensso.war	OpenSSO Enterprise 8.0 Update 1 server WAR with samples.
fam-idpdiscovery.list	Files list for an IDP Discovery Service WAR.
fam-distauth.list	Files list for a Distributed Authentication UI server WAR.
fam-console.list	Files list for a Console Only WAR.
fam-noconsole.list	Files list for an OpenSSO Enterprise 8.0 Update 1 server WAR without the console.
classes	Directory containing resources and classes for executing the createwar.sh and createwar.bat scripts.
idpdiscovery	Directory containing additional files for an IDP Discovery Service WAR.
distauth	Directory containing additional files for Distributed Authentication UI server WAR.
console	Directory containing additional files for a console only WAR.
noconsole	Directory containing additional files for an OpenSSO Enterprise 8.0 Update 1 server WAR without the console.

Running the createwar Script

To run the createwar script, follow this usage:

```
createwar --staging|-s stagingDir
--type |-t typeOfWarFile
--warfile|-w warfileName
[--locale|-l locale]
```

where:

- createwar is either createwar.sh or createwar.bat, depending on your platform.
- stagingDir is the staging directory where the opensso.war files are extracted.
- typeOfWarFile is the type of WAR file you want to create:
  - console - OpenSSO Administration Console only WAR
  - distauth - Distributed authentication UI server WAR
  - noconsole - OpenSSO server only WAR (without the Administration Console)
  - ldpdiscovery - IDP Discovery Service
- warfileName is the name of the specialized WAR file to be created.
- locale (optional) indicates the locale of your system where you are running the createwar script. Choices can be de (German), en_US (US English), es (Spanish), fr (French), ja (Japanese), zh_CN (Simplified Chinese), or zh_TW (Traditional Chinese).

To display the createwar help:

```
createwar -help|? [--locale|-l locale]
```

Before You Begin Creating a Specialized WAR

- If necessary, download and unzip the opensso_enterprise_80U1.zip file for OpenSSO Enterprise 8.0 Update 1.
- Set your JAVA_HOME environment variable to the location of your JDK installation. The createwar script requires JDK 1.5 or later.
- On Solaris and Linux systems, make the createwar.sh script executable. For example: chmod +x createwar.sh

Examples of Creating Specialized OpenSSO War Files

The following examples are intended for Solaris and Linux systems. If you are on a Windows system, run the createwar.bat script and adjust the paths for Windows conventions.

- “Creating a Console Only WAR File” on page 49
- “Creating a Distributed Authentication UI Server WAR File” on page 50
- “Creating a Server Only (No Admin Console) WAR File” on page 50
- “Creating an IDP Discovery Service WAR File” on page 51

Creating a Console Only WAR File

To Create a Console Only WAR File

1. Create a staging directory and extract the files from opensso.war. For example:

```
mkdir /tmp/consolewarstaging
cd /tmp/consolewarstaging
jar xvf /downloads/opensso/deployable-war/opensso.war
```

2. Run the `createwar.sh` script to create a console only WAR file named `console.war`.

You must execute `createwar.sh` (or `createwar.bat` on Windows) script in the `deployable-war` directory.

```
cd /downloads/opensso/deployable-war
./createwar.sh -s /tmp/consolewarstaging -t console -w /tmp/console.war
```

3. You are now ready to deploy your new WAR, as described in "After You Finish Creating a Specialized WAR" on page 51.

Creating a Distributed Authentication UI Server WAR File

To Create a Distributed Authentication UI Server WAR File

1. Create a staging directory and extract the files from `opensso.war`. For example:

```
mkdir /tmp/dawarstaging
cd /tmp/dawarstaging
jar xvf /downloads/opensso/deployable-war/opensso.war
```

2. Run the `createwar.sh` script to create a Distributed Authentication UI server WAR file named `distauth.war`.

You must execute `createwar.sh` (or `createwar.bat` on Windows) script in the `deployable-war` directory.

```
cd /downloads/opensso/deployable-war
./createwar.sh -s /tmp/dawarstaging -t distauth -w /tmp/distauth.war
```

3. You are now ready to deploy your new WAR, as described in "After You Finish Creating a Specialized WAR" on page 51.

Creating a Server Only (No Admin Console) WAR File

To Create a Server Only (No Admin Console) WAR File

1. Create a staging directory and extract the files from `opensso.war`. For example:

```
mkdir /tmp/serveronlywarstaging
cd /tmp/serveronlywarstaging
jar xvf /downloads/opensso/deployable-war/opensso.war
```

2. Run the createwar.sh script to create a server only WAR file named serveronly.war.

You must execute createwar.sh (or createwar.bat on Windows) script in the deployable-war directory.

```
cd /downloads/opensso/deployable-war
./createwar.sh -s /tmp/serveronlywarstaging -t noconsole -w /tmp/serveronly.war
```

3. You are now ready to deploy your new WAR, as described in "After You Finish Creating a Specialized WAR" on page 51.

Creating an IDP Discovery Service WAR File

To Create an IDP Discovery Service WAR File

1. Create a staging directory and extract the files from opensso.war. For example:

```
mkdir /tmp/idpdiscoverywarstaging
cd /tmp/idpdiscoverywarstaging
jar xvf /downloads/opensso/deployable-war/opensso.war
```

2. Run the createwar.sh script to create a IDP Discovery Service WAR file named idpdiscovery.war.

You must execute createwar.sh (or createwar.bat on Windows) script in the deployable-war directory.

```
cd /downloads/opensso/deployable-war
./createwar.sh -s /tmp/idpdiscoverywarstaging -t idpdiscovery -w /tmp/idpdiscovery.war
```

3. You are now ready to deploy your new WAR, as described in "After You Finish Creating a Specialized WAR" on page 51.

After You Finish Creating a Specialized WAR

1. If you are deploying a Distributed Authentication UI server WAR file, see "Creating a Distributed Authentication UI Server WAR File" on page 50.
2. Deploy the specialized WAR file into your web container.
3. Access the specialized WAR deployment URL from your browser. For example, for a console only WAR: http://console-host.example.com:8080/console

4. When the configurator page is displayed, enter the configuration information for the specialized WAR.

The OpenSSO Configurator creates configuration data as follows:

- OpenSSO Administration Console only: `AMConfig.properties` file in the home directory of the user running the web container where the specialized WAR file is deployed
- Distributed Authentication UI server: `/FAMDistAuth/AMDistAuthConfig.properties` file in the home directory of the user running the web container where the specialized WAR file is deployed
- OpenSSO server only:
  - Default configuration directory, which is `opensso` in the home directory of the user running the Configurator. For example, if the Configurator is run by super user (`root`), the configuration directory is `/opensso`.
  - *user-home-directory*`.openssocfg`, where *user-home-directory* is the home directory of the user who deployed the WAR file. For example, if the user is super user (`root`), the directory is `/.openssocfg`.
- IDP Discovery Service: `libIDPDiscoveryConfig.properties` file in the home directory of the user running the web container where the specialized WAR file is deployed

Note – It is highly recommended that you change the permissions of a configuration file to limit access to sensitive configuration information, such as the administrator password.

Next Steps

Access the specialized OpenSSO WAR from your browser for its specific function, using the same URL from Step 3.

For example: `http://console-host.example.com:8080/console`

Creating a Distributed Authentication UI Server User

Before you configure a Distributed Authentication UI server WAR file, on the OpenSSO Enterprise sserver, create a user who has the “Read and write access to all realm and policy properties” privilege. You will specify this user (and password) when you run the Configurator (`distAuthConfigurator.jsp`) to configure the WAR file.

To Create a Distributed Authentication UI Server User

1. Log in to the OpenSSO server administration console as `amadmin`.
2. Click Access Control, the default realm, and then Subjects.
3. Create a new user. For example: `dauser`

4. Create a new group. For example: `dagroup`
5. Add the new user to the new group.
6. Click Privileges and then the link for the new group.
7. Select the "Read and write access to all realm and policy properties" privilege and click Save.

Related Information

For information about deploying and configuring your new specialized WAR file, see the *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.

CHAPTER 5

Deploying IBM WebSphere Application Server 7.0 as the OpenSSO Enterprise 8.0 Update 1 Web Container

WebSphere Application Server 7.0 is supported on Solaris, Linux, Windows, and IBM AIX 5.3 systems.

Before Deploying OpenSSO on WebSphere Application Server 7.0

Complete the following steps:

1. Add genericJvmArguments and Security Permissions
2. Run the JSP compiler

Before making changes to any file described in this chapter, it a good practice to stop the web container and make a backup of the file.

