Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0

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

Part No: 820–5986
November 2008

List of Remarks

Contents

Preface

Sun OpenSSO Enterprise 8.0 provides a comprehensive solution for protecting network resources that integrates authentication and authorization services, policy agents, and identity federation. This Preface to the *Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0* contains the following sections:

About This Guide

Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0 provides detailed instructions for enabling the Security Assertion Markup Language version 2 (SAML v2) in a federated environment. The procedures in this guide were used to build, deploy and test this deployment in a lab facility but you can adapt these instructions to suit your company's needs. Best results will be obtained by executing the tasks in the exact sequence in which they are presented. Use the Table of Contents as a master task list. Tasks are numbered for your convenience.

Caution – If deviating from the task sequence or details described in this guide, you should refer to the relevant product documentation for information or necessary requirements.

Before You Read This Book

This book is intended for use by IT administrators and software developers who implement a web access platform using Sun servers and software. Readers of this guide should be familiar with the following technologies:

- Security Assertion Markup Language (SAML) version 2
- eXtensible Markup Language (XML)

- Lightweight Directory Access Protocol (LDAP)
- Java™
- JavaServer Pages™ (JSP)
- HyperText Transfer Protocol (HTTP)
- HyperText Markup Language (HTML)

Related Documentation

Related documentation is available as follows:

- "OpenSSO Enterprise 8.0 Core Documentation" on page 14
- "Related Product Documentation" on page 15

OpenSSO Enterprise 8.0 Core Documentation

The OpenSSO Enterprise 8.0 core documentation set contains the following titles:

- The *Sun OpenSSO Enterprise 8.0 Release Notes* will be available online after the product is released. It gathers an assortment of last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.
- The *Sun OpenSSO Enterprise 8.0 Technical Overview* provides high level explanations of how OpenSSO Enterprise components work together to protect enterprise assets and web-based applications. It also explains basic concepts and terminology.
- The *Sun OpenSSO Enterprise 8.0 Deployment Planning Guide* provides planning and deployment solutions for OpenSSO Enterprise based on the solution life cycle
- The *Deployment Example: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0* provides instructions for building an OpenSSO solution incorporating authentication, authorization and access control. Procedures for load balancing and session failover are also included.
- The *Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0* (this guide) provides instructions for building an OpenSSO solution incorporating SAML v2 federation. Installation and configuration procedures are included.
- The *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide* provides information for installing and configuring OpenSSO Enterprise.
- The *Sun OpenSSO Enterprise 8.0 Performance Tuning Guide* provides information on how to tune OpenSSO Enterprise and its related components for optimal performance.
- The *Sun OpenSSO Enterprise 8.0 Administration Guide* describes administrative tasks such as *how to create a realm* and *how to configure a policy*. Most of the tasks described can be performed using the administration console as well as the `ssoadm` command line utilities.

- The *Sun OpenSSO Enterprise 8.0 Administration Reference* is a guide containing information about the command line interfaces, configuration attributes, internal files, and error codes. This information is specifically formatted for easy searching.
- The *Sun OpenSSO Enterprise 8.0 Developer's Guide* offers information on how to customize OpenSSO Enterprise and integrate its functionality into an organization's current technical infrastructure. It also contains details about the programmatic aspects of the product and its API.
- The *Sun OpenSSO Enterprise 8.0 C API Reference for Application and Web Policy Agent Developers* provides summaries of data types, structures, and functions that make up the public OpenSSO Enterprise C SDK for application and web agent development.
- The *Sun OpenSSO Enterprise 8.0 Java API Reference* provides information about the implementation of Java packages in OpenSSO Enterprise.
- The *Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for Web Agents* and *Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for J2EE Agents* provide an overview of the policy functionality and policy agents available for OpenSSO Enterprise.

Updates to the *Release Notes* and links to modifications of the core documentation can be found on the OpenSSO Enterprise page at docs.sun.com. Updated documents will be marked with a revision date.

Related Product Documentation

The following table provides links to documentation for related products.

TABLE P–1 Related Product Documentation

Product	Link
Sun Java System Directory Server 6.3	`http://docs.sun.com/coll/1224.4`
Sun Java System Web Server 7.0 Update 3	`http://docs.sun.com/coll/1653.3`
Sun Java System Application Server 9.1	`http://docs.sun.com/coll/1343.4`
Sun Java System Message Queue 4.1	`http://docs.sun.com/coll/1307.3`
Sun Java System Web Proxy Server 4.0.6	`http://docs.sun.com/coll/1311.6`
Sun Java System Identity Manager 8.0	`http://docs.sun.com/coll/1514.5`

Searching Sun Product Documentation

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

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

For example, to search for “broker,” type the following:

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

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

Documentation, Support, and Training

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

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

Third-Party Web Site References

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

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

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the title of this book is *Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0*, and the part number is 820–5986.

Typographical Conventions

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

TABLE P–2 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Default Paths and Directory Names

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

TABLE P–3 Default Paths and Directory Names

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

TABLE P–3 Default Paths and Directory Names *(Continued)*

Term	Description
OpenSSO-Deploy-base	Represents the directory where the web container deploys `opensso.war`. The location varies depending on the web container used. To determine the value of *OpenSSO-Deploy-base*, view the file in the `.openssocfg` directory (located in the home directory of the user who deployed `opensso.war`). For example, consider this scenario with Application Server 9.1 as the web container: ■ Application Server 9.1 is installed in the default directory: `/opt/SUNWappserver`. ■ The `opensso.war` file is deployed by super user (`root`) on Application Server 9.1. The `.openssocfg` directory is in the `root` home directory (`/`), and the file name in `.openssocfg` is `AMConfig_opt_SUNWappserver_domains_domain1_applications_j2ee-modules_opensso_`. Thus, the value for *OpenSSO-Deploy-base* is: `/opt/SUNWappserver/domains/domain1/applications/j2ee-modules/opensso`
ConfigurationDirectory	Represents the name of the directory specified during the initial configuration of OpenSSO Enterprise. The default is `opensso` in the home directory of the user running the Configurator. Thus, if the Configurator is run by `root`, *ConfigurationDirectory* is `/opensso`.

PART I

About This Deployment

This first part of *Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0* provides introductory material and an overview of the deployment. It contains the following chapters:

- Chapter 1, “Components and Features”
- Chapter 2, “Technical Overview”
- Chapter 3, “Before You Begin”

CHAPTER 1

Components and Features

Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0 provides detailed instructions for enabling the Security Assertion Markup Language version 2 (SAML v2) in a federated environment. The book includes procedures for installing, deploying and configuring a number of host machines and applications. This chapter contains the following introductory information on the deployment.

1.1 Key Features of Deployment

Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0 is designed to highlight the following key features:

- All instances of OpenSSO Enterprise are deployed behind a load balancer for high-availability.
- Instances of OpenSSO Enterprise acting as an identity provider are configured to work with instances of Sun Java™ System Directory Server configured as the user data store.
- XML Signing is enabled for all SAML v2 protocols.
- The SAML v2 URL end points are exposed through load balancers with SSL termination and regeneration configuration.
- A web policy agent and a J2EE policy agent are deployed in front of the service provider instances of OpenSSO Enterprise; the policy agents work in single sign-on mode only.

1.2 Deployment Architecture and Components

In a deployment configured for communication using SAML v2 a service provider and an identity provider must be created within a *circle of trust*. The circle of trust enables business providers to easily conduct cross-network transactions for an individual while protecting the individual's identity. The following sections contain information on the architecture of the two providers in this deployment.

- "1.2.1 Identity Provider Deployment" on page 22
- "1.2.2 Service Provider Deployment" on page 24

1.2.1 Identity Provider Deployment

An identity provider specializes in providing authentication services. As the administrating service for authentication, an identity provider maintains and manages identity information. It establishes trust with a service provider in order to exchange user credentials, enabling single sign-on between the providers. Authentication by an identity provider is honored by all service providers with whom the identity provider is partnered. The identity provider domain is `idp-example.com`. The following image illustrates the identity provider architecture in this deployment.

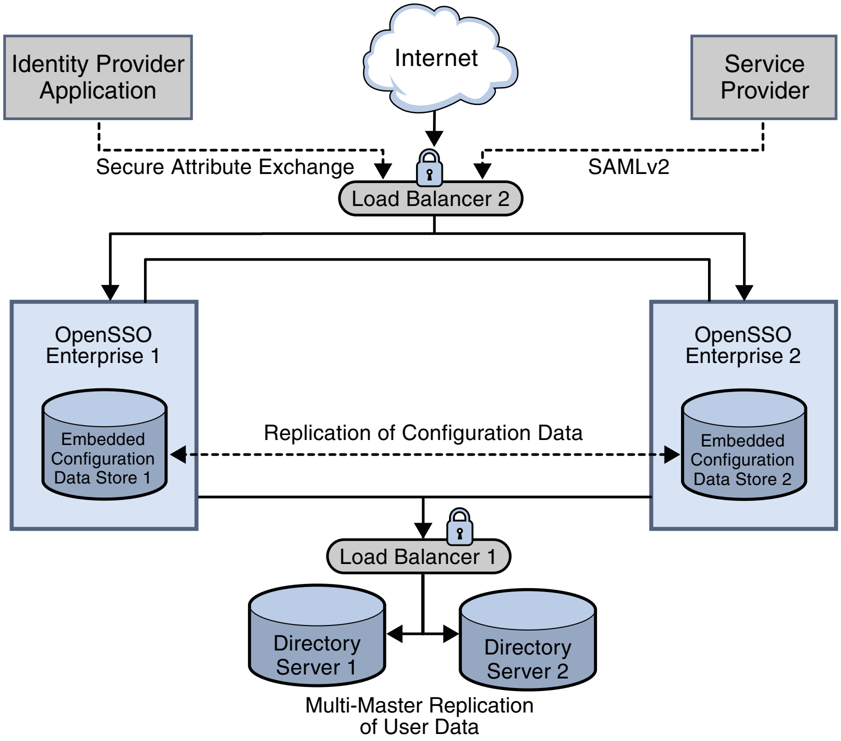
Identity Provider Application
Internet
Service Provider
Secure Attribute Exchange
SAMLv2
Load Balancer 2
OpenSSO Enterprise 1
Embedded Configuration Data Store 1
Replication of Configuration Data
OpenSSO Enterprise 2
Embedded Configuration Data Store 2
Load Balancer 1
Directory Server 1
Directory Server 2
Multi-Master Replication of User Data

FIGURE 1–1 Identity Provider Deployment Architecture

The identity provider domain in this deployment is `idp-example.com`. The identity provider application represents a legacy system which relies on OpenSSO Enterprise to act as a secure gateway through which identity information can be transferred to another application in a different domain. This functionality is provided by the Secure Attribute Exchange feature of OpenSSO Enterprise which uses SAML v2 without having to deal with federation protocol and processing.

The following list of components will be installed and configured on the identity provider side using the procedures documented in Part II.

Sun OpenSSO Enterprise

Two instances of OpenSSO Enterprise provide the core functionality. Each instance is created with a configuration data store. Configuration data includes information about services, administrative users, realms, policies, and more. Two instances of **Sun Java System Application Server** are installed on the OpenSSO Enterprise host machines into which the OpenSSO Enterprise WAR is then deployed.

Note – User data is accessed through a single load balancer deployed in front of two instances of Sun Java System Directory Server.

Sun Java System Directory Server

Two instances of Directory Server provide storage for user entries that will be created for testing this deployment. Both instances of Directory Server are masters that engage in multi-master replication, providing high availability to the OpenSSO Enterprise layer.