Add GenericJvmArguments and Security Permissions

Add the genericJvmArguments using the WebSphere Admin Console or by editing the server.xml file:

1. Open the following file:

   ```
   install_root/IBM/WebSphere/AppServer/profiles/AppSrv01/config/cells/<cellName>/nod
   ```

2. Find the jvmEntries element.
3. Add the following JVM options to genericJVMArguments in server.xml and save the file:

   ```
    genericJvmArguments="-Djava.awt.headless=true -DamCryptoDescriptor.provider=IBMJCE
     -DamKeyGenDescriptor.provider=IBMJCE -Djavax.management.builder.initial=  /
   -Dcom.sun.management.jmxremote"
   ```

4. If the Java Security Manager is enabled, add the following security permissions to the server.policy file, and then save the file:

```
grant {
permission java.net.SocketPermission "*", "listen,connect,accept,resolve";
permission java.util.PropertyPermission "*", "read, write";
permission java.lang.RuntimePermission "modifyThreadGroup";
permission java.lang.RuntimePermission "setFactory";
permission java.lang.RuntimePermission "accessClassInPackage.*";
permission java.util.logging.LoggingPermission "control";
permission java.lang.RuntimePermission "shutdownHooks";
permission javax.security.auth.AuthPermission "getLoginConfiguration";
permission javax.security.auth.AuthPermission "setLoginConfiguration";
permission javax.security.auth.AuthPermission "modifyPrincipals";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission java.io.FilePermission "<<ALL FILES>>", "read,write,execute,delete";
permission java.util.PropertyPermission "java.util.logging.config.class", "write";
permission java.security.SecurityPermission "removeProvider.SUN";
permission java.security.SecurityPermission "insertProvider.SUN";
permission javax.security.auth.AuthPermission "doAs";
permission java.util.PropertyPermission "java.security.krb5.realm", "write";
permission java.util.PropertyPermission "java.security.krb5.kdc", "write";
permission java.util.PropertyPermission "java.security.auth.login.config", "write";
permission java.util.PropertyPermission "user.language", "write";
permission javax.security.auth.kerberos.ServicePermission "*", "accept";
permission javax.net.ssl.SSLPermission "setHostnameVerifier";
permission java.security.SecurityPermission "putProviderProperty.IAIK";
permission java.security.SecurityPermission "removeProvider.IAIK";
permission java.security.SecurityPermission "insertProvider.IAIK";
permission java.lang.RuntimePermission "setDefaultUncaughtExceptionHandler";
permission javax.management.MBeanServerPermission "newMBeanServer";
permission javax.management.MBeanPermission "*", "registerMBean";
permission java.lang.RuntimePermission "createClassLoader";
permission javax.security.auth.AuthPermission "getSubject";
permission javax.management.MBeanTrustPermission "register";
permission java.lang.management.ManagementPermission "monitor";
permission javax.management.MBeanPermission "*", "queryMBeans";
permission javax.management.MBeanServerPermission "createMBeanServer";
permission java.security.SecurityPermission "getProperty.authconfigprovider.factory";
permission java.security.SecurityPermission "setProperty.authconfigprovider.factory";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.lang.RuntimePermission "setContextClassLoader";
permission java.lang.RuntimePermission "setFactory";
permission java.lang.RuntimePermission "setIO";
permission java.lang.RuntimePermission "modifyThread";
permission java.lang.RuntimePermission "stopThread";
permission java.lang.RuntimePermission "getProtectionDomain";
permission java.lang.RuntimePermission "readFileDescriptor";
permission java.lang.RuntimePermission "writeFileDescriptor";
permission java.lang.RuntimePermission "loadLibrary.*";
permission java.lang.RuntimePermission "accessClassInPackage.*";
permission java.lang.RuntimePermission "defineClassInPackage.*";
permission java.lang.RuntimePermission "accessDeclaredMembers";
permission java.lang.RuntimePermission "queuePrintJob";
permission java.io.FilePermission "<<ALL FILES>>", "read,write,execute,delete";
permission java.util.PropertyPermission "*", "read,write";
permission com.ibm.oti.shared.SharedClassPermission "*", "read,write";
permission com.ibm.websphere.security.WebSphereRuntimePermission "getSSLConfig",  /
"read,write,execute,delete";
permission java.lang.reflect.ReflectPermission "suppressAccessChecks";
permission javax.management.MBeanPermission "*", "isInstanceOf";
permission javax.management.MBeanPermission "*", "getAttribute";
permission java.net.NetPermission "getProxySelector";
};
```

5. Restart WebSphere Application Server 7.0.

Using the ssoadm and ampassword Utilities with the IBM JDK

After deploying OpenSSO on WebSphere Application Server 7.0, you can use the setup script in ssoAdminTools.zip to install the utilities and scripts. For information, see Chapter 3, "Installing the OpenSSO Enterprise 8.0 Update 1 Admin Tools."

1. Before you run the setup script to install the utilities and scripts, modify the setup script. Before -cp... in the last line, insert:

   ```
   -D"amCryptoDescriptor.provider=IBMJCE"
   -D"amKeyGenDescriptor.provider=IBMJCE"
   ```

2. Before you run ssoadm, add the following items to the ssoadm script:
   a. Add xalan.jar to the classpath after openfedlib.jar. For example:

      ```
      $<TOOLS_HOME>/lib/xalan.jar
      ```

   b. Add the following items before com.sun.identity.cli.CommandManager and com.sun.identity.tools.bundles.Main

      ```
      -D"amKeyGenDescriptor.provider=IBMJCE"
      -D"amCryptoDescriptor.provider=IBMJCE"
      ```

3. Before you run ampassword, add the following items to the ampassword script before com.iplanet.services.ldap.ServerConfigMgr and com.sun.identity.tools.bundles.Main

   ```
   -D"amCryptoDescriptor.provider=IBMJCE"
   -D"amKeyGenDescriptor.provider=IBMJCE"
   ```

4. If the OpenSSO server is SSL-enabled, then you must add the IBM JAR files and set -D options in the ssoadm script.
   a. Add the following IBM JAR files:

```
WAS_HOME/deploytool/itp/plugins/com.ibm.ast.ws.v7.jaxrpc.jee5_1.0.0.v200808141532/lib/emfwor /
kbench.jar
<WAS_HOME>/deploytool/itp/plugins/com.ibm.websphere.v7_7.0.0.v20080817/wasJars/bootstrap.jar
<WAS_HOME>/deploytool/itp/plugins/com.ibm.websphere.v7_7.0.0.v20080817/wasJars/wsexception.jar
<WAS_HOME>/dev/was_public.jar
<WAS_HOME>/deploytool/itp/plugins/com.ibm.websphere.v7_7.0.0.v20080817/wasJars/ras.jar
<WAS_HOME>/runtimes/com.ibm.jaxws.thinclient_7.0.0.jar
```

b. Set the following -D options :

```
-D"java.protocol.handler.pkgs=com.ibm.net.ssl.www.protocol"
-D"javax.net.ssl.trustStoreType=<storeType>"
-D"javax.net.ssl.trustStore=<trustStore_with_path>"
-D"javax.net.ssl.trustStorePassword=<password>"
```

CHAPTER 6

Centralizing SAML Error Display in OpenSSO Enterprise 8.0 Update 1

A centralized error processing URL is now supported to display all error conditions caught during SAML versions 1.x and 2 protocol processing. (This URL does not handle external application errors, but only those thrown by OpenSSO when using the SAMLv1.x and SAMLv2 protocols.) By default, the error processing URL points to `saml2error.jsp`, a JavaServer Page (JSP) that ships with OpenSSO. `saml2error.jsp` can be found in the `/saml2/jsp` directory inside the exploded `opensso.war`.

How Does it Work?

The error processing URL provides the path to which a user agent is redirected or forwarded when a SAML processing error occurs. The Error Processing URL attribute is configured using the OpenSSO console. Out-of-the-box, `saml2error.jsp` is hosted within the OpenSSO WAR. It (or any customized page) can also be hosted with the external customer application.

- If the page is hosted within `opensso.war`, a forward is used to send the user agent to the URL. In this case, the value of the Error Processing URL attribute is `/saml2/jsp/saml2error.jsp`. (This is the default configuration.)
- If the page is hosted outside of `opensso.war`, an HTTP-REDIRECT or HTTP-POST (depending on the configuration) is used to send the user agent to the URL. In this case, the value of the Error Processing URL attribute is a URL like the following and must be modified as documented in “Configuring the Error Processing URL Attribute” on page 60.

  ```
  http://www.your-app.com/app/saml2error.jsp
  ```

Which Parameters are Sent?

Three query parameters that define the error condition are sent to the error processing URL.

- `errorcode` is the i18n key of the error message. See "SAML Error Messages" on page 60 for a list.
- `httpstatuscode` is the HTTP status code of the error.
- `message` contains the details of the i18n error message.

Configuring the Error Processing URL Attribute

▼ To Configure the Error Processing URL Attribute

1. **Login to the OpenSSO console as administrator; by default,** `amadmin`**.**

2. **Click the Configuration tab.**

3. **Click the Global tab.**

4. **Click the Common Federation Configuration link.**

5. **Enter the appropriate URL as the value for the SAML Error Page URL attribute.**

6. **Enter the appropriate binding as the value for the SAML Error Page HTTP Binding attribute.**

   The default binding is HTTP-POST. You may change this to HTTP-REDIRECT.

7. **Click Save.**

8. **Log out of the console.**

SAML Error Messages

- "SAMLv2 Error Codes" on page 60
- "SAMLv1.x Error Codes" on page 63

SAMLv2 Error Codes

- `nullSPEntityID` : Service provider entity identifier is blank.
- `nullIDPEntityID` : Identity provider entity identifier is blank.

- `idpNotFound` : Identity provider (using the SourceID in the artifact) is not found.
- `requestProcessingError` : Error processing `AuthnRequest`.
- `failedToProcessSSOResponse` : Failed to process the single sign-on response.
- `nullInput` : Blank input.
- `requestProcessingMNIError` : Error processing `ManageNameIDRequest`.
- `nullRequestType` : Request Type is not specified.
- `nullSSOToken` : No `SSOToken` is found.
- `LogoutRequestProcessingError` : Error processing `LogoutRequest`.
- `LogoutResponseProcessingError` : Error processing `LogoutResponse`.
- `largeContentLength` : Length of the content in the SOAP request is too long.
- `errorMetaManager` : Error getting an instance of the metadata manager.
- `metaDataError` : Error retrieving metadata.
- `nullSessionProvider` : Session Provider is not specified.
- `SSOFailed` : Single sign on failed.
- `LogoutRequestCreationError` : Error creating `LogoutRequest`.
- `nullAssertionID` : No `AssertionID` specified.
- `failedToGetAssertionIDRequestMapper` : Error retrieving the AssertionID request mapper.
- `failedToAuthenticateRequesterURI` : Failed to authenticate the requester using the URI binding.
- `invalidAssertionID` : Invalid `AssertionID` value.
- `invalidAssertion` : Invalid `Assertion`.
- `unsupportedEncoding` : Character encoding used is not supported.
- `MissingSAMLRequest` : `SAMLRequest` ID is missing from the `HttpRequest`.
- `nullDecodedStrFromSamlResponse` : Decoded string from `LogoutResponse` is null.
- `nullIDPMetaAlias` : Identity provider `metaAlias` is null.
- `metaDataError` : Error retrieving the metadata.
- `invalidSOAPMessage` : The `SOAPMessage` sent by the client is not valid.
- `unableToCreateArtifactResponse` : Unable to create a SAMLv2 `ArtifactResponse`.
- `LogoutRequestCreationError` : Error creating a `LogoutRequest`.
- `UnableToRedirectToAuth` : Unable to redirect to the Authentication Service URL.
- `errorCreateArtifact` : Error creating the Artifact.
- `failedToSendECPResponse` : Failed to send ECP response.
- `notSupportedHTTPMethod` : The specified single sign-on profile is not supported.

- missingArtifact : The SAMLArt is missing from the HttpRequest.
- errorObtainArtifact : Could not obtain the Artifact from the HttpRequest.
- failedToGetIDPSSODescriptor : Failed to get SSODescriptor element from the identity provider metadata.
- errorCreateArtifactResolve : Could not create an ArtifactResolve.
- errorInSOAPCommunication : Could not obtain the ArtifactResponse due to an error in SOAP communication.
- cannotFindIDP : Could not find the identity provider based on the Artifact string.
- cannotFindArtifactResolutionUrl : Could not find the identity provider's Artifact Resolution URL.
- soapError : Error occurred in SOAP communication.
- failedToCreateArtifactResponse : Failed to create the ArtifactResponse object.
- missingArtifactResponse : ArtifactResponse is missing from SOAPMessage.
- invalidSignature : Invalid signature in the ArtifactResponse.
- invalidInResponseTo : Invalid InResponseTo attribute in the ArtifactResponse.
- invalidIssuer : Invalid Issuer attribute in the ArtifactResponse.
- invalidStatusCode : Invalid StatusCode attribute in the ArtifactResponse.
- failedToCreateSOAPMessage : SOAPMessage was not created.
- failedToCreateResponse : Response was not created.
- assertionNotSigned : SAML Assertion is not signed.
- missingSAMLResponse : SAMLResponse is missing from the HttpRequest.
- errorObtainResponse : Couldn't obtain SAMLResponse from the HttpRequest.
- errorDecodeResponse : Error decoding the SAMLResponse in the HttpRequest.
- invalidHttpRequestFromECP : Invalid HttpRequest from the ECP.
- failedToProcessQueryRequest : Failed to process the query request.
- failedToCreateAssertionIDRequest : Could not create the AssertionIDRequest.
- nullPathInfo : No URI path information found in the request.
- invalidMetaAlias : Entity's metaAlias is invalid.
- failedToCreateAttributeQuery : Unable to create the AtributeQuery object.
- failedToCreateAuthnQuery : Unable to create the AuthnQuery object.
- nameIDMappingFailed : Name identifier mapping failed.
- failedToInitECPRequest : Failed to initiate the ECP request.
- singleLogoutFailed : Single logout failed.
- nullRequestUri : The request URI is not specified.

- `invalidRequestUri` : Unable to determine federation protocol based on the request URI.
- `noRedirectionURL` : No redirection URL is specified.
- `readerServiceFailed` : Reader service failed.

SAMLv1.x Error Codes

- `untrustedSite` : Site corresponding to the `SiteID` is not trusted.
- `nullInputParameter` : Input parameter is blank.
- `invalidConfig` : Invalid configuration
- `missingTargetHost` : Target host information is missing.
- `nullTrustedSite` : Trusted site is blank.
- `errorCreateArtifact` : Could not create the `Artifact`.
- `targetForbidden` : Access to target host is forbidden.
- `failedCreateSSOToken` : Did not create an `SSOToken`.
- `missingTargetSite` : Target site is missing.
- `couldNotCreateResponse` : Could not create the `SAMLResponse`.
- `errorSigningResponse` : Could not sign the `SAMLResponse`.
- `errorEncodeResponse` : Could not encode the `SAMLResponse`.
- `missingSAMLResponse` : The `SAMLResponse` is not there.
- `errorDecodeResponse` : Could not decode the `SAMLResponse`.
- `errorObtainResponse` : Could not get the `SAMLResponse`.
- `invalidResponse` : The `SAMLResponse` is invalid.

?

CHAPTER 7

Encrypting Data in a Secure Attribute Exchange in OpenSSO Enterprise 8.0 Update 1

Secure Attribute Exchange (also referred to as Virtual Federation) allows one application to communicate identity data to a second application in a different domain. This chapter contains information on how to encrypt the data using the `com.sun.identity.sae.api` package, including:

How Secure Attribute Exchange Data Encryption Works

When data encryption is used, attributes pushed from the identity provider application to its local instance of OpenSSO are encrypted as well as attributes sent from the local instance of OpenSSO on the service provider side to the service provider application. Both symmetric and asymmetric data encryption is supported in a Secure Attribute Exchange.

For symmetric encryption, the encryption key is the same shared secret used to sign the data. For asymmetric encryption, the sender will encrypt the encryption key using the receiving party's public key. Upon receipt of the encrypted data, the receiving party will decrypt the encryption key using its private key, and then decrypt the data using the encryption key.

Planning the Encryption Specifics

Before getting into the application coding and configuration of OpenSSO for Secure Attribute Exchange encryption, the administrator must decide:

- The encryption algorithm (for example, AES or DES)
- The key strength (key size)
- The security mechanism (symmetric or asymmetric)

  If using asymmetric encryption, make sure the identity provider's public key is accessible (for example, in a keystore or through a URL) to the application on the identity provider side. Additionally, add the public key of the application on the service provider side to the service provider's keystore.

To Use the `com.sun.identity.sae.api`

1. Initialize an instance of `com.sun.identity.sae.api.SecureAttrs` using the:

   `SecureAttrs.init(String` *instance-name*`, String` *type*`, Properties` *properties*`)` method.

   Be sure to add values for `SAE_CONFIG_DATA_ENCRYPTION_ALG` and `SAE_CONFIG_ENCRYPTION_KEY_STRENGTH` to `Properties`.
2. To encrypt the data, call the:

   saInstance`.getEncodedString(Map` *attrs*`, String` *secret*`, String` *encSecret*`)` method. If `encSecret` is null, the data is not encrypted. This is the same as calling:

   saInstance`.getEncodedString(Map` *attrs*`, String` *secret*`)`
3. To decrypt the data, call the:

   saInstance`.verifyEncodedString(String` *str*`, String` *secret*`, String` *encSecret*`)` method. If *encSecret* is null (or the data is not encrypted), decryption is not done. This would be equal to calling the *saInstance*`.verifyEncodedString(String` *str*`, String` *secret*`)` method.

See the OpenSSO Java API Reference for specifics on the parameters. This Javadoc can be found in the `docs` directory of the exploded `opensso.war`. Sample code can be found in the `saeIDPApp.jsp` and `saeSPApp.jsp` files included with the Secure Attribute Exchange sample. Be sure to include the SAE class files in your web application; they are included in Client SDK jars.