Note – The command line is used for all Directory Server configurations in this guide.

Load Balancers

The load balancer hardware and software used for this deployment is BIG-IP® manufactured by F5 Networks. They are deployed as follows:

OpenSSO Enterprise Load Balancer. This load balancer exposes the web-based OpenSSO Enterprise console to internal administrators. Alternatively, internal administrators can bypass this load balancer and log in directly.

Directory Server Load Balancer. The load balancer in front of the Directory Server instances provide round-robin load balancing and a single virtual Directory Server host name. It detects individual Directory Server failures and recoveries, taking failed servers off the load balancer list.

1.2.2 Service Provider Deployment

A service provider offers web-based services to an identity. This broad category can include portals, retailers, transportation providers, financial institutions, entertainment companies, libraries, universities, governmental agencies, and other organizations that consume identity information for purposes of access. The service provider domain is `sp-example.com`. The following image illustrates the service provider architecture in this deployment.

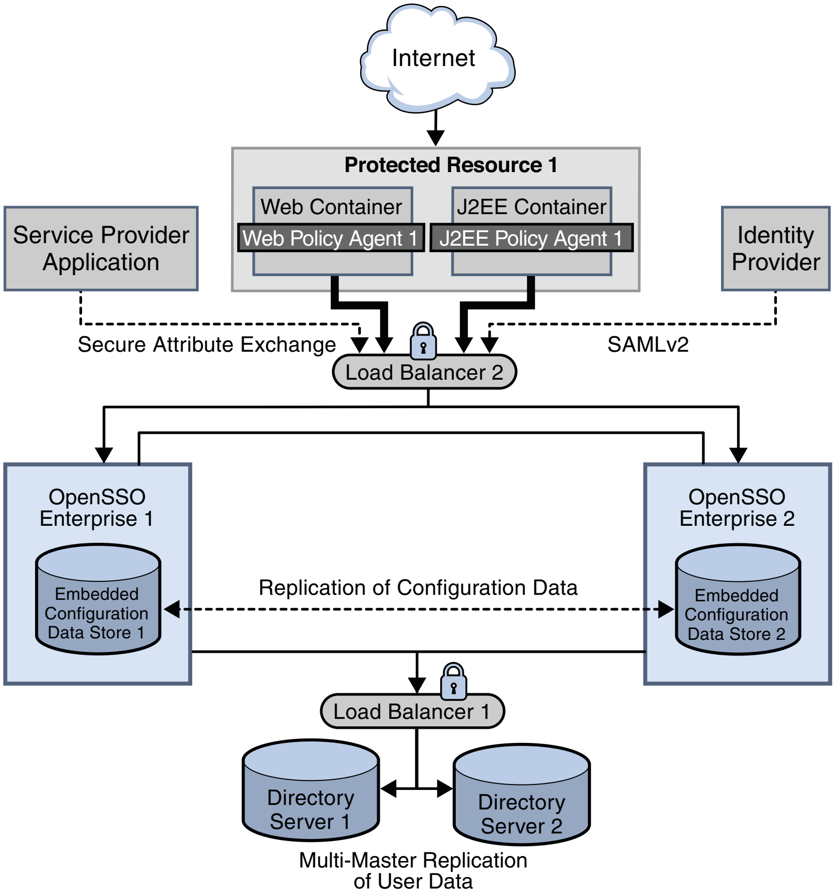

FIGURE 1–2 Service Provider Deployment Architecture

The service provider domain in this deployment is `sp-example.com`. The service provider application represents a legacy system which relies on OpenSSO Enterprise to act as a secure gateway through which identity information can be received from the identity provider. This functionality is provided by the Secure Attribute Exchange feature of OpenSSO Enterprise which uses SAML v2 without having to deal with federation protocol and processing.

The following list of components will be installed and configured using the procedures documented in Part III.

Sun OpenSSO Enterprise

Two instances of OpenSSO Enterprise provide the core functionality. Each instance is created with a configuration data store. Configuration data includes information about services, administrative users, realms, policies, and more. Two instances of **Sun Java System Application Server** are installed on the OpenSSO Enterprise host machines into which the OpenSSO Enterprise WAR is then deployed.

Note – User data is accessed through a single load balancer deployed in front of two instances of Sun Java System Directory Server.

Sun Java System Directory Server

Two instances of Directory Server provide storage for user entries that will be created for testing this deployment. Both instances of Directory Server are masters that engage in multi-master replication, providing high availability to the OpenSSO Enterprise layer.

Note – The command line is used for all Directory Server configurations in this guide.

Load Balancers

The load balancer hardware and software used for this deployment is BIG-IP® manufactured by F5 Networks. They are deployed as follows:

OpenSSO Enterprise Load Balancer. This load balancer exposes the web-based OpenSSO Enterprise console to internal administrators. Alternatively, internal administrators can bypass this load balancer and log in directly.

Directory Server Load Balancer. The load balancer in front of the Directory Server instances provides round-robin load balancing and a single virtual Directory Server host name. It detects individual Directory Server failures and recoveries, taking failed servers off the load balancer list.

Sun OpenSSO Enterprise Policy Agents

Policy agents are used to restrict access to hosted content or applications. The policy agents intercept HTTP requests from external users and redirect the request to OpenSSO Enterprise for authentication. Web policy agents protect any resources under the `doc` root of the web container. J2EE policy agents protect a variety of hosted J2EE applications; in this deployment, `agentsample` is used. The agents communicate with the OpenSSO Enterprise instances through the configured load balancer.

Protected Resource Host Machine

The protected resource host machine contains the content to which access is restricted. Towards this end, **BEA WebLogic Server**, **Sun Java System Web Server**, and the respective J2EE and web policy agents will be installed. A sample Java Server Page included with OpenSSO Enterprise will be used to emulate a legacy application for purposes of demonstrating Secure Attribute Exchange using SAML v2. The protected resource host machine will be used in Chapter 14, “Testing Attribute Mapping”

1.3 Sequential Component Interactions

The following image describes the interactions between the various components during the attribute mapping use case. See Chapter 14, "Testing Attribute Mapping."

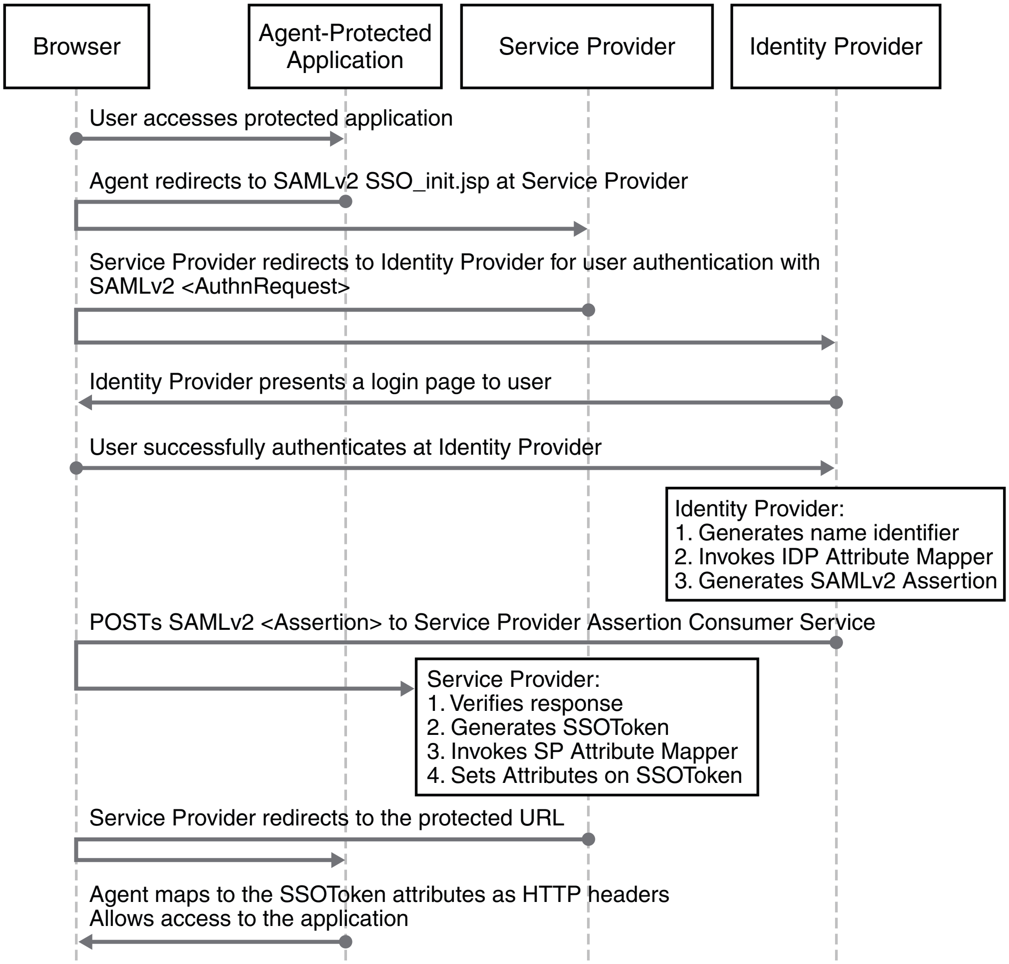
Browser
Agent-Protected Application
Service Provider
Identity Provider
User accesses protected application
Agent redirects to SAMLv2 SSO_init.jsp at Service Provider
Service Provider redirects to Identity Provider for user authentication with SAMLv2 <AuthnRequest>
Identity Provider presents a login page to user
User successfully authenticates at Identity Provider
Identity Provider:
1. Generates name identifier
2. Invokes IDP Attribute Mapper
3. Generates SAMLv2 Assertion
POSTs SAMLv2 <Assertion> to Service Provider Assertion Consumer Service
Service Provider:
1. Verifies response
2. Generates SSOToken
3. Invokes SP Attribute Mapper
4. Sets Attributes on SSOToken
Service Provider redirects to the protected URL
Agent maps to the SSOToken attributes as HTTP headers
Allows access to the application

FIGURE 1–3 Process Flow

The following image describes the interactions between the various components during the single logout use case. See Chapter 12, "Testing the SAML v2 Profiles."