To Set Up the Identity Provider

1. Login to the OpenSSO console as the administrator.
2. Click the Federation tab.
3. Select the name of the appropriate hosted identity provider.
4. Click the Advanced tab.
5. Add one entry for each identity provider application as a value of the Application Security Configuration attribute. Each application should have one entry using the one of following formats:
   - `url=IDPAppURL|type=symmetric|secret=encoded-shared-secret|encryptionalgorithm=en`

     Sample Symmetric Entry

     ```
     url=http://www.idpapp.com:8080/idpapp/samples/saml2/sae/saeIDPApp.jsp|
     type=symmetric|secret=AQICNeg4ahYuOLmXG5w5yUgvmCUP0rr1HFGf|
     encryptionalgorithm=DES|encryptionkeystrength=56
     ```
   - `url=IDPAppURL|type=asymmetric|pubkeyalias=IDPApp-signing-certificate|encryptiona`

     Sample Asymmetric Entry

     ```
     url=http://www.idpapp.com:8080/idpapp/samples/saml2/sae/saeIDPApp.jsp|
     type=asymmetric|pubkeyalias=idpapp-cert|encryptionalgorithm=DES|
     encryptionkeystrength=56
     ```
6. Save the configuration.
7. Log out of the console.

To Set Up the Service Provider

1. Login to the OpenSSO console as the administrator.
2. Click the Federation tab.
3. Select the name of the appropriate hosted service provider.
4. Click the Advanced tab.
5. Add one entry for each service provider application as a value of the Application Security Configuration attribute. Each application should have one entry using the one of following formats:
   -

     ```
     url=SPAppURL|type=symmetric|secret=encoded-shared-secret|
     encryptionalgorithm=encryption-algorithm|
     encryptionkeystrength=encryption-strength
     ```

     Sample Symmetric Entry

```
url=http://www.spapp.com:8080/spapp/samples/saml2/sae/saeSPApp.jsp|
type=symmetric|secret=AQICNeg4ahYuOLmXG5w5yUgvmCUP0rr1HFGf|
encryptionalgorithm=DES|encryptionkeystrength=56
```

-

```
url=SPAppURL|type=asymmetric|privatekeyalias=
SP-signing-certificate-alias|encryptionalgorithm=encryption-algorithm|
encryptionkeystrength=encryption-strength|
pubkeyalias=SPApp-public-key-alias
```

  The `privatekeyalias` attribute may be omitted if the `signing-certificate-alias` is already configured in the service provider metadata.

  Sample Asymmetric Entry

```
url=http://www.spapp.com:8080/spapp/samples/saml2/sae/saeSPApp.jsp|
type=asymmetric|privatekeyalias=test|encryptionalgorithm=DES|
encryptionkeystrength=56|pubkeyalias=spapp-cert
```

6. Save the configuration.
7. Log out of the console.

To Test the Configurations

Use the SAE sample included with OpenSSO to test the configuration. You can find it in the `samples/saml2/sae` directory of the `opensso.war` or in the `saml2/sae` directory of the `opensso-client-jdk14.war` or the `opensso-client-jdk15.war`.

◆ ◆ ◆ CHAPTER 8

Configuring OpenSSO Enterprise 8.0 Update 1 in FIPS Mode

This chapter describes how to configure Sun OpenSSO Enterprise 8.0 Update 1 in Federal Information Processing Standards (FIPS) mode. The following procedures use Sun Java System Web Server 7.0 as the OpenSSO Enterprise web container, with the NSS Certificate DB (certdb) as the key/certificate store.

- "Before You Begin" on page 69
- "Configuring the NSS Database in FIPS Mode" on page 69
- "Enabling the FIPS-140 Standard for Sun Java System Web Server 7.0" on page 70
- "Configuring an OpenSSO Enterprise 8.0 Instance Using the Console" on page 71

Before You Begin

- After you enable FIPS mode, the bootstrap file cannot be decrypted (CR 6835816). Therefore, before you enable FIPS mode, backup the CONFIG_DIR/bootstrap file. Then, after you enable FIPS mode, replace the bootstrap file with the backup copy.
- If Web Server 7.0 has the Java security manager enabled, add the following additional permissions to the server.policy file:

```
permission java.security.SecurityPermission "insertProvider.Mozilla-JSS";
permission java.security.SecurityPermission "putProviderProperty.Mozilla-JSS";
permission java.security.SecurityPermission "removeProvider.Mozilla-JSS";
```

Configuring the NSS Database in FIPS Mode

Configure the NSS database in FIPS mode. For example, using the modutil command:

```
modutil -fips true -dbdir location-of-your-nss-database
```

For information about modutil, see http://www.mozilla.org/projects/security/pki/nss/tools/modutil.html.

To Enable the FIPS-140 Standard for Web Server 7.0

To enable the FIPS-140 Standard for Web Server 7.0, you must change the `certdb` password and enable FIPS mode as true. (By default, Web Server 7.0 sets the password to blank for its `certdb`.)

Set the password for the internal PKCS11 token using either the Web Server 7.0 Admin Console or CLI command.

Enabling the FIPS-140 Standard for Sun Java System Web Server 7.0

To enable the FIPS-140 Standard for Web Server 7.0, you must change the `certdb` password and enable FIPS mode as true. (By default, Web Server 7.0 sets the password to blank for its certdb.)

- "To Enable the FIPS-140 Standard for Web Server 7.0" on page 70
- "To Set the Password Using the Web Server 7.0 Admin Console" on page 70
- "To Set the Password Using Web Server 7.0 CLI" on page 71
- "To Enable FIPS mode for Web Server 7.0 With `modutil`" on page 71
- "To Pull the Changes into the Admin Server" on page 71
- "To Test the FIPS Mode Change" on page 71

▼ To Enable the FIPS-140 Standard for Web Server 7.0

● **Set the password for the internal PKCS11 token using either the Web Server 7.0 Admin Console or CLI command.**

▼ To Set the Password Using the Web Server 7.0 Admin Console

1. **Log in to the Admin Console.**

2. **Go to the configuration page in the Admin console.**

3. **Click the Certificates > PKCS11 Tokens tab.**

4. **Click the PKCS11 token name (default is internal).**

5. **Select the Token State checkbox.**

6 **Enter the password information.**

7 **Save your changes.**

▼ To Set the Password Using Web Server 7.0 CLI

- **Execute the `wadm` command. For example:**

```
wadm> set-token-pin -user=admin -password-file=admin.pwd -host=serverhost
-port=8989 -config=config1 -token=internal
```

▼ To Enable FIPS mode for Web Server 7.0 With `modutil`

- **Use `modutil` in the `WS70_ROOT/bin` directory. For example:**

`modutil -fips true -dbdir` *location-of-your-nss-database*

By default, the NSS database is in the `config` directory for the Web Server 7.0 instance.

▼ To Pull the Changes into the Admin Server

- **If you use `certutil` or `modutil` to modify files in the `config` directory, you must pull the changes into the Web Server 7.0 Admin Server. For example, using `wadm`:**

```
wadm pull-config -user=admin -password-file=_admin-pwfile_ -host=_server-host_
-port=8989 -config=config1 node1
```

▼ To Test the FIPS Mode Change

- **After you enable FIPS mode for Web Server 7.0, confirm that FIPS is enabled by restarting server. You should see a new prompt for the `certdb` password. For example:**

```
> Please enter the PIN for the "NSS FIPS 140-2 Certificate DB" token:
```

Configuring an OpenSSO Enterprise 8.0 Instance Using the Console

To enable a single OpenSSO Enterprise 8.0 instance in FIPS mode, you must first configure the instance to use the JSS-based implementation class for encryption, Secure Random, SSL sockets, and the HTTPS Protocol Handler.

Before You Begin

- `jss4.jar` - The `WS_INSTALL_DIR/lib/jss4.jar` file must be compatible with the NSS version you are using. If necessary, download a compatible `jss4.jar` file and copy it to the `WS_INSTALL_DIR/lib` directory.
- **Multiple OpenSSO Enterprise 8.0 instances** - If you are configuring multiple OpenSSO Enterprise 8.0 instances that are part of a site, first add and configure all instances in the site in non-FIPS mode. Then, after all instances are added and configured for the site, configure the instances in FIPS mode.

▼ To Configure an OpenSSO Enterprise 8.0 Instance Using the Console

1. **Log in to the OpenSSO Enterprise Administration Console.**

2. **Click Configuration, Servers and Sites, and then the Server Name instance.**

3. **Click the Security tab.**

4. **Click the Inheritance Settings button.**

5. **Uncheck the Encryption class, FIPS Mode, and Secure Random Factory Class properties.**

6. **Click Save and then Back to Server Profile.**

7. **Change Encryption class to** `com.iplanet.services.util.JSSEncryption`**.**

8. **Change Secure Random Factory Class to** `com.iplanet.am.util.JSSSecureRandomFactoryImpl`**.**

9. **Check Yes for FIPS Mode.**

10. **Click Save and then the Advanced tab.**

11. **Change the** `com.iplanet.security.SSLSocketFactoryImpl` **property to** `com.iplanet.services.ldap.JSSSocketFactory`**.**

12. **Click Add and add following property with the value:**

    ```
    opensso.protocol.handler.pkgs=com.iplanet.services.comm
    ```

13. **Click Add and add following property with the value:**

    ```
    com.iplanet.am.admin.cli.certdb.dir=path-to-FIPS-enabled-NSS-certdb
    ```

14 Click Save.

15 Restart the OpenSSO Enterprise server instance.

CHAPTER 9

Using OpenDS as a User Data Store for OpenSSO Enterprise 8.0 Update 1

This chapter provides instructions for installing and configuring OpenDS to store user profiles, authentication data, and policies, including:

Before You Begin

- Before following the instructions in this chapter, an OpenSSO Enterprise 8.0 Update 1 server must be already installed and configured on a supported web container.
- Creating a user data store using the OpenSSO Configurator is not supported in OpenSSO Enterprise 8.0 Update 1.
- Static groups with the `member` and `uniquemember` attributes have been tested and work as designed.

  If you use these attributes, then you must add the `groupOfNames` object class to the User Data Store Configuration page.
- Testing is in progress for groups with other (virtual) attributes such as `member`, `memberof`, and `ismemberof`.
- The Referential Integrity plug-in must be enabled in the OpenDS.

  The Referential Integrity plug-in ensures that when the groups are removed from the directory, all references in the users' entries are removed automatically. If the Referential Integrity plug-in is not enabled, you will see deleted groups displayed the users' profiles even after the group has been removed from the directory server.

When configuring OpenDS as a user store, keep the following in mind:

- OpenSSO Enterprise doesn't support the extensive password policy features provided by OpenDS.
- Only static groups are supported from the OpenSSO console for now.

To use OpenDS as the OpenSSO user data store, complete these steps. Detailed instructions are provided in the following sections.

1. Download and install OpenDS.
2. Add the OpenSSO schema and supporting OpenDS user management data to OpenDS..
3. Configure OpenSSO to Use OpenDS as the User Data Store.

To Download and Install OpenDS

1. Download an OpenDS build from the following website: `http://www.opends.org/promoted-builds/`
2. Follow the instructions in the *OpenDS Installation Guide* from the OpenDS website.
3. Enable the OpenDS Referential Integrity plug-in. See “Maintaining Referential Integrity” in the *OpenDS Installation Guide*.

To Add the OpenSSO Schema and Supporting OpenDS User Management Data to OpenDS

OpenSSO leverages certain LDAPv3-compliant attributes. Additionally, other object classes and user attributes are required and must be added to OpenDS to take full advantage of OpenSSO's functionality.

User schema is contained in the following file: *opensso_configuration_directory*/`am_remote_opends_schema.ldif`

1. To load the schema, run the following command:

   ```
   ldapmodify -h opends_host -p opends_port
        -D"RootDN" -w RootDN_password -c -f am_remote_opends_schema.ldif
   ```

2. To load the configuration for the `openssouser` and `ldapuser` users, special users required by OpenSSO, do the following:

   a. Download the text contained in the `configure_opends_userstore.ldif` file to a local file named `configure_opends_userstore.ldif` on your system.

   b. Edit the the following:

      - Change `ROOT_SUFFIX` to the root suffix of your user directory

- Change `OPENSSO_USER_PASSWD` to a password for the openssouser user
- Change `LDAP_USER_PASSWD` to a password for the ldapuser user
- Save the file.

c. Run the following command:

```
ldapmodify -h opends_host -p opends_port -D"RootDN" -w RootDN_password -c
  -a  -f  configure_opends_userstore.ldif
```

Configuring OpenSSO to Use OpenDS as the User Data Store

Once you have configured OpenDS, you can configure OpenSSO to work with OpenDS. Complete the following steps. Detailed instructions are provided in the following sections.

1. Create a new LDAPv3-compliant user data store.

   You can use the command-line interface or use the OpenSSO Administration Console.
2. Add OpenSSO object classes and user attributes to the user data store.
3. (Optional) Remove the OpenSSO schema from OpenDS.

To Create a New LDAPv3-Compliant User Data Store at the Command Line

The `ssoadm` command line tool must already be configured in the OpenSSO server.

1. Log into the OpenSSO host.
2. Download the text from Example 9–1 to a local file named `datastore_opends_attrs.txt` on you system. Modify the file as needed for your deployment. Be sure to replace the default OpenDS server name and port number with your OpenDS server name and port number. In the following example, the root suffix is `dc=opensso,dc-Java,dc=net`
3. Run the following command:

```
ssoadm create-datastore -m "OpenDS User Store" -t "LDAPv3" -D datastore_opends_attrs.txt
  -u amadmin -f /tmp/.pass_of_amadmin  -e /
```

   The file `.pass_of_amadmin` contains the amadmin user's password in plain text.
4. (Optional) To use this server as the LDAP authentication data store:

   a. Configure the LDAP authentication instance with the bind user `cn=ldapuser`.

   b. Configure the policy configuration service with the bind user `cn=ldapuser`

   For more information, see the *Sun OpenSSO Enterprise 8.0 Administration Reference*.

EXAMPLE 9–1 Example for LDAPv3-Compliant User Data Store

```
com.iplanet.am.ldap.connection.delay.between.retries=1000
RequiredValueValidator=
sun-idrepo-ldapv3-config-active=Active
sun-idrepo-ldapv3-config-auth-naming-attr=uid
sun-idrepo-ldapv3-config-authenticatable-type=User
sun-idrepo-ldapv3-config-authid=cn=openssouser,ou=opensso adminusers,dc=opensso,dc=java,dc=net
sun-idrepo-ldapv3-config-authpw=amsecret12
sun-idrepo-ldapv3-config-cache-enabled=false
sun-idrepo-ldapv3-config-cache-size=10240
sun-idrepo-ldapv3-config-cache-ttl=600
sun-idrepo-ldapv3-config-connection_pool_max_size=10
sun-idrepo-ldapv3-config-connection_pool_min_size=1
sun-idrepo-ldapv3-config-createuser-attr-mapping=cn
sun-idrepo-ldapv3-config-createuser-attr-mapping=sn
sun-idrepo-ldapv3-config-dftgroupmember=
sun-idrepo-ldapv3-config-errorcodes=80
sun-idrepo-ldapv3-config-errorcodes=81
sun-idrepo-ldapv3-config-errorcodes=91
sun-idrepo-ldapv3-config-filterrole-attributes=
sun-idrepo-ldapv3-config-filterrole-objectclass=
sun-idrepo-ldapv3-config-group-attributes=cn
sun-idrepo-ldapv3-config-group-attributes=description
sun-idrepo-ldapv3-config-group-attributes=dn
sun-idrepo-ldapv3-config-group-attributes=iplanet-am-group-subscribable
sun-idrepo-ldapv3-config-group-attributes=objectclass
sun-idrepo-ldapv3-config-group-attributes=ou
sun-idrepo-ldapv3-config-group-attributes=uniqueMember
sun-idrepo-ldapv3-config-group-container-name=ou
sun-idrepo-ldapv3-config-group-container-value=groups
sun-idrepo-ldapv3-config-group-objectclass=groupofuniquenames
sun-idrepo-ldapv3-config-group-objectclass=iplanet-am-managed-group
sun-idrepo-ldapv3-config-group-objectclass=iplanet-am-managed-static-group
sun-idrepo-ldapv3-config-group-objectclass=top
sun-idrepo-ldapv3-config-groups-search-attribute=cn
sun-idrepo-ldapv3-config-groups-search-filter=(objectclass=groupOfUniqueNames)
sun-idrepo-ldapv3-config-idletimeout=0
sun-idrepo-ldapv3-config-inactive=Inactive
sun-idrepo-ldapv3-config-isactive=inetuserstatus

sun-idrepo-ldapv3-config-ldap-server=<hostName.domain:portNumber>

sun-idrepo-ldapv3-config-max-result=1000
sun-idrepo-ldapv3-config-memberof=
sun-idrepo-ldapv3-config-memberurl=memberUrl
sun-idrepo-ldapv3-config-nsrole=
sun-idrepo-ldapv3-config-nsroledn=
```

EXAMPLE 9–1 Example for LDAPv3-Compliant User Data Store *(Continued)*

```
sun-idrepo-ldapv3-config-nsrolefilter=
sun-idrepo-ldapv3-config-numretires=3
sun-idrepo-ldapv3-config-organization_name=dc=opensso,dc=java,dc=net
sun-idrepo-ldapv3-config-people-container-name=ou
sun-idrepo-ldapv3-config-people-container-value=people
sun-idrepo-ldapv3-config-psearch-filter=(objectclass=*)
sun-idrepo-ldapv3-config-psearch-scope=SCOPE_SUB
sun-idrepo-ldapv3-config-psearchbase=dc=opensso,dc=java,dc=net
sun-idrepo-ldapv3-config-referrals=true
sun-idrepo-ldapv3-config-search-scope=SCOPE_ONE
sun-idrepo-ldapv3-config-service-attributes=
sun-idrepo-ldapv3-config-ssl-enabled=false
sun-idrepo-ldapv3-config-time-limit=10
sun-idrepo-ldapv3-config-uniquemember=uniqueMember
sun-idrepo-ldapv3-config-user-attributes=adminRole
sun-idrepo-ldapv3-config-user-attributes=authorityRevocationList
sun-idrepo-ldapv3-config-user-attributes=caCertificate
sun-idrepo-ldapv3-config-user-attributes=cn
sun-idrepo-ldapv3-config-user-attributes=distinguishedName
sun-idrepo-ldapv3-config-user-attributes=dn
sun-idrepo-ldapv3-config-user-attributes=employeeNumber
sun-idrepo-ldapv3-config-user-attributes=facsimileTelephoneNumber
sun-idrepo-ldapv3-config-user-attributes=givenName
sun-idrepo-ldapv3-config-user-attributes=homePhone
sun-idrepo-ldapv3-config-user-attributes=homePostalAddress
sun-idrepo-ldapv3-config-user-attributes=inetUserHttpURL
sun-idrepo-ldapv3-config-user-attributes=inetUserStatus
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-auth-configuration
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-session-add-session-listener-on-all-sessions
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-session-destroy-sessions
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-session-get-valid-sessions
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-session-max-caching-time
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-session-max-idle-time
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-session-max-session-time
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-session-quota-limit
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-session-service-status
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-static-group-dn
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-account-life
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-admin-start-dn
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-alias-list
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-auth-config
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-auth-modules
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-failure-url
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-federation-info
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-federation-info-key
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-login-status
```

EXAMPLE 9–1 Example for LDAPv3-Compliant User Data Store *(Continued)*