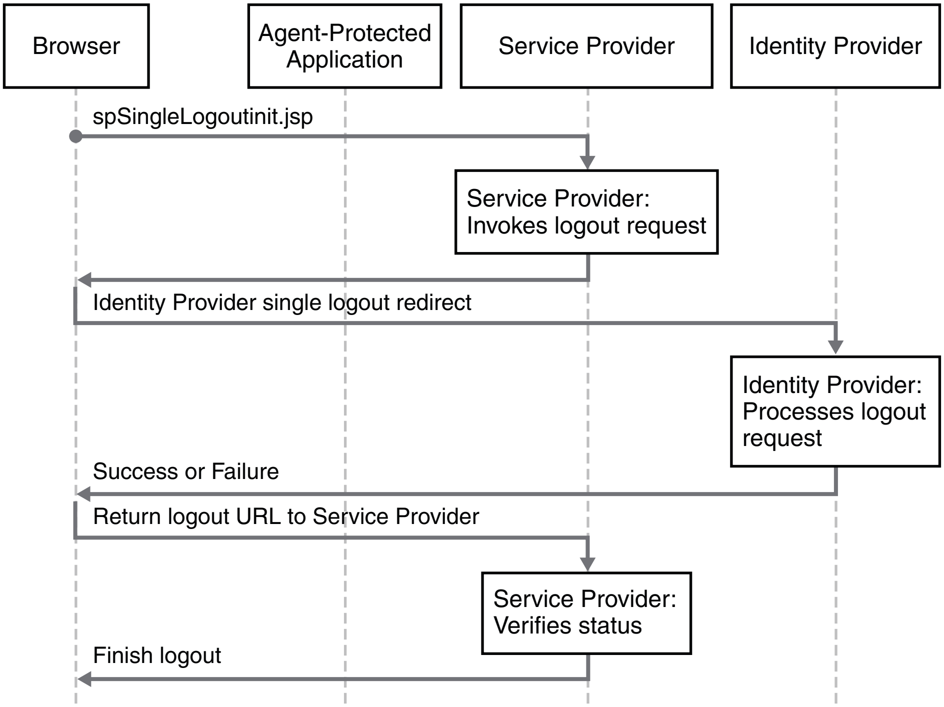
Browser
Agent-Protected Application
Service Provider
Identity Provider
spSingleLogoutinit.jsp
Service Provider: Invokes logout request
Identity Provider single logout redirect
Identity Provider: Processes logout request
Success or Failure
Return logout URL to Service Provider
Service Provider: Verifies status
Finish logout

FIGURE 1–4 Process Flow 2

CHAPTER 2

Technical Overview

This chapter contains technical information regarding the machines, software, and other components used in this deployment example. It contains the following sections:

2.1 Host Machines

The following table lists the attributes of the host machines used for this deployment example.

TABLE 2–1 Host Machines and Operating Systems

Host Machine	Architecture	Operating System
ds1.idp-example.com	x86	Solaris 10
ds2.idp-example.com	x86	Solaris 10
osso1.idp-example.com	SPARC	Solaris 10
osso2.idp-example.com	SPARC	Solaris 10
lb1.idp-example.com	SPARC	Solaris 10
lb2.idp-example.com	SPARC	Solaris 10
ds1.sp-example.com	SPARC	Solaris 10
ds2.sp-example.com	SPARC	Solaris 10
osso1.sp-example.com	SPARC	Solaris 10

TABLE 2–1 Host Machines and Operating Systems *(Continued)*

Host Machine	Architecture	Operating System
`osso2.sp-example.com`	SPARC	Solaris 10
`lb3.sp-example.com`	SPARC	Solaris 10
`lb4.sp-example.com`	SPARC	Solaris 10
`pr1.sp-example.com`	SPARC	Solaris 10

2.2 Software

The following table lists the software used in this deployment example.

TABLE 2–2 Software and Download Locations

Product	Version	Download Location
Sun OpenSSO Enterprise	8.0	`http://www.sun.com/download/`
Sun Java System Web Server	7.0 Update 3	`http://www.sun.com/download/`
Sun Java System Directory Server Enterprise Edition	6.3 Update 3	`http://www.sun.com/download/`
BEA Weblogic Server	10	`http://www.bea.com`
Web Policy Agent (for Sun Java System Web Server)	3.0	`http://www.sun.com/download/`
J2EE Policy Agent (for BEA Weblogic Server)	3.0	`http://www.sun.com/download/`
Java (for OpenSSO Enterprise and policy agents)	1.5.0_09	`http://www.java.com/en/`
BIG-IP Load Balancer		`http://www.f5.com`

2.3 Main Service URLs

The following sections summarize the main service URLs for the components used in this deployment example. For detailed configuration information, see Part V.

- "2.3.1 Identity Provider Main Service URLs" on page 31
- "2.3.2 Service Provider Main Service URLs" on page 32

2.3.1 Identity Provider Main Service URLs

The following tables summarize the main service URLs for the identity provider components.

TABLE 2–3 Identity Provider Components and Main Service URLs

Components	Main Service URL
Directory Server Host Machines and Load Balancer	
Directory Server 1	`ds1.idp-example.com:1736` (for monitor node) `ldaps://ds1.idp-example.com:1736` (for user data)
Directory Server 2	`ds2.idp-example.com:1736` (for monitor node) `ldaps://ds2.idp-example.com:1736` (for user data)
Load Balancer 1	`ldaps://lb1.idp-example.com:489` (for Directory Server access)
OpenSSO Enterprise Host Machines and Load Balancer	
Application Server 1	**Default Domain** `http://osso1.idp-example.com:4848` (for console) `http://osso1.idp-example.com:8080` (for HTTP) `https://osso1.idp-example.com:8181` (for HTTPS) **Non—Root User Domain** `http://osso1.idp-example.com:8989` (for console) `http://osso1.idp-example.com:1080` (for HTTP) `https://osso1.idp-example.com:1081` (for HTTPS)
OpenSSO Enterprise 1	`https://osso1.idp-example.com:1081/opensso/console`
Application Server 2	**Default Domain** `http://osso2.idp-example.com:4848` (for console) `http://osso2.idp-example.com:8080` (for HTTP) `https://osso2.idp-example.com:8181` (for HTTPS)

TABLE 2–3 Identity Provider Components and Main Service URLs *(Continued)*

Components	Main Service URL
	Non—Root User Domain `http://osso2.idp-example.com:8989` (for console) `http://osso2.idp-example.com:1080` (for HTTP) `https://osso2.idp-example.com:1081` (for HTTPS)
OpenSSO Enterprise 2	`https://osso2.idp-example.com:1081/opensso/console`
Load Balancer 2	`https://lb2.idp-example.com:1081/opensso` (for OpenSSO Enterprise access) `http://lb2.idp-example.com:1082` (for virtual server proxy)

2.3.2 Service Provider Main Service URLs

The following tables summarize the main service URLs for the service provider components.

TABLE 2–4 Service Provider Components and Main Service URLs

Components	Main Service URL
Directory Server Host Machines and Load Balancers	
Directory Server 1	`ds1.sp-example.com:1736` (for monitor node) `ldaps://ds1.sp-example.com:1736` (for user data)
Directory Server 2	`ds2.sp-example.com:1736` (for monitor node) `ldaps://ds2.sp-example.com:1736` (for user data)
Load Balancer 3	`ldaps://lb3.sp-example.com:489` (for user data)
OpenSSO Enterprise Host Machines and Load Balancer	

TABLE 2–4 Service Provider Components and Main Service URLs *(Continued)*

Components	Main Service URL
Application Server 1	**Default Domain** `http://osso1.sp-example.com:4848` (for console) `http://osso1.sp-example.com:8080` (for HTTP) `https://osso1.sp-example.com:8181` (for HTTPS) **Non—Root User Domain** `http://osso1.sp-example.com:8989` (for console) `http://osso1.sp-example.com:1080` (for HTTP) `https://osso1.sp-example.com:1081` (for HTTPS)
OpenSSO Enterprise 1	`https://osso1.sp-example.com:1081/opensso/console`
Application Server 1	**Default Domain** `http://osso2.sp-example.com:4848` (for console) `http://osso2.sp-example.com:8080` (for HTTP) `https://osso2.sp-example.com:8181` (for HTTPS) **Non—Root User Domain** `http://osso2.sp-example.com:8989` (for console) `http://osso2.sp-example.com:1080` (for HTTP) `https://osso2.sp-example.com:1081` (for HTTPS)
OpenSSO Enterprise 2	`https://osso2.sp-example.com:1081/opensso/console`
Load Balancer 4	`https://lb4.sp-example.com:1081/opensso` (for OpenSSO Enterprise access) `http://lb4.sp-example.com:1082` (for virtual server proxy)
Protected Resource 1 Host Machine Web Containers and Policy Agents	

TABLE 2–4 Service Provider Components and Main Service URLs *(Continued)*

Components	Main Service URL
Web Server	`https://pr1.sp-example.com:8989` (for Sun Java System Web Server administration console) `http://pr1.sp-example.com:1080` (for Sun Java System Web Server managed instance)
Web Policy Agent	`http://pr1.sp-example.com:1080`
WebLogic Server	`http://pr1.sp-example.com:7001/console` (for BEA Weblogic administration server) `http://pr1.sp-example.com:1081` (for BEA Weblogic managed server)
J2EE Policy Agent	`http://pr1.sp-example.com:1081/agentapp`

2.4 Viewing Replicated Entries

Throughout this deployment example, we use `ldapsearch` to view replicated entries. An alternative would be to enable the Directory Server audit log and run `tail -f`. Enabling the audit log will also help to track changes and updates made during OpenSSO Enterprise configuration.

CHAPTER 3

Before You Begin

This chapter contains information you need to know before beginning the documented installation and configuration procedures. It contains the following sections:

3.1 Technical Reference

See Chapter 2, "Technical Overview," for a quick reference of host machines, port numbers, operating systems, naming conventions, and component names used in this deployment example. See Part V for more detailed information.

3.2 Setting Up the Load Balancers

The load balancer hardware and software used in this deployment environment is BIG-IP® manufactured by F5 Networks. If you are using different load balancer software, see the documentation that comes with that product for detailed settings information. This document assumes that you have already installed the required load balancers. The following identity provider sections require load-balancing hardware and software.

- "4.5 Configuring the Directory Server Load Balancer" on page 64
- "5.2 Configuring the OpenSSO Enterprise Load Balancer" on page 94

The following service provider sections require load-balancing hardware and software.

- "7.5 Configuring the Directory Server Load Balancer" on page 149
- "8.2 Configuring the OpenSSO Enterprise Load Balancer" on page 184

3.3 Obtaining Secure Socket Layer Certificates

In order to enable secure communications using the Secure Sockets Layer (SSL) protocol you need to obtain root certificates and server certificates from a certificate authority (CA). A CA root certificate proves that the particular CA issued a particular server certificate. CA root certificates are publicly available. The root certificate used in this deployment is a self-signed certificate issued by OpenSSL for testing purposes only; it is named ca.cer. You can obtain a root certificate from any commercial certificate issuer such as VeriSign, Thawte, Entrust, or GoDaddy.

The server certificates are requested from, and issued by, OpenSSL within each procedure. You should know how to request server certificates from your CA of choice before beginning this deployment. The following identity provider sections are related to requesting, installing, and importing root and server certificates.

- “To Import a Root Certificate and a Server Certificate to Directory Server 1” on page 60
- “To Import a Root Certificate and a Server Certificate to Directory Server 2” on page 62
- “To Import the Root Certificate to Directory Server Load Balancer 1” on page 64
- “To Request a Certificate for OpenSSO Enterprise Load Balancer 2” on page 96
- “To Install the Certificate Authority Root Certificate to OpenSSO Enterprise Load Balancer 2” on page 97
- “To Install the Server Certificate to OpenSSO Enterprise Load Balancer 2” on page 97
- “To Install a Root Certificate and a Server Certificate on Directory Server 1” on page 146

The following service provider sections are related to requesting, installing, and importing root and server certificates.

- “To Install a Root Certificate and a Server Certificate on Directory Server 1” on page 146
- “To Install a Root Certificate and a Server Certificate on Directory Server 2” on page 148
- “To Import the Root Certificate to the User Data Load Balancer” on page 150
- “To Request a Certificate for OpenSSO Enterprise Load Balancer 2” on page 185
- “To Install a CA Root Certificate to OpenSSO Enterprise Load Balancer 2” on page 186
- “To Install the Server Certificate to OpenSSO Enterprise Load Balancer 2” on page 187
- “To Import a Certificate Authority Root Certificate to Protected Resource 1” on page 221
- “To Import a Certificate Authority Root Certificate to Protected Resource 1” on page 239

3.4 Resolving Host Names

There are many ways to resolve the host names used in this deployment. You may use a DNS naming service, or you can map IP addresses to host names in the local host file on all UNIX® hosts. The same entries must also be added to equivalent files on Windows hosts, and on client machines where browsers are used. For example:

```
1xx.xx.xx.x1        ds1     ds1.sp-example.com
1xx.xx.xx.x2        ds2     ds2.sp-example.com
```

```
1xx.xx.xx.x3        osso1         osso1.idp-example.com
1xx.xx.xx.x4        osso2         osso2.idp-example.com
```

3.5 Known Issues and Limitations

See Appendix G, “Known Issues and Limitations,” for descriptions of problems you may encounter when implementing the deployment example. This list will be updated as new information becomes available.