```
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-password-reset-force-reset
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-password-reset-options
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-password-reset-question-answer
sun-idrepo-ldapv3-config-user-attributes=iplanet-am-user-success-url
sun-idrepo-ldapv3-config-user-attributes=mail
sun-idrepo-ldapv3-config-user-attributes=manager
sun-idrepo-ldapv3-config-user-attributes=memberOf
sun-idrepo-ldapv3-config-user-attributes=mobile
sun-idrepo-ldapv3-config-user-attributes=ds-pwp-account-disabled
sun-idrepo-ldapv3-config-user-attributes=objectClass
sun-idrepo-ldapv3-config-user-attributes=pager
sun-idrepo-ldapv3-config-user-attributes=postalAddress
sun-idrepo-ldapv3-config-user-attributes=postofficebox
sun-idrepo-ldapv3-config-user-attributes=preferredlanguage
sun-idrepo-ldapv3-config-user-attributes=preferredLocale
sun-idrepo-ldapv3-config-user-attributes=preferredtimezone
sun-idrepo-ldapv3-config-user-attributes=secretary
sun-idrepo-ldapv3-config-user-attributes=sn
sun-idrepo-ldapv3-config-user-attributes=street
sun-idrepo-ldapv3-config-user-attributes=sun-fm-saml2-nameid-info
sun-idrepo-ldapv3-config-user-attributes=sun-fm-saml2-nameid-infokey
sun-idrepo-ldapv3-config-user-attributes=sunAMAuthInvalidAttemptsData
sun-idrepo-ldapv3-config-user-attributes=sunIdentityMSISDNNumber
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerDiscoEntries
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPAddressCard
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPCommonNameAltCN
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPCommonNameCN
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPCommonNameFN
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPCommonNameMN
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPCommonNamePT
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPCommonNameSN
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPDemographicsAge
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPDemographicsBirthDay
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPDemographicsDisplayLanguage
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPDemographicsLanguage
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPDemographicsTimeZone
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPEmergencyContact
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPEmploymentIdentityAltO
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPEmploymentIdentityJobTitle
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPEmploymentIdentityOrg
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPEncryPTKey
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPFacadegreetmesound
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPFacadeGreetSound
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPFacadeMugShot
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPFacadeNamePronounced
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPFacadeWebSite
```

EXAMPLE 9–1 Example for LDAPv3-Compliant User Data Store *(Continued)*

```
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPInformalName
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPLegalIdentityAltIdType
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPLegalIdentityAltIdValue
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPLegalIdentityDOB
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPLegalIdentityGender
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPLegalIdentityLegalName
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPLegalIdentityMaritalStatus
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPLegalIdentityVATIdType
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPLegalIdentityVATIdValue
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPMsgContact
sun-idrepo-ldapv3-config-user-attributes=sunIdentityServerPPSignKey
sun-idrepo-ldapv3-config-user-attributes=telephoneNumber
sun-idrepo-ldapv3-config-user-attributes=uid
sun-idrepo-ldapv3-config-user-attributes=userCertificate
sun-idrepo-ldapv3-config-user-attributes=userPassword
sun-idrepo-ldapv3-config-user-objectclass=inetadmin
sun-idrepo-ldapv3-config-user-objectclass=inetorgperson
sun-idrepo-ldapv3-config-user-objectclass=inetUser
sun-idrepo-ldapv3-config-user-objectclass=iplanet-am-managed-person
sun-idrepo-ldapv3-config-user-objectclass=iplanet-am-user-service
sun-idrepo-ldapv3-config-user-objectclass=iPlanetPreferences
sun-idrepo-ldapv3-config-user-objectclass=organizationalPerson
sun-idrepo-ldapv3-config-user-objectclass=person
sun-idrepo-ldapv3-config-user-objectclass=sunFederationManagerDataStore
sun-idrepo-ldapv3-config-user-objectclass=sunFMSAML2NameIdentifier
sun-idrepo-ldapv3-config-user-objectclass=sunIdentityServerLibertyPPService
sun-idrepo-ldapv3-config-user-objectclass=top
sun-idrepo-ldapv3-config-users-search-attribute=uid
sun-idrepo-ldapv3-config-users-search-filter=(objectclass=inetorgperson)
sun-idrepo-ldapv3-ldapv3Generic=
sunIdRepoAttributeMapping=
sunIdRepoClass=com.sun.identity.idm.plugins.ldapv3.LDAPv3Repo
sunIdRepoSupportedOperations=group=read,create,edit,delete
sunIdRepoSupportedOperations=realm=read,create,edit,delete,service
sunIdRepoSupportedOperations=user=read,create,edit,delete,service
```

To Create a New LDAPv3-compliant User Data Store Using the OpenSSO Administration Console

1. Log in to the OpenSSO administration console.
2. Click Access, Top-level Realm, and Data Stores.
3. On the Data Stores tab, click the Generic LDAP v3 user data store.
4. On the Generic LDAP v3 data store page, add the LDAP User object classes and attributes.

a. If they do not already exist, add the following LDAP User Object Classes:

```
inetadmin
inetorgperson
inetUser
iplanet-am-managed-person
iplanet-am-user-service
iPlanetPreferences
organizationalPerson
person
sunFederationManagerDataStore
sunFMSAML2NameIdentifier
sunIdentityServerLibertyPPService
top
```

b. If they do not already exist, add the following LDAP User Attributes:

```
adminRole
authorityRevocationList
caCertificate
cn
distinguishedName
dn
ds-pwp-account-disabled
employeeNumber
facsimileTelephoneNumber
givenName
homePhone
homePostalAddress
inetUserHttpURL
inetUserStatus
iplanet-am-auth-configuration
iplanet-am-session-add-session-listener-on-all-sessions
iplanet-am-session-destroy-sessions
iplanet-am-session-get-valid-sessions
iplanet-am-session-max-caching-time
iplanet-am-session-max-idle-time
iplanet-am-session-max-session-time
iplanet-am-session-quota-limit
iplanet-am-session-service-status
iplanet-am-static-group-dn
iplanet-am-user-account-life
iplanet-am-user-admin-start-dn
iplanet-am-user-alias-list
iplanet-am-user-auth-config
iplanet-am-user-auth-modules
iplanet-am-user-failure-url
iplanet-am-user-federation-info
iplanet-am-user-federation-info-key
iplanet-am-user-login-status
```

```
iplanet-am-user-password-reset-force-reset
iplanet-am-user-password-reset-options
iplanet-am-user-password-reset-question-answer
iplanet-am-user-success-url
mail
manager
memberOf
mobile
objectClass
pager
postalAddress
postofficebox
preferredlanguage
preferredLocale
preferredtimezone
secretary
sn
street
sunAMAuthInvalidAttemptsData
sun-fm-saml2-nameid-info
sun-fm-saml2-nameid-infokey
sunIdentityMSISDNNumber
sunIdentityServerDiscoEntries
sunIdentityServerPPAddressCard
sunIdentityServerPPCommonNameAltCN
sunIdentityServerPPCommonNameCN
sunIdentityServerPPCommonNameFN
sunIdentityServerPPCommonNameMN
sunIdentityServerPPCommonNamePT
sunIdentityServerPPCommonNameSN
sunIdentityServerPPDemographicsAge
sunIdentityServerPPDemographicsBirthDay
sunIdentityServerPPDemographicsDisplayLanguage
sunIdentityServerPPDemographicsLanguage
sunIdentityServerPPDemographicsTimeZone
sunIdentityServerPPEmergencyContact
sunIdentityServerPPEmploymentIdentityAltO
sunIdentityServerPPEmploymentIdentityJobTitle
sunIdentityServerPPEmploymentIdentityOrg
sunIdentityServerPPEncryPTKey
sunIdentityServerPPFacadegreetmesound
sunIdentityServerPPFacadeGreetSound
sunIdentityServerPPFacadeMugShot
sunIdentityServerPPFacadeNamePronounced
sunIdentityServerPPFacadeWebSite
sunIdentityServerPPInformalName
sunIdentityServerPPLegalIdentityAltIdType
sunIdentityServerPPLegalIdentityAltIdValue
```