Although the instructions and procedures documented in this book incorporate many *best practices*, and may be suitable in many different scenarios, this is not the only way to achieve the same results. If you plan to deviate from the task sequence or details described, you should refer to the relevant product documentation for information on differences in platforms, software versions or other requirement constraints.

PART II

Building the Identity Provider Environment

This second part of *Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0* provides the instructions for installing and configuring OpenSSO Enterprise, Sun Java System Directory Server, applicable web containers and policy agents to function as the identity provider. Best results will be obtained by executing the tasks in the exact sequence in which they are presented. This part contains the following chapters:

- Chapter 4, "Installing Sun Java System Directory Server and Creating Instances for User Data"
- Chapter 5, "Deploying and Configuring OpenSSO Enterprise"
- Chapter 6, "Configuring OpenSSO Enterprise Realms for User Authentication"

Caution – If deviating from the task sequence or details described, refer to the relevant product documentation for information or necessary requirements.

CHAPTER 4

Installing Sun Java System Directory Server and Creating Instances for User Data

This chapter contains instructions for installing Sun Java™ System Directory Server and creating the instances in which user data will be stored. Additionally, the procedure for enabling multi-master replication between the two instances and the procedure for configuring the user data load balancer are included. This chapter contains the following sections:

- "4.1 Installing and Configuring Directory Server 1 and Directory Server 2" on page 41
- "4.2 Enabling Multi-Master Replication of the User Data Instances" on page 51
- "4.3 Modifying the Directory Server Schema" on page 59
- "4.4 Enabling Secure Communication for the Directory Server User Data Instances" on page 60
- "4.5 Configuring the Directory Server Load Balancer" on page 64

Note – If you have an existing user data store, you can go directly to the instructions in Chapter 5, "Deploying and Configuring OpenSSO Enterprise."

4.1 Installing and Configuring Directory Server 1 and Directory Server 2

This section contains the instructions for installing Directory Server on two different host machines on the identity provider side. Post installation, create the directory instances named `idp-users` in which the user data will be stored. Use the following list of procedures as a checklist for completing the task.

1. "To Download the Directory Server Bits and Required Patches to the Host Machines" on page 42
2. "To Patch the Directory Server Host Machines" on page 44
3. "To Install Directory Server 1" on page 45
4. "To Create a User Data Instance on Directory Server 1" on page 46
5. "To Create a Base Suffix for the User Data Instance on Directory Server 1" on page 47

▼ To Download the Directory Server Bits and Required Patches to the Host Machines

Use this procedure to download the Directory Server Enterprise Edition (EE) 6.3 bits and the required system patches to both the Directory Server 1 host machine (`ds1.idp-example.com`) and the Directory Server 2 host machine (`ds2.idp-example.com`).

1 **Access** `http://www.sun.com/software/products/directory_srvr_ee/get.jsp` **from a web browser and click Download Now.**

2 **Provide the following information in the** ***Select product configuration*** **section and click View Downloads.**

Step 1: Select Component	**`Directory Server Enterprise Edition 6.x`**
Step 2: Select Version	**`6.3`**
Step 3: Select Delivery Type	**`Compress Archive (ZIP)`**
Step 4: Select Platform	Choose the platform you are using.

The Selection Results page will be displayed with links to the download sites for the Directory Server bits and required patches.

Note – The patch numbers generated for download on the Selection Results page are based on your input. Check the most recent Directory Server Enterprise Edition 6.3 Release Notes to determine if you need to install other patches based on your machine's architecture and operating system. In this deployment, the Release Notes indicate that based on the hardware and operating system being used, patch 118855, patch 127112, patch 119964, patch 125379, and patch 119255 are required.

3 **Log into the** `ds1.idp-example.com` **host machine as a root user.**

4 **Run** `patchadd` **to see if the patches are already installed.**

See the `patchadd` man page for more information.

```
# /usr/sbin/patchadd -p | grep 118855
```

No results are returned which indicates that the patch is not yet installed on the system.

```
# /usr/sbin/patchadd -p | grep 127112
```

No results are returned which indicates that the patch is not yet installed on the system.

```
# /usr/sbin/patchadd -p | grep 119964
```

No results are returned which indicates that the patch is not yet installed on the system.

```
# /usr/sbin/patchadd -p | grep 125379
```

No results are returned which indicates that the patch is not yet installed on the system.

```
# /usr/sbin/patchadd -p | grep 119255
```

No results are returned which indicates that the patch is not yet installed on the system.

Note – If these patches are already installed on your machine, proceed to step 7.

5 **Make a directory for the patch downloads and change into it.**

```
# mkdir /export/patches
# cd /export/patches
```

6 **Download the patches.**

You can click on the patch links from the Selection Results page or search for patches directly at `http://sunsolve.sun.com`. If searching directly, navigate to the PatchFinder page and enter the patch number. For each patch you are downloading, click the HTTP link beside the heading *Download Signed Patch (xxx bytes)*.

Note – Signed patches are downloaded as JAR files. Unsigned patches are downloaded as ZIP files. In this step, ZIP files are downloaded.

7 **Make a directory for the Directory Server download and change into it.**

```
# mkdir /export/DS63
# cd /export/DS63
```

8 **Download the *Base Full Install* of Directory Server EE 6.3 — Zip Distribution, Multi-Language, (DS/DPS/DE/ISW/DSRK) bits.**

Note – No Directory Server Administration Console is installed with these bits. This deployment example uses the command line to configure the software.

9 **Log out of the `ds1.idp-example.com` host machine.**

10 **Repeat this same procedure on the `ds2.idp-example.com` host machine.**

▼ To Patch the Directory Server Host Machines

If necessary, use this procedure to patch both the `ds1.idp-example.com` host machine and the `ds2.idp-example.com` host machine.

1 **Log in to the `ds1.idp-example.com` host machine as a root user.**

2 **Change into the directory that contains the downloaded patch files.**

```
# cd /export/patches
```

3 **Unzip the patch files.**

```
# unzip 118855.zip
# unzip 127112.zip
# unzip 119964.zip
# unzip 125379.zip
# unzip 119255.zip
```

4 **Install the patches.**

```
# /usr/sbin/patchadd /export/patches/118855
# /usr/sbin/patchadd /export/patches/127112
# /usr/sbin/patchadd /export/patches/119964
# /usr/sbin/patchadd /export/patches/125379
# /usr/sbin/patchadd /export/patches/119255
```

Tip – You can use the `-M` option to install all patches at once. See the `patchadd` man page for more information.

---

5 **Reboot your machine, if requested.**

6 **After installation is complete, verify that each patch was added successfully.**

```
# /usr/sbin/patchadd -p | grep 118855
```

A series of patch numbers are displayed, and the patch 118855 is present.

```
# /usr/sbin/patchadd -p | grep 127112
```

A series of patch numbers are displayed, and the patch 127112 is present.

```
# /usr/sbin/patchadd -p | grep 119964
```

A series of patch numbers are displayed, and the patch 119964 is present.

```
# /usr/sbin/patchadd -p | grep 125379
```

A series of patch numbers are displayed, and the patch 125379 is present.

```
# /usr/sbin/patchadd -p | grep 119255
```

A series of patch numbers are displayed, and the patch 119255 is present.

7 **Log out of the `ds1.idp-example.com` host machine.**

8 **Repeat this same procedure on the `ds2.idp-example.com` host machine.**

▼ To Install Directory Server 1

Before You Begin This procedures assumes "To Download the Directory Server Bits and Required Patches to the Host Machines" on page 42 and "To Patch the Directory Server Host Machines" on page 44 have been completed.

1 **Log in to the `ds1.idp-example.com` host machine as a root user.**

2 **(Optional) Resolve the following issues, if necessary.**

- The `LD_LIBRARY_PATH` environment variable should *not* be set to the default setting. Change the value to *empty* as in the following example:

  ```
  # setenv LD_LIBRARY_PATH
  ```

- The `JAVA_HOME` environment variable should be set appropriately for your system architecture as in the following example:

  ```
  # setenv JAVA_HOME /usr/jdk/jdk1.5.0_09
  ```

3 **Unzip the Directory Server ZIP file.**

```
# cd /export/DS63
# ls

DSEE.6.1.Solaris10-X86_AMD64-full.tar.gz

# gunzip DSEE.6.3.Solaris10-X86_AMD64-full.tar.gz
```

4 **Untar the resulting `.tar` file.**

```
# tar xvf DSEE.6.1.Solaris10-X86_AMD64-full.tar
```

The `DSEE_ZIP_Distribution` directory is the result of the decompression.

5 **Change into** `DSEE_ZIP_Distribution` **and run** `dsee_deploy install` **to install Directory Server.**

```
# cd DSEE_ZIP_Distribution
# ./dsee_deploy install -i /var/opt/mps/serverroot
```

The Licensing Agreement is displayed. At each `Type return to continue` prompt, press Return to continue.

6 **When** `Do you accept the license terms?` **is displayed, enter** `yes` **to continue.**

Once you accept the license terms, the Directory Server binaries will be installed in the `/var/opt/mps/serverroot/ds6` directory.

▼ To Create a User Data Instance on Directory Server 1

Use this procedure to create a Directory Server instance named `idp-users` for storing user data. The instance uses port 1489 for LDAP and port 1736 for LDAPS.

Before You Begin This procedure assumes you have just completed "To Install Directory Server 1" on page 45 and are still logged into the `ds1.idp-example.com` host machine as a root user.

1 **Change to the** `bin` **directory.**

```
# cd /var/opt/mps/serverroot/ds6/bin
```

2 **Run** `dsadm create` **to create a user data instance called** `idp-users`**.**

```
# ./dsadm create -p 1489 -P 1736 /var/opt/mps/idp-users

Choose the Directory Manager password: dsmanager

Confirm the Directory Manager password: dsmanager

use 'dsadm start /var/opt/mps/idp-users' to start the instance
```

3 **Run** `dsadm start` **to start the instance.**

```
# ./dsadm start /var/opt/mps/idp-users

Server started: pid=5810
```

4 **Run** `netstat` **to verify that the new instance is up and running on both ports.**

```
# netstat -an | grep 1736

.1736        *.*        0        0  65536        0 LISTEN
.1736        *.*        0        0  65536        0 LISTEN
```

```
# netstat -an | grep 1489

.1489        *.*        0        0  65536        0 LISTEN
.1489        *.*        0        0  65536        0 LISTEN
```

5 **Run `ldapsearch` to verify that you can read the root Directory Server entry of the new instance.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds1.idp-example.com
-p 1489 -b "" -s base "(objectclass=*)"

version: 1
dn:
objectClass: top
...
supportedLDAPVersion: 3
vendorname: Sun Microsystems, Inc.
vendorVersion: Sun-Java(tm)-System-Directory/6.3
...
```

▼ To Create a Base Suffix for the User Data Instance on Directory Server 1

Use this procedure to create the base suffix in which the user entries will be stored.

Before You Begin This procedure assumes you have just completed "To Create a User Data Instance on Directory Server 1" on page 46 and are still logged into the `ds1.idp-example.com` host machine as a root user.

1 **Run `dsconf create-suffix` to create a base suffix.**

```
# ./dsconf create-suffix -p 1489 -B dbExample
-L /var/opt/mps/idp-users/db/exampleDS dc=company,dc=com
```

2 **Provide the appropriate information when prompted.**

```
Certificate "CN=ds1, CN=1736, CN=directory Server, O=Sun Microsystems"
presented by the server is not trusted.

Type "Y" to accept, "y" to accept just once, "n" to refuse, "d" for more details: Y

Enter "cn=Directory Manager" password: dsmanager
```

Tip – When you enter an uppercase **Y**, you are not asked for the certificate again in the next steps.

3 **Run `dsconf list-suffixes` to verify that the base suffix was successfully created.**

```
# ./dsconf list-suffixes -p 1489

Enter "cn=Directory Manager" password: dsmanager

dc=company,dc=com
```

If the base suffix was successfully created, `dc=company,dc=com` is returned. You can also see `idp-users` in a command line list of directory instances.

```
# cd /var/opt/mps
# ls

idp-users serverroot
```

4 **Log out of the `ds1.idp-example.com` host machine.**

▼ To Install Directory Server 2

Before You Begin This procedures assumes "To Download the Directory Server Bits and Required Patches to the Host Machines" on page 42 and "To Patch the Directory Server Host Machines" on page 44 have been completed.

1 **Log in to the `ds2.idp-example.com` host machine as a root user.**

2 **(Optional) Resolve the following issues, if necessary.**

- The `LD_LIBRARY_PATH` environment variable should *not* be set to the default setting. Change the value to empty as in the following example:

  ```
  # setenv LD_LIBRARY_PATH
  ```

- The `JAVA_HOME` environment variable should be set appropriately for your system architecture as in the following example:

  ```
  # setenv JAVA_HOME /usr/jdk/jdk1.5.0_09
  ```

3 **Unzip the Directory Server ZIP file.**

```
# cd /export/DS63
# ls
```

```
DSEE.6.3.Solaris10-X86_AMD64-full.tar.gz

# gunzip DSEE.6.3.Solaris10-X86_AMD64-full.tar.gz
```

4 **Untar the resulting `.tar` file.**

```
# tar xvf DSEE.6.3.Solaris10-X86_AMD64-full.tar
```

The `DSEE_ZIP_Distribution` directory is the result of the decompression.

5 **Change into `DSEE_ZIP_Distribution` and run `dsee_deploy install` to install Directory Server.**

```
# cd DSEE_ZIP_Distribution
# ./dsee_deploy install -i /var/opt/mps/serverroot
```

The Licensing Agreement is displayed. At each `Type return to continue` prompt, press Return to continue.

6 **When `Do you accept the license terms?` is displayed, enter yes to continue.**

Once you accept the license terms, the Directory Server binaries will be installed in the `/var/opt/mps/serverroot/ds6` directory.

▼ To Create a User Data Instance on Directory Server 2

Use this procedure to create a Directory Server instance named `idp-users` for storing user data. The instance uses port 1489 for LDAP and port 1736 for LDAPS.

Before You Begin This procedure assumes you have just completed "To Install Directory Server 2" on page 48 and are still logged into the `ds2.idp-example.com` host machine as a root user.

1 **Change to the `bin` directory.**

```
# cd /var/opt/mps/serverroot/ds6/bin
```

2 **Run `dsadm create` to create a user data instance called `idp-users`.**

```
# ./dsadm create -p 1489 -P 1736 /var/opt/mps/idp-users

Choose the Directory Manager password: dsmanager

Confirm the Directory Manager password: dsmanager

use 'dsadm start /var/opt/mps/idp-users' to start the instance
```

3 **Run `dsadm start` to start the instance.**

```
# ./dsadm start /var/opt/mps/idp-users

Server started: pid=5810
```

4 **Run `netstat` to verify that the new instance is up and running on both ports.**

```
# netstat -an | grep 1736

.1736        *.*        0        0  65536        0 LISTEN
.1736        *.*        0        0  65536        0 LISTEN

# netstat -an | grep 1489

.1489        *.*        0        0  65536        0 LISTEN
.1489        *.*        0        0  65536        0 LISTEN
```

5 **Run `ldapsearch` to verify that you can read the root Directory Server entry of the new instance.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds2.idp-example.com
-p 1489 -b "" -s base "(objectclass=*)"

version: 1
dn:
objectClass: top
...
supportedLDAPVersion: 3
vendorname: Sun Microsystems, Inc.
vendorVersion: Sun-Java(tm)-System-Directory/6.3
...
```

▼ To Create a Base Suffix for the User Data Instance on Directory Server 2

Use this procedure to create the base suffix in which the user entries will be stored.

Before You Begin This procedure assumes you have just completed "To Create a User Data Instance on Directory Server 2" on page 49 and are still logged into the `ds2.idp-example.com` host machine as a root user.

1 **Run `dsconf create-suffix` to create a base suffix.**

```
# ./dsconf create-suffix -p 1489 -B dbExample
-L /var/opt/mps/idp-users/db/exampleDS dc=company,dc=com
```

2 **Provide the appropriate information when prompted.**

```
Certificate "CN=ds2, CN=1736, CN=directory Server, O=Sun Microsystems"
presented by the server is not trusted.

Type "Y" to accept, "y" to accept just once, "n" to refuse, "d" for more details: Y

Enter "cn=Directory Manager" password: dsmanager
```

Tip – When you enter an uppercase **Y**, you are not asked for the certificate again in the next steps.

---

3 **Run** `dsconf list-suffixes` **to verify that the base suffix was successfully created.**

```
# ./dsconf list-suffixes -p 1489

Enter "cn=Directory Manager" password: dsmanager

dc=company,dc=com
```

If the base suffix was successfully created, `dc=company,dc=com` is returned. You can also see `idp-users` in a command line list of directory instances.

```
# cd /var/opt/mps
# ls

idp-users serverroot
```

4 **Log out of the** `ds2.idp-example.com` **host machine.**

4.2 Enabling Multi-Master Replication of the User Data Instances

This section contains the instructions to enable multi-master replication (MMR) between two Directory Server instances, each configured as a *master*. This includes creating replication agreements between the masters and initializing the second directory master with the data and schema from the first directory master. The previously created `idp1-user` and `idp2-user` instances will serve as the two master instances. Use the following list of procedures as a checklist for completing the task.

1. "To Enable Multi-Master Replication for User Data Instance on Directory Server 1" on page 138
2. "To Enable Multi-Master Replication for User Data Instance on Directory Server 2" on page 139

3. "To Change the Default Replication Manager Password for Each User Data Instance" on page 140
4. "To Create Replication Agreements for Each User Data Instance" on page 141
5. "To Initialize the Replication Agreements" on page 142
6. "To Verify Successful User Data Replication" on page 143

▼ To Enable Multi-Master Replication for the User Data Instance on Directory Server 1

1 **Log in to the `ds1.idp-example.com` host machine as a root user.**

2 **(Optional) Run `dsconf list-suffixes` to verify that the user data instance is not already enabled for replication.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf list-suffixes -p 1489 -v

Enter "cn=Directory Manager" password: dsmanager
...
dc=company,dc=com     1        not-replicated        N/A        N/A        29    0

The "list-suffixes" operation succeeded on "ds1.idp-example.com:1489"
```

The base suffix of the user data instance is not replicated.

3 **Run `dsconf enable-repl` to enable replication of the user data instance.**

```
# ./dsconf enable-repl -h ds1.idp-example.com
-p 1489 -d 11 master dc=company,dc=com

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf create-repl-agmt" to create replication agreements on
"dc=company,dc=com".
```

The `-d` option takes as input a randomly chosen identifier to represent the Directory Server 1 user data instance; in this case, `11 master` indicates that the user data instance is a master and not a replica. The base suffix is specified as `dc=company,dc=com`.

4 **Run `dsconf list-suffixes` again to verify that the instance is now enabled for replication.**

```
# ./dsconf list-suffixes -p 1489 -v

Enter "cn=Directory Manager" password: dsmanager
...
dc=company,dc=com     1        master(11)        N/A        N/A        29    0
```

```
The "list-suffixes" operation succeeded on
"ds1.idp-example.com:1489"
```

The base suffix of the instance is `master(11)` indicating that the master was successfully enabled.

5 **Log out of the `ds1.idp-example.com` host machine.**

To Enable Multi-Master Replication for the User Data Instance on Directory Server 2

1 **Log in to the `ds2.idp-example.com` host machine as a root user.**

2 **(Optional) Run `dsconf list-suffixes` to verify that the user data instance is not already enabled for replication.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf list-suffixes -p 1489 -v

Enter "cn=Directory Manager" password: dsmanager
...
dc=company,dc=com     1        not-replicated        N/A        N/A        29    0

The "list-suffixes" operation succeeded on
"ds2.idp-example.com:1489"
```

The base suffix of the user data instance is not replicated.

3 **Run `dsconf enable-repl` to enable replication of the user data instance.**

```
# ./dsconf enable-repl -h ds2.idp-example.com -p 1489
-d 22 master dc=company,dc=com

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf create-repl-agmt" to create replication agreements on
"dc=company,dc=com".
```

The `-d` option takes as input a randomly chosen identifier to represent the Directory Server 2 user data instance; in this case, 22 `master` indicates that the user data instance is a master and not a replica. The base suffix is specified as `dc=company,dc=com`.

4 **Run `dsconf list-suffixes` again to verify that the instance is now enabled for replication.**

```
# ./dsconf list-suffixes -p 1489 -v
```

```
Enter "cn=Directory Manager" password: dsmanager
...
dc=company,dc=com     1        master(22)        N/A        N/A        29        0

The "list-suffixes" operation succeeded on
"ds2.idp-example.com:1489"
```

The base suffix of the instance is `master(22)` indicating that the master was successfully enabled.

5 **Log out of the `ds2.idp-example.com` host machine.**

▼ To Change the Default Replication Manager Password for Each User Data Instance

The *replication manager* is the user that data suppliers use to bind to the consumer server when sending replication updates. (In MMR the consumer server refers to whichever master happens to be the consumer for a particular operation.) It is recommended to change the default password created during the process of enabling replication.

1 **Log in to the `ds1.idp-example.com` host machine as a root user.**

2 **Create a temporary file that contains the new replication manager password.**

This file will be read once, and the password stored for future use.

```
# cd /var/opt/mps/serverroot/ds6/bin
# echo replmanager > pwd.txt
```

3 **Verify that the file was successfully created.**

```
# cat pwd.txt

replmanager
```

4 **Run `dsconf set-server-prop` to set the replication manager password using `pwd.txt` as input.**

```
# ./dsconf set-server-prop -h ds1.idp-example.com
-p 1489 def-repl-manager-pwd-file:pwd.txt

Enter "cn=Directory Manager" password: dsmanager
```

5 **Remove the `pwd.txt` file.**

6 **Log out of the `ds1.idp-example.com` host machine.**

7 **Log in to the `ds2.idp-example.com` host machine as a root user.**

8 **Create a temporary file that contains the new replication manager password.**

This file will be read once, and the password stored for future use.

```
# cd /var/opt/mps/serverroot/ds6/bin
# echo replmanager > pwd.txt
```

9 **Verify that the file was successfully created.**

```
# cat pwd.txt

replmanager
```

10 **Run `dsconf set-server-prop` to set the replication manager password using `pwd.txt` as input.**

```
# ./dsconf set-server-prop -h ds2.idp-example.com
-p 1489 def-repl-manager-pwd-file:pwd.txt

Enter "cn=Directory Manager" password: dsmanager
```

11 **Remove the `pwd.txt` file.**

12 **Log out of the `ds2.idp-example.com` host machine.**

▼ To Create Replication Agreements for Each User Data Instance

A *replication agreement* is a set of parameters on a supplier that controls how updates are sent to a given consumer. In this deployment, the agreement simply makes the user data instances aware of each other.