```
sunIdentityServerPPLegalIdentityDOB
sunIdentityServerPPLegalIdentityGender
sunIdentityServerPPLegalIdentityLegalName
sunIdentityServerPPLegalIdentityMaritalStatus
sunIdentityServerPPLegalIdentityVATIdType
sunIdentityServerPPLegalIdentityVATIdValue
sunIdentityServerPPMsgContact
sunIdentityServerPPSignKey
telephoneNumber
uid
userCertificate
userPassword
```

5. Click Save.

To Remove the OpenSSO schema from OpenDS

At some point if you want to remove the schema you added to OpenDS in these instructions, log into the OpenDS host and run the following command

```
ldapmodify -h opends-host -p opends_port -D"cn=directory manager" /
-w  password -c -f  remove_am_remote_opends_schema.ldif
```

This will remove the OpenSSO user schema.

Troubleshooting

When an administrator tries to change a user's password using the OpenSSO console or CLI or using the `ldap-modify{}`utility, if the following message is displayed in the OpenDS access log: "You do not have sufficient privileges to reset user passwords, " then the password-reset privilege is not configured.

In OpenDS, you must add the password-reset privilege and assign it to an administrator. In the following example, the administrator is named `openssouser`. This privilege enables the administrator to reset the passwords of other users in the directory. The password-reset privilege works in association with the OpenDS ACIs that are set in the target.

About the OpenSSO User Data Store

OpenSSO uses an identity repository to store user data such as users and groups. During OpenSSO Enterprise installation, you must specify which user data store you want to use. For example, you can use Sun Directory Server Enterprise Edition, Microsoft Active Directory, IBM Tivoli, or OpenDS.

Use the tables in this section to help you determine which user data store meets your needs.

- "Supported Features for Various Directory Servers" on page 85
- "Data Stores and Supported Operations" on page 86
- "Additional Information for Determining Which User Data Store to Use" on page 87

Supported Features for Various Directory Servers

In the following table, a Policy Subject refers to the "who" part of the policy definition. The Policy Subject specifies the members or entities to which the policy applies. Policy Condition refers to the additional restrictions with which the policy applies. Examples are a specified window of time in a day, a specified IP address, or a specified authentication method.

OpenSSO Enterprise Feature	Sun Directory Server LDAPv3	OpenDS	Microsoft Active Directory LDAPv3	IBM Tivoli Directory	Generic LDAPv3
User Data Storage	Yes	Yes	Yes	Yes	No
Configuration Data Storage	Yes	Yes	No	No	No
AMSDK (legacy)	Yes	No	No	No	No
LDAP Authentication	Yes	Yes	Yes	Yes	Yes
Membership Authentication	Yes	Yes	No	Yes	No
Active Directory Authentication	Not Applicable	Not Applicable	Yes, with limitations	Not Applicable	Not Applicable
Policy Subjects and Policy LDAP Filter Condition	Yes	Yes	Yes	Yes	Yes
Password Reset	Yes	Yes	No	No	No
Account Lockout	Yes	Yes	No	Yes	No
Cert Authentication	Yes	Yes	Yes	Yes	Yes
MSISDN Authentication	Yes	Yes	Yes	Yes	Yes
Data Store Authentication (through LDAPv3 user store configuration)	Yes	Yes	Yes	Yes	Yes

OpenSSO Enterprise Feature	Sun Directory Server LDAPv3	OpenDS	Microsoft Active Directory LDAPv3	IBM Tivoli Directory	Generic LDAPv3
User creation with Password and Password Management	Yes	Yes	No	Yes	Yes
Password Policy	Yes	Limited support	No	No	No

Data Stores and Supported Operations

The following table summarizes the user management operations supported through the IDRepo interface for various user data stores. An interface has been implemented specifically for Sun Directory Server and Microsoft Active Directory. The default implementation of this interface can be used and supported for any LDAPv3 user repository.

Feature	Sun Directory Server LDAPv3	OpenDS	Microsoft Active Directory LDAPv3	IBM Tivoli Directory	AMSDK (Legacy)
Create User	Yes	Yes	Yes*	Yes	Yes
Modify User	Yes	Yes	Yes*	Yes	Yes
Delete User	Yes	Yes	Yes*	Yes	Yes
Create Role	Yes	No	No	No	Yes
Modify Role	Yes	No	No	No	Yes
Delete Role	Yes	No	No	No	Yes
Assign Role	Yes	No	No	No	Yes
Evaluate Role for Membership	Yes	No	No	No	Yes
Create Group	Yes	Yes	Yes*	Yes**	Yes
Modify Group	Yes	Yes	Yes*	Yes**	Yes
Delete Group	Yes	Yes	Yes*	Yes**	Yes
Evaluate Group for Membership	Yes	Yes	Yes*	Yes**	Yes
Federation Attributes	Yes	Yes	Yes	Yes	Yes

Feature	Sun Directory Server LDAPv3	OpenDS	Microsoft Active Directory LDAPv3	IBM Tivoli Directory	AMSDK (Legacy)
*Some limitations exist, or additional configuration is required. **See the limitations described in the next section.					

Additional Information About Using IBM Tivoli Directory Server Configured as the IDRepo Data Store

IBM Tivoli Directory Server's groups can be Static, Dynamic, and Nested. However, the OpenSSO Enterprise IDRepo framework (IDRepo DataStore) supports only the

Static group. A Static group defines each member individually using either of the following:

- Structural ObjectClass: `groupofNames`, `groupOfUniqueNames`, `accessGroup`, or `accessRole`
- Auxilary ObjectClass: `ibm-staticgroup` or `ibm-globalAdminGroup`

A Static group using the Structural ObjectClass `groupOfNames` and `groupOfUniqueNames` requires at least one member for `ObjectClass groupOfNames` or one `uniquemember` for `groupOfUniqueNames`. The Static group using the ObjectClass `ibm-staticgroup` does not have this requirement. The ObjectClass `ibm-staticgroup` is the only ObjectClass for which members are optional; all other object classes require at least one member.

OpenSSO Enterprise supports only one ObjectClass for groups. If you choose a type of group with an ObjectClass that requires at leas one member, then a user value must be present. This user will automatically be added to the group when a group is created. You can remove this user from the group afterward if you don't

want this user to be a member of the group.

The value for the filter for searching of groups must the value specified by the chosen LDAP Group ObjectClass.

Most IBM Tivoli groups require at least one member when the group is created. When a group is created using the OpenSSO Enterprise console, no users are assigned to the group by default. Since IBM Tivoli has this restriction, when a group is created, the default user or member `cn=auser1,dc=opensso,dc=java,dc=net` is always automatically created and added to the group.

Additional Information for Determining Which User Data Store to Use

- Account Lockout locks a user account based on the policies defined in the Directory Server.

  For example, the user account can be locked when a specified number of login failures occurs.

- The key difference between using a policy LDAP subject and the IDRepo interface subject is that policy LDAP subjects don't provide caching and notification updates. The `AMIdentity Subject` does provide caching an notification updates.

  The policy LDAP subjects provide LDAP Organization, Role (if Sun Directory Server), Group, and User subjects to evaluate membership of a user and determine if

  the user belongs to one of these subjects. The same result can be obtained using the Identity Repository (IDRepo) interface subject named `AMIdentity Subject`. This interface subject was introduced when the product was named Access Manager 7.0. You can develop a policy subject for a JDBC user store. Authentication also supports the JDBC repository through the JDBC authentication module.
- The IDRepo interface provides basic user management features for user, group,

  role, and Access Manager policy agent entities.

  This interface enables OpenSSO Enterprise to support any user repository through the development of new plug-ins. Although limited to Sun Directory Server, Microsoft Active Directory, and IBM Tivoli Directory today, the IDRepo interface could potentially be expanded to include any LDAPv3 directory server such as OpenLDAP or Novel Directory for JDBC, flat files, and so forth.
- Prior to Access Manager 7.0, user management was supported using Access Manager object classes and attributes in addition to using specific features from Sun

  Directory Server. This support still exists through the legacy AMSDK interface. But this support is deprecated and will be removed future releases.

CHAPTER 10

Using the ASP.NET Fedlet with OpenSSO Enterprise 8.0 Update 1

The OpenSSO Fedlet is a small archive that can be embedded into a service provider's web application to allow for SAMLv2 single sign-on between an identity provider instance of OpenSSO and the service provider application - WITHOUT installing OpenSSO on the service provider side. With the release of OpenSSO Enterprise 8.0 Update 1, the Fedlet technology has been extended to the ASP.NET platform.

OpenSSO Enterprise 8.0 Update 1 includes the `Fedlet.dll`, template metadata files, and a sample ASP.NET application for testing the communications. The `Fedlet.dll` initiates single sign-on with an identity provider and enables the receipt of an authentication response by the service provider using an HTTP-POST binding.

To configure for communications with the ASP.NET Fedlet, you need to configure the identity provider, the service provider, the Fedlet, and the service provider application.

- "To Configure the Identity Provider" on page 89
- "To Configure the Service Provider and the ASP.NET Fedlet" on page 90
- "To Configure the Sample Application and Test the ASP.NET Fedlet" on page 91
- "To Integrate the ASP.NET Fedlet with an Existing Application" on page 92

To Configure the Identity Provider

1. Create the hosted identity provider using the Common Tasks work flow in the OpenSSO Enterprise console.

   You will need the name of the circle of trust in the next procedure, "To Configure the Service Provider and the ASP.NET Fedlet" on page 90.

2. Export the identity provider's standard metadata file.

   `idp.xml` can be exported by accessing the export metadata page at `http://idp-machine.domain:8080/opensso/saml2/jsp/exportmetadata.jsp`.

3. Register the remote service provider using the modified standard metadata file `sp.xml` and the Register Remote Service Provider work flow in the OpenSSO Enterprise console.

This step is done after you have finished "To Configure the Service Provider and the ASP.NET Fedlet" on page 90.

To Configure the Service Provider and the ASP.NET Fedlet

1. Download the OpenSSO Enterprise ZIP archive to the service provider machine and unzip it.
2. Unzip the `Fedlet-unconfigured.zip` in the `/opensso/fedlet/` folder.
3. Move the `/opensso/fedlet/asp.net/` folder to a temporary directory.
4. Change to the `/tmp/asp.net/conf` directory.
5. Make copies of the template files.
   - Copy `sp.xml-template` to `sp.xml`.
   - Copy `sp-extended.xml-template` to `sp-extended.xml`.
   - Copy `idp-extended.xml-template` to `idp-extended.xml`.
   - Copy `fedlet.cot-template` to `fedlet.cot`.
6. Swap out the following tags in the copied metadata files.
   - Replace FEDLET_COT with the name of the circle of trust of which the remote identity provider and the local service provider are members.
   - Replace FEDLET_ENTITY_ID with a unique identifier used to locate the Fedlet. This value is analogous to the service provider `EntityID`. The `EntityID` attribute is under the `EntityDescriptor` element that is passed to the service provider as part of the XML exchange. The Name attribute of a configured entity provider when looking in the OpenSSO console is the value of the `EntityID`.
   - Replace FEDLET_URL with the URL of the Fedlet; for example, `http://sp-machine.domain/SampleApp/fedletapplication.aspx`.
   - Replace IDP_ENTITY_ID with the entity ID of the remote identity provider. The `EntityID` attribute is under the `EntityDescriptor` element that is passed to the service provider as part of the XML exchange. The Name attribute of a configured entity provider in the OpenSSO console is the value of the `EntityID`.

At this point, return to the identity provider machine to register the service provider using the modified `sp.xml` file and making sure to associate the service provider and the identity provider with the same circle of trust.

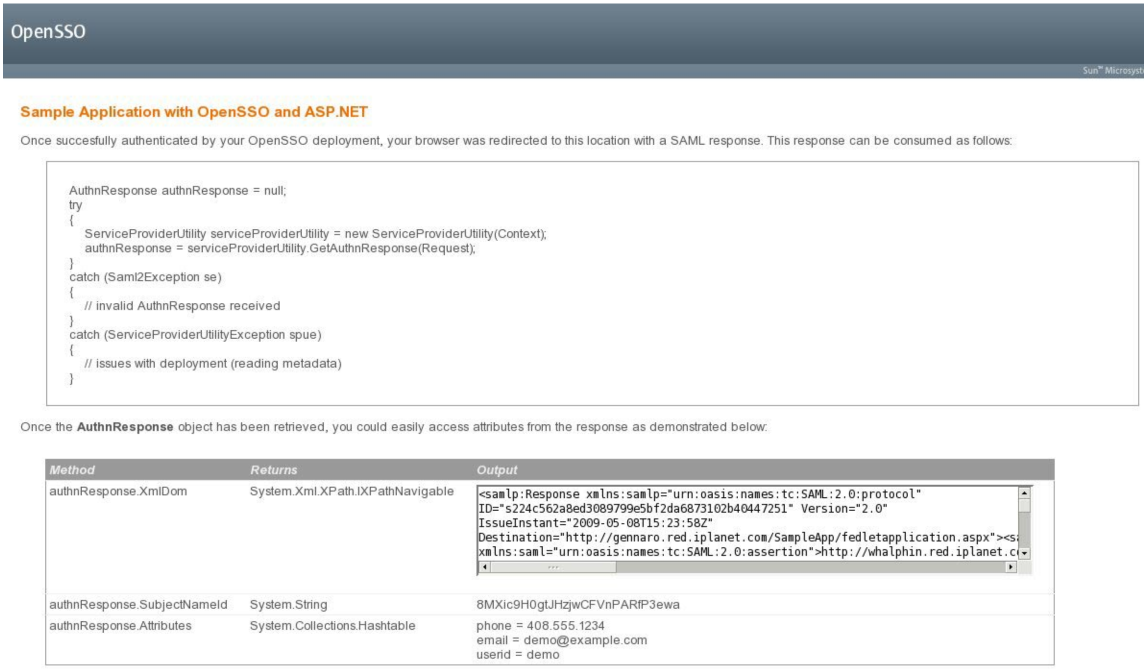
To Configure the Sample Application and Test the ASP.NET Fedlet

The Sample Application should be deployed using ASP.NET version 3.5 and Microsoft Internet Information Server versions 6 or 7.

1. Navigate to the `/tmp/asp.net/conf` folder on the service provider machine.
2. Copy the modified metadata files `idp-extended.xml`, `sp.xml`, `sp-extended.xml`, and `fedlet.cot` to `/tmp/asp.net/SampleApp/App_Data/`.
3. Copy the remote identity provider's standard metadata file to the service provider machine.

   Be sure the file is named `idp.xml`.
4. Place `idp.xml` in `/tmp/asp.net/SampleApp/App_Data/`.
5. Confirm that the `Fedlet.dll` is in the Sample Application's `/tmp/asp.net/SampleApp/bin/` folder.
6. Within Internet Information Server (IIS), create a virtual directory using the `/tmp/asp.net/SampleApp/` directory.
   - IIS 6 (Windows 2003) has Add Virtual Directory. Be sure to have Read and Script permissions set for the application.
   - IIS 7 (Windows 2008 and Vista) has Add Application with no additional options required to be set.
7. Open the Sample Application in your browser using the URL, `http://sp.example.com/SampleApp`
8. Click the IDP Initiated SSO link to perform identity provider-initiated single sign-on.
9. Enter the appropriate user credentials.

   The OpenSSO user **demo** with a password of **changeit** will work. After a successful authentication, the `fedletapplication.aspx` page is displayed with access to the `AuthnResponse` information. Click on the thumbnail to see a screenshot.



Sample Application with OpenSSO and ASP.NET

Once succesfully authenticated by your OpenSSO deployment, your browser was redirected to this location with a SAML response. This response can be consumed as follows:

```
AuthnResponse authnResponse = null;
try
{
    ServiceProviderUtility serviceProviderUtility = new ServiceProviderUtility(Context);
    authnResponse = serviceProviderUtility.GetAuthnResponse(Request);
}
catch (Saml2Exception se)
{
    // invalid AuthnResponse received
}
catch (ServiceProviderUtilityException spue)
{
    // issues with deployment (reading metadata)
}
```

Once the **AuthnResponse** object has been retrieved, you could easily access attributes from the response as demonstrated below:

Method	Returns	Output

authnResponse.XmlDom	System.Xml.XPath.IXPathNavigable	`<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"`
		`ID="s224c562a8ed3089799e5bf2da6873102b40447251" Version="2.0"`
		`IssueInstant="2009-05-08T15:23:58Z"`
		`Destination="http://gennaro.red.iplanet.com/SampleApp/fedletapplication.aspx"><s`
		`xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">http://whalphin.red.iplanet.c`

authnResponse.SubjectNameId	System.String	8MXic9H0gtJHzjwCFVnPARfP3ewa

authnResponse.Attributes	System.Collections.Hashtable	phone = 408.555.1234
		email = demo@example.com
		userid = demo

FIGURE 10–1 Sample Application with OpenSSO and ASP.NET

To Integrate the ASP.NET Fedlet with an Existing Application

The Sample Application demonstrates how to retrieve attributes and subject information from the SAMLv2 assertion in an `AuthnResponse` object. The following code can be integrated in custom applications to do the same. It is expected to be placed in an `aspx` page or ASP.NET URI to receive the authentication response in an HTTP-POST binding.

```
AuthnResponse authnResponse = null;
try
{
    ServiceProviderUtility spu = new ServiceProviderUtility(Context);
    authnResponse = spu.GetAuthnResponse(Request);
}
catch (Saml2Exception se)
{
    // invalid AuthnResponse received
}
catch (ServiceProviderUtilityException spue)
{
    // issues with deployment (reading metadata)
}
```

For more information about the Fedlet, see the *Sun OpenSSO Enterprise 8.0 Technical Overview* and the *Sun OpenSSO Enterprise 8.0 Deployment Planning Guide*.