1 **Log in to the `ds1.idp-example.com` host machine as a root user.**

2 **Run `dsconf create-repl-agmt` to create the replication agreement.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf create-repl-agmt -h ds1.idp-example.com
 -p 1489 dc=company,dc=com ds2.idp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf init-repl-dest dc=company,dc=com
ds2.idp-example.com:1489" to start replication of
"dc=company,dc=com" data.
```

3 **Run** `dsconf list-repl-agmts` **to verify that the replication agreement was successfully created.**

```
# ./dsconf list-repl-agmts -p 1489

Enter "cn=Directory Manager" password: dsmanager

dc=company,dc=com ds2.idp-example.com:1489
```

This response indicates that the Directory Server 1 base suffix will be replicated to Directory Server 2.

4 **Log out of the** `ds1.idp-example.com` **host machine.**

5 **Log in to the** `ds2.idp-example.com` **host machine as a root user.**

6 **Run** `dsconf create-repl-agmt` **to create the replication agreement.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf create-repl-agmt -h ds2.idp-example.com
-p 1489 dc=company,dc=com ds1.idp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf init-repl-dest dc=company,dc=com ds1.idp-example.com:1489"
to start replication of "dc=company,dc=com" data.
```

7 **Run** `dsconf list-repl-agmts` **to verify that the replication agreement was successfully created.**

```
# ./dsconf list-repl-agmts -p 1489

Enter "cn=Directory Manager" password: dsmanager

dc=company,dc=com ds1.idp-example.com:1489
```

This response indicates that the Directory Server 2 base suffix will be replicated to Directory Server 1.

8 **Log out of the** `ds2.idp-example.com` **host machine.**

▼ To Initialize the Replication Agreements

Use this procedure to initialize the user data instance on Directory Server 1. The previously created agreements will allow the data to replicate on Directory Server 2.

Note – Initialization is **not** required on both instances when configuring for MMR.

1 **Log in to the `ds1.idp-example.com` host machine as a root user.**

2 **Run `dsconf show-repl-agmt-status` to verify that the replication agreements are not yet initialized.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf show-repl-agmt-status -h ds1.idp-example.com
 -p 1489 dc=company,dc=com ds2.idp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Configuration Status         : OK
Authentication Status        : OK
Initialization Status        : NOT OK

Status:                        : Dest. Not Initialized
```

3 **Run `dsconf init-repl-dest` to initialize the replication agreements.**

```
# ./dsconf init-repl-dest -h ds1.idp-example.com
 -p 1489 dc=company,dc=com ds2.idp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Started initialization of "ds2.idp-example.com:1489"; Aug 25, 2008 3:10:01 PM
Sent 2 entries.
Completed initialization of "ds1.idp-example.com:1489"; Aug 25, 2008 3:10:04 PM
```

4 **Run `dsconf show-repl-agmt-status` again to verify that the replication agreements are now initialized.**

```
# ./dsconf show-repl-agmt-status -h ds1.idp-example.com
 -p 1489 dc=company,dc=com ds2.idp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Configuration Status         : OK
Authentication Status        : OK
Initialization Status        : OK

Status:                        : Enabled
Last Update Date                :  Aug 25, 2008 3:10:08 PM
```

▼ To Verify Successful User Data Replication

Before You Begin This procedure assumes you have just completed "To Initialize the Replication Agreements" on page 56 and are still logged into the ds1.idp-example.com host machine as a root user.

1 Run ldapmodify on the ds1.idp-example.com host machine to create a new directory entry.

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapmodify -a -h ds1.idp-example.com -p 1489
 -D cn=admin,cn=Administrators,cn=config -w dsmanager

dn: ou=People,dc=company,dc=com
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries
```

Hit ENTER to indicate end of input.

```
adding new entry ou=People,dc=company,dc=com
```

Hit Control C to terminate the command.

```
^C
```

This step creates a new organizational unit on Directory Server 1.

2 After the entry is created, log in to the ds2.idp-example.com host machine as a root user.

3 Run ldapsearch on Directory Server 2 to verify that the directory entry was successfully replicated.

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -b "dc=company,dc=com" -p 1489
-D "cn=Directory Manager" -w dsmanager
"objectclass=organizationalUnit"

version: 1
dn: ou=People,dc=company,dc=com
objectClass: top
objectClass: organizationalUnit
ou: People
description Container for user entries
```

4 Run ldapdelete on Directory Server 2 to delete the entry just found.

```
# ./ldapdelete -h ds2.idp-example.com -p 1489
 -D "cn=Directory Manager" -w dsmanager
"ou=People,dc=company,dc=com"
```

5 **Run** `ldapsearch` **on Directory Server 1 to verify that the entry was deleted.**

```
# ./ldapsearch -b "dc=company,dc=com"
-p 1489 -D "cn=Directory Manager" -w dsmanager
"objectclass=organizationalUnit"
```

The search will return no results as the delete was successfully replicated.

6 **Log out of both Directory Server host machines.**

4.3 Modifying the Directory Server Schema

This deployment will be used to test SAML v2 communications. Towards this end, modify the LDAP schema used by the Directory Server user data instances on the identity provider side to recognize and store SAML v2 attributes.

▼ To Modify the Directory Server LDAP Schema for SAML v2 User Data

1 **Log in to the** `ds1.idp-example.com` **host machine as a root user.**

2 **Create an LDIF file with the following information and save it as** `/tmp/saml.ldif`**.**

This file includes SAML v2 LDAP attributes.

```
dn: CN=schema
changetype:modify
add:attributeTypes
attributeTypes: ( 1.3.6.1.4.1.42.2.27.9.1.500
NAME 'sun-fm-saml2-nameid-infokey'
DESC 'SAML 2.0 Name Identifier Information Key'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN
'Sun Java System Access Management' )

attributeTypes: ( 1.3.6.1.4.1.42.2.27.9.1.501
NAME 'sun-fm-saml2-nameid-info'
DESC 'SAML 2.0 Name Identifier Information'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN
'Sun Java System Access Management' )
-
add:objectClasses
objectClasses: ( 1.3.6.1.4.1.42.2.27.9.2.200
NAME 'sunFMSAML2NameIdentifier'
DESC 'SAML 2.0 name identifier objectclass'
SUP top AUXILIARY MAY
( sun-fm-saml2-nameid-infokey $ sun-fm-saml2-nameid-info )
X-ORIGIN 'Sun Java System Access Management' )
```

3 **Run `ldapmodify` on the `ds1.idp-example.com` host machine using `/tmp/saml.ldif` as input.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ldapmodify -a -h ds1.idp-example.com -p 1489
-D "cn=Directory Manager" -w dsmanager -f /tmp/saml.ldif

modifying entry CN=schema
```

4 **Log out of the `ds1.idp-example.com` host machine.**

4.4 Enabling Secure Communication for the Directory Server User Data Instances

By default, when an instance of Directory Server is created, its SSL port is secured with a self-signed certificate named `defaultCert`. A *self-signed certificate* contains a public and private key; the public key is signed by the private key. The `idp-users` instances, though, need to use a server certificate signed by a certificate authority (CA) to allow for secure communication between the instances and the soon-to-be-installed load balancer. This entails installing a CA root certificate and a server certificate (signed by the CA root certificate) on both Directory Server host machines. Use the following list of procedures as a checklist for completing this task.

1. "To Import a Root Certificate and a Server Certificate to Directory Server 1" on page 60
2. "To Import a Root Certificate and a Server Certificate to Directory Server 2" on page 62

▼ To Import a Root Certificate and a Server Certificate to Directory Server 1

Before You Begin You should already have a root certificate from the CA of your choice. Send server certificate requests to the same CA. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1 **Log in to the `ds1.idp-example.com` host machine as a root user.**

2 **Generate a request for a server certificate signed by a CA.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsadm request-cert -S "CN=ds1.idp-example.com,
OU=OpenSSO Enterprise, O=Sun Microsystems, L=Santa Clara
 ST=California, C=US" -F ascii -o ds1.csr /var/opt/mps/idp-users
```

`ds1.csr` is the certificate request.

3 **Send `ds1.csr` to the CA of your choice.**

The CA issues and returns a certified server certificate named `ds1.cer`.

4 **Add `ds1.cer`, the CA-signed server certificate, to the certificate store.**

```
# ./dsadm add-cert /var/opt/mps/idp-users ds1 ds1.cer
```

5 **(Optional) Verify that the certificate was successfully added.**

```
# ./dsadm list-certs /var/opt/mps/idp-users
```

A list of certificates for the `idp-users` instance is displayed including the `defaultCert` and `ds1`.

6 **Add `ca.cer`, the root certificate, to the certificate store.**

```
# ./dsadm add-cert --ca /var/opt/mps/idp-users CA-cert ca.cer
```

7 **(Optional) Verify that the root certificate was successfully added.**

```
# ./dsadm list-certs -C /var/opt/mps/idp-users | grep CA-cert

CA-cert
2007/09/20 11:41  2010/06/17 11:41  n
E=nobody@nowhere.com,CN=openssltestca,OU=am,
O=sun,L=santa clara,ST=california,C=us  Same as issuer
```

8 **Configure the Directory Server instance to use the imported certificates.**

```
# ./dsconf set-server-prop -h ds1.idp-example.com
-p 1489 ssl-rsa-cert-name:ds1

Enter "cn=Directory Manager" password: dsmanager

Before setting SSL configuration, export Directory Server data.

Do you want to continue [y/n] ?  y

Directory Server must be restarted for changes to take effect.
```

9 **Restart the Directory Server instance.**

```
# ./dsadm stop /var/opt/mps/idp-users
# ./dsadm start /var/opt/mps/idp-users

Server started: pid=5472
```

10 **Run `ldapsearch` on Directory Server 1 to verify that the directory entries can be accessed through the secure port.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds1.idp-example.com -p 1736
-Z -P /var/opt/mps/idp-users/alias/slapd-cert8.db
-b "" -s base "(objectclass=*)"

version: 1
dn:
objectClass:top
namingContexts: dc=company,dc=com
supportedExtension: 2.16.840.1.113730.3.5.7
:
supportedSSLCiphers: SSL-CK_RC4_128_EXPORT40_WITH_MD5
supportedSSLCiphers: SSL-CK_RC2_128_CBC_EXPORT40_WITH_MD5
```

This confirms that the Directory Server instance can be accessed through the secure port.

11 **Log out of the `ds1.idp-example.com` host machine.**

▼ To Import a Root Certificate and a Server Certificate to Directory Server 2

Before You Begin You should already have a root certificate from the CA of your choice. Send any server certificate requests to the same CA. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1 **Log in to the `ds2.idp-example.com` host machine as a root user.**

2 **Generate a request for a server certificate signed by a CA.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsadm request-cert -S "CN=ds2.idp-example.com,
OU=OpenSSO Enterprise, O=Sun Microsystems, L=Santa Clara
 ST=California, C=US" -F ascii -o ds2.csr /var/opt/mps/idp-users
```

`ds2.csr` is the certificate request.

3 **Send `ds2.csr` to the CA of your choice.**

The CA issues and returns a certified server certificate named `ds2.cer`.

4 **Add `ds2.cer`, the CA-signed server certificate, to the certificate store.**

```
# ./dsadm add-cert /var/opt/mps/idp-users ds2 ds2.cer
```

5 **(Optional) Verify that the certificate was successfully added.**

```
# ./dsadm list-certs /var/opt/mps/idp-users
```

A list of certificates for the `idp-users` instance is displayed including the `defaultCert` and `ds2`.

6 **Add `ca.cer`, the root certificate, to the certificate store.**

```
# ./dsadm add-cert --ca /var/opt/mps/idp-users CA-cert ca.cer
```

7 **(Optional) Verify that the root certificate was successfully added.**

```
# ./dsadm list-certs -C /var/opt/mps/idp-users | grep CA-cert

CA-cert
2007/09/20 11:41  2010/06/17 11:41  n
E=nobody@nowhere.com,CN=openssltestca,OU=am,
O=sun,L=santa clara,ST=california,C=us  Same as issuer
```

8 **Configure the Directory Server instance to use the imported certificates.**

```
# ./dsconf set-server-prop -h ds2.idp-example.com
-p 1489 ssl-rsa-cert-name:ds2

Enter "cn=Directory Manager" password: dsmanager

Before setting SSL configuration, export Directory Server data.

Do you want to continue [y/n] ?  y

Directory Server must be restarted for changes to take effect.
```

9 **Restart the Directory Server instance.**

```
# ./dsadm stop /var/opt/mps/idp-users
# ./dsadm start /var/opt/mps/idp-users

Server started: pid=5472
```

10 **Run `ldapsearch` on Directory Server 2 to verify that the directory entries can be accessed through the secure port.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds2.idp-example.com -p 1736
-Z -P /var/opt/mps/idp-users/alias/slapd-cert8.db
-b "" -s base "(objectclass=*)"

version: 1
dn:
objectClass:top
namingContexts: dc=company,dc=com
```

```
supportedExtension: 2.16.840.1.113730.3.5.7
:
supportedSSLCiphers: SSL-CK_RC4_128_EXPORT40_WITH_MD5
supportedSSLCiphers: SSL-CK_RC2_128_CBC_EXPORT40_WITH_MD5
```

This confirms that the Directory Server instance can be accessed through the secure port.

11 **Log out of the `ds2.idp-example.com` host machine.**

4.5 Configuring the Directory Server Load Balancer

Load Balancer 1 (`lb1.idp-example.com`) is configured in front of the Directory Server instances on the identity provider side. This section assumes that you have already installed the load balancer. Before beginning, note the following:

- The load balancer hardware and software used in the lab facility for this deployment is BIG-IP® manufactured by F5 Networks. If you are using different load balancer software, see the documentation that comes with that product for detailed settings information.
- Contact your network administrator to obtain an available virtual IP address for the load balancer you want to configure.
- Know the IP address of the load balancer hardware, the URL for the load balancer login page, and a username and password for logging in to the load balancer application.
- Get the IP addresses for Directory Server 1 and Directory Server 2 by running the following command on each host machine:

```
# ifconfig -a
```

Use the following list of procedures as a checklist for completing the task.

1. "To Import the Root Certificate to Directory Server Load Balancer 1" on page 64
2. "To Configure the Directory Server Load Balancer 1" on page 65

▼ To Import the Root Certificate to Directory Server Load Balancer 1

Import the CA root certificate to the Directory Server Load Balancer 1 to ensure that a link between Load Balancer 1 can be maintained with the CA.

Before You Begin Use the same root certificate that you imported in "4.4 Enabling Secure Communication for the Directory Server User Data Instances" on page 60. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1 **Access `https://lb1.idp-example.com`, the BIG-IP load balancer login page, in a web browser.**

2 **Log in to the load balancer as administrator.**

3 **Click Proxies.**

4 **Click the Cert-Admin tab.**

5 **Click Import.**

6 **In the Import Type field, choose Certificate and click Continue.**

7 **Click Browse in the Certificate File field on the Install SSL Certificate page.**

8 **Choose Browser in the Choose File dialog box.**

9 **Navigate to `ca.cer` and click Open.**

10 **Enter `OpenSSL_CA_cert` in the Certificate Identifier field.**

11 **Click Install Certificate.**

The CertificateOpenSSL_CA_Cert page is displayed.

12 **Click Return to Certificate Administration on the CertificateOpenSSL_CA_Cert page.**

OpenSSL_CA_Cert, the root certificate, is now included in the Certificate ID list.

▼ To Configure the Directory Server Load Balancer 1

Before You Begin This procedure assumes that you have just completed "To Import the Root Certificate to Directory Server Load Balancer 1" on page 64 and are still logged into the load balancer console.

1 **Click** ***Configure your BIG-IP (R) using the Configuration Utility*****.**

2 **Create a Pool.**

A pool contains all the backend server instances.

a. **In the left pane, click Pools.**

b. **On the Pools tab, click Add.**

c. **In the Add Pool dialog, provide the following information:**

Pool Name	**`DirectoryServerIDP-UserData-Pool`**
Load Balancing Method	Round Robin
Resources	Add the IP address and port number of both Directory Server host machines. **Note** – User port number `1736`.

d. **Click Done.**

3 **Add a Virtual Server.**

The virtual server presents an address to the outside world and, when users attempt to connect, it would forward the connection to the most appropriate real server.

Tip – If you encounter JavaScript™ errors or otherwise cannot proceed to create a virtual server, try using Internet Explorer.

---

a. **In the left frame, click Virtual Servers.**

b. **Click Add on the Virtual Servers tab.**

c. **In the Add a Virtual Server dialog box, provide the following information:**

Address	Enter the IP address for `lb1.idp-example.com`.
Service	**`489`**

d. **Continue to click Next until you reach the Pool Selection dialog box.**

e. **Assign `DirectoryServerIDP-UserData-Pool` to the virtual server in the Pool Selection dialog box.**

f. **Click Done.**

4 **Add Monitors**

Monitors are required for the load balancer to detect the backend server failures.

a. **In the left frame, click Monitors.**

b. **Click the Basic Associations tab.**

c. **Add an LDAP monitor for the Directory Server 1 node.**

   In the Node column, locate the IP address and port number previously entered for Directory Server 1 and select the Add checkbox.

d. **Add an LDAP monitor for the Directory Server 2 node.**

   In the Node column, locate the IP address and port number previously entered for Directory Server 2 and select the Add checkbox.

e. **At the top of the Node column, in the drop-down list, choose `tcp`.**

f. **Click Apply.**

5 **Configure the load balancer for simple persistence.**

With simple persistence, all requests sent *within a specified interval* are processed by the same Directory Server instance, ensuring complete replication of entries. For example, when a request requires information to be written to Directory Server 1, that information must also be replicated to Directory Server 2. As the replication takes time to complete, if a related request is directed by the load balancer to Directory Server 2 during the replication process itself, the request may fail as the entry might only be partially created. When properly configured, simple persistence ensures that both requests are routed to Directory Server 1 and processed in consecutive order; the first request is finished before the second request begins processing. Simple persistence ensures that within the specified interval, no errors or delays occur due to replication time or redirects when retrieving data. Simple persistence tracks connections based only on the client IP address.

a. **In the left frame, click Pools.**

b. **Click the name of the pool you want to configure.**

   In this example, `DirectoryServerIDP-UserData-Pool`.

c. **Click the Persistence tab.**

d. **Under Persistence Type, select Simple.**

e. **Enter 300 seconds for the Timeout interval.**

f. **Click Apply.**

6 **Verify the load balancer configuration with the following sub procedure.**

a. **Log in as a root user on each Directory Server host machine.**

b. **On each host machine, use the `tail` command to monitor the Directory Server access log.**

```
# cd /var/opt/mps/idp-users/logs
# tail -f access
```

You should see connections to the load balancer IP address opening and closing. For example:

```
[12/July/2008:13:10:20-0700] conn=69755 op=-1 msgId=-1 - closed
[12/July/2008:13:10:25-0700] conn=69756 op=-1 msgId=-1
- fd=27 slot=27 LDAP connection from IP_address to IP_address
[12/July/2008:13:10:25-0700] conn=69756 op=0 msgId=0
- RESULT err=80 tag=120 nentries=0 etime=0
[12/July/2008:13:10:25-0700] conn=69756 op=-1 msgId=-1
- closing from IP_address
```

c. **Execute the following LDAP search against the Directory Server load balancer from Directory Server 1.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h lb1.idp-example.com -p 489 -Z
-P /var/opt/mps/idp-users/alias/slapd-cert8.db
-b "dc=company,dc=com" -D "cn=directory manager"
-w dsmanager "(objectclass=*)"

version: 1
dn: dc=company,dc=com
dc: company
objectClass: top
objectClass: domain
```

Make sure the returned entries display in the access log on only one Directory Server host machine.

d. **Run `dsadm stop` to stop Directory Server 1.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsadm stop /var/opt/mps/idp-users
```

e. **Perform the (same) LDAP search against the Directory Server load balancer from Directory Server 2.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h lb1.idp-example.com -p 489 -Z
-P /var/opt/mps/idp-users/alias/slapd-cert8.db
-b "dc=company,dc=com" -D "cn=directory manager"
-w dsmanager "(objectclass=*)"

version: 1
dn: dc=company,dc=com
```

```
dc: company
objectClass: top
objectClass: domain
```

Make sure that the returned entries display in the access log on only Directory Server 2.

Note – You may encounter the following error message:

```
ldap_simple_bind: Cant’ connect to the LDAP
server — Connection refused
```

This means that the load balancer may not fully detect that Directory Server 1 is stopped. In this case, you may have started the search too soon based on the polling interval setting. For example, if the polling interval is set to 10 seconds, you should wait ten seconds to start the search. You can reset the timeout properties to a lower value using the load balancer console.

a. **Click the Monitors tab.**

b. **Click the tcp monitor name.**

c. **In the Interval field, set the value to 5.**

   This tells the load balancer to poll the server every 5 seconds.

d. **In the Timeout field, set the value to 16.**

e. **Click Apply and repeat the LDAP search.**

See your load balancer documentation for more information on the timeout property.

---

f. **Start Directory Server 1.**

```
# ./dsadm start /var/opt/mps/idp-users
```

g. **Stop Directory Server 2.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsadm stop /var/opt/mps/idp-users
```

h. **Perform the following LDAP search against the Directory Server load balancer from Directory Server 1.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
 ./ldapsearch -h lb1.idp-example.com -p 489 -Z
-P /var/opt/mps/idp-users/alias/slapd-cert8.db
-b "dc=company,dc=com" -D "cn=directory manager"
-w dsmanager "(objectclass=*)"

version: 1
dn: dc=company,dc=com
```

```
dc: company
objectClass: top
objectClass: domain
```

Make sure the returned entries display in the access log on only Directory Server 1.

i. **Start Directory Server 2.**

```
# ./dsadm start /var/opt/mps/idp-users
```

j. **Log out of both Directory Server host machines and the load balancer console.**

4.6 Creating a Test User

Create a user entry in the replicated Directory Server user data instances for idpuser.

Note – If you are using an existing user data store, create the appropriate users in it and move on to Chapter 6, "Configuring OpenSSO Enterprise Realms for User Authentication."

▼ To Import Test User Data into the Replicated Directory Server Instances

Create an LDIF file for the test user and import the file into ds1.idp-example.com. The test user data will then be replicated to ds2.idp-example.com.

1 **Log in to the ds1.idp-example.com host machine as a root user.**

2 **Create an LDIF file with the following entries.**

```
dn: ou=users,dc=company,dc=com
objectclass: top
objectclass: organizationalUnit
ou: users
description: Container for user entries

dn: ou=Groups,dc=company,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Groups
description: Container for group entries

dn: uid=idpuser,ou=users,dc=company,dc=com
uid: idpuser
```

```
givenName: idp
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetadmin
objectClass: inetorgperson
objectClass: inetUser
sn: user
cn: idp user
userPassword: idpuser
inetUserStatus: Active
```

3 **Save the file as `idp-users.ldif` in the `/tmp` directory.**

4 **Import the LDIF file into Directory Server 1 using `ldapmodify`.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapmodify -h ds1.idp-example.com -p 1489
 -D "cn=Directory Manager" -w dsmanager
 -a -f /tmp/idp-users.ldif

adding new entry ou=users,dc=company,dc=com

adding new entry ou=Groups,dc=company,dc=com

adding new entry uid=idpuser,ou=users,dc=company,dc=com
```

5 **Verify that the new users were imported using `ldapsearch`.**

```
# ./ldapsearch -h ds1.idp-example.com
 -b "dc=company,dc=com" -p 1489 -D "cn=Directory Manager"
 -w dsmanager "uid=idpuser"

version: 1
dn: uid=idpuser,ou=users,dc=company,dc=com
uid: idpuser
givenName: idp
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetadmin
objectClass: inetorgperson
objectClass: inetUser
sn: user
cn: idp user
userPassword:
 {SSHA}H5LpB+QLZMoL9SiXzY/DokHKXRclELVy7w25AA==
inetUserStatus: Active
```

6 **Log out of the `ds1.idp-example.com` host machine.**

7 **(Optional) Verify that the entries were replicated to Directory Server 2 by logging in as a root user to the `ds2.idp-example.com` host machine and using `ldapsearch`.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds2.idp-example.com
 -b "dc=company,dc=com" -p 1489 -D "cn=Directory Manager"
 -w dsmanager ""

version: 1
dn: dc=company,dc=com
objectClass: top
objectClass: domain
dc: company

dn: ou=users,dc=company,dc=com
objectClass: top
objectClass: organizationalUnit
ou: users
description: Container for user entries

dn: ou=Groups,dc=company,dc=com
objectClass: top
objectClass: organizationalUnit
ou: Groups
description: Container for group entries

dn: uid=idpuser,ou=users,dc=company,dc=com
uid: idpuser
givenName: idp
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetadmin
objectClass: inetorgperson
objectClass: inetUser
sn: user
cn: idp user
userPassword:
 {SSHA}H5LpB+QLZMoL9SiXzY/DokHKXRclELVy7w25AA==
inetUserStatus: Active
```

8 **Log out of the `ds2.idp-example.com` host machine.**

CHAPTER 5

Deploying and Configuring OpenSSO Enterprise

This chapter includes instructions on how to deploy and configure two instances of Sun OpenSSO Enterprise 8.0. It begins with the installation of Sun Java™ System Application Server onto each host machine, followed by the deployment and configuration of the OpenSSO Enterprise WAR. This chapter contains the following sections:

- "5.1 Installing the Application Server Web Containers" on page 73
- "5.2 Configuring the OpenSSO Enterprise Load Balancer" on page 94
- "5.3 Deploying and Configuring OpenSSO Enterprise 1 and OpenSSO Enterprise 2" on page 102
- "5.4 Configuring the OpenSSO Enterprise Platform Service" on page 112
- "5.5 Configuring OpenSSO Enterprise for SAML v2" on page 115

5.1 Installing the Application Server Web Containers

In this section, we create a non-root user with the `roleadd` command in the Solaris Operating Environment on each OpenSSO Enterprise host machine and install Sun Java System Application Server 9.1 Update 1 using the non-root user. The final procedures in the installation process is to request and import certificates for secure communications with a soon-to-be-configured load balancer. Use the following list of procedures as a checklist for completing the task.

1. "To Patch the OpenSSO Enterprise Host Machines" on page 74
2. "To Create a Non-Root User on the OpenSSO Enterprise 1 Host Machine" on page 74
3. "To Install Application Server on the OpenSSO Enterprise 1 Host Machine" on page 75
4. "To Create a Non-Root User on the OpenSSO Enterprise 2 Host Machine" on page 84
5. "To Install Application Server on the OpenSSO Enterprise 2 Host Machine" on page 85

Note – We use roleadd rather than useradd for security reasons; roleadd disables the ability of the user to log in.

▼ To Patch the OpenSSO Enterprise Host Machines

On our lab machines, the required Application Server patch is 117461–08. Results for your machine might be different. Read the latest documentation for your web container to determine if you need to install patches and, if so, what they might be. You can search for patches directly at http://sunsolve.sun.com. Navigate to the PatchFinder page, enter the patch number, click Find Patch, and download the appropriate patch for the OpenSSO Enterprise 1 host machine (osso1.idp-example.com) and the OpenSSO Enterprise 2 host machine (osso2.idp-example.com).

1 **Log in to the** osso1.idp-example.com **host machine as a root user.**

2 **Run** patchadd **to see if the patch is already installed.**

```
# patchadd -p | grep 117461-08
```

A series of patch numbers are displayed, and patch 117461–08 is present so there is no need to install any patches at this time.

3 **Log out of the** osso1.idp-example.com **host machine.**

4 **Log in to the** osso2.idp-example.com **host machine as a root user.**

5 **Run** patchadd **to see if the patch is already installed.**

```
# patchadd -p | grep 117461-08
```

A series of patch numbers are displayed, and patch 117461–08 is present so there is no need to install any patches at this time.

6 **Log out of the** osso2.idp-example.com **host machine.**

▼ To Create a Non-Root User on the OpenSSO Enterprise 1 Host Machine

1 **Log in to the** osso1.idp-example.com **host machine as a root user.**

2 **Create a new user with** roleadd**.**

```
# roleadd -s /sbin/sh -m -g staff -d /export/osso80adm osso80adm
```

3 **(Optional) Verify that the user was created.**

```
# cat /etc/passwd

root:x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1::/:
...
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
osso80adm:x:223830:10::/export/osso80adm:/sbin/sh
```

4 **(Optional) Verify that the user's directory was created.**

```
# cd /export/osso80adm
# ls

local.cshrc    local.profile    local.login
```

5 **Create a password for the non-root user.**

```
# passwd osso80adm
New Password: nonroot1pwd
Re-ener new Pasword: nonroot1pwd

passwd: password successfully changed for osso80adm
```

Caution – If you do not perform this step, you will not be able to *switch user* (su) when logged in as the non-root user.

To Install Application Server on the OpenSSO Enterprise 1 Host Machine

Before You Begin This procedure assumes you have just completed "To Create a Non-Root User on the OpenSSO Enterprise 1 Host Machine" on page 74 and are still logged into the osso1.idp-example.com host machine as a root user.

1 **Create a directory into which the Application Server bits can be downloaded and change into it.**

```
# mkdir /export/AS91
# cd /export/AS91
```

2 **Download the Sun Java System Application Server 9.1 Update 1 binary from the Sun Microsystems Product Download page to the /export/AS91 directory.**

3 **Grant the downloaded binary execute permission using the chmod command.**

```
# chmod +x sjsas-9_1_01-solaris-sparc.bin
```

4 **Install the software.**

```
# ./sjsas-9_1_01-solaris-sparc.bin -console
```

5 **When prompted, provide the following information.**

`You are running the installation program for the Sun Java System Application Server. This program asks you to supply configuration preference settings that it uses to install the server.`
`This installation program consists of one or more selections that provide you with information and let you enter preferences that determine how Sun Java System Application Server is installed and configured.`
`When you are presented with the following question, the installation process pauses to allow you to read the information that has been presented When you are ready to continue, press Enter.`
	Press Enter to continue.

`Have you read, and do you accept, all of the terms of the preceding Software License Agreement [no] {"<" goes back, "!" exits}?`	Enter **`yes`**.
`Installation Directory [/opt/SUNWappserver] {"<" goes back, "!" exits}`	Enter **`/opt/SUNWappserver91`**

`The specified directory "/opt/SUNWappserver91" does not exist. Do you want to create it now or choose another directory?`
`1. Create Directory`
`2. Choose New.`
`Enter the number corresponding to your choice [1] {"<" goes back, "!" exits}`
	Enter **`1`** to create the directory.

`The Sun Java System Application Server requires a Java 2 SDK. Please provide the path to a Java 2 SDK 5.0 or greater. [/usr/jdk/instances/jdk1.5.0] {"<" goes back, "!" exits}`	Press Enter to accept the default value.

`Supply the admin user’s password and override any of the other initial configuration settings as necessary.`
`Admin User [admin] {"<" goes back, "!" exits}`
	Press Enter to accept the default value.

`Admin User’s Password (8 chars minimum):`
`Re-enter Password:`
	Enter **`domain1pwd`** and then re-enter **`domain1pwd`**.

`Do you want to store admin user name and`
`password in .asadminpass file in user’s home`
`directory [yes] {"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`Admin Port [4848] {"<" goes back, "!" exits}`
`HTTP Port [8080] {"<" goes back, "!" exits}`
`HTTPS Port [8181] {"<" goes back, "!" exits}`
	Press Enter to accept the three default values.

`Do you want to enable Updatecenter client`
`[yes] {"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`Do you want to upgrade from previous`
`Applicatin Server version [no]`
`{"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`The following items for the product Sun Java`
`System Application Server will be installed:`
 	
`Product: Sun Java System Application Server`
`Location: /opt/SUNWappserver91`
`Space Required: 161.61 MB`
`-------------------------------------------`
`Sun Java System message Queue 4.1`
`Application Server`
`Startup`
 	
`Ready To Install`
 	
`1. Install Now`
`2. Start Over`
`3. Exit Installation`
 	
`What would you like to do [1]`
`{"<" goes back, "!" exits}?`
	Press Enter to accept the default value and begin the installation process.

`- Installing Sun Java System Application`
`Server`
 	
`|-1%-----25%-----50%-----75%-----100%|`
 	
` - Installation Successful.`
	When installation is complete, an `Installation Successful` message is displayed:

Next Steps: 1. Access the About Application Server 9.1 welcome page at: file:///opt/SUNWappserver91/docs/about.html 2. Start the Application Server by executing: /opt/SUNWappserver91/bin/asadmin start-domain domain1 3. Start the Admin Console: http://osso1.idp-example.com:4848 Please press Enter/Return key to exit the installation program. {"!" exits}	Press Enter to exit the installation program.

6 Create a second Application Server domain for the non-root user.

The default domain created during the installation process is owned by root. We create a new domain for the non-root user `osso80adm` into which we will deploy OpenSSO Enterprise.

```
# cd /opt/SUNWappserver91/bin
# su osso80adm
# ./asadmin create-domain
--domaindir /export/osso80adm/domains
--adminport 8989 --user domain2adm --instanceport 1080
--domainproperties http.ssl.port=1081 ossodomain

 Please enter the admin password>

domain2pwd

Please enter the admin password again>

domain2pwd

Please enter the master password

  [Enter to accept the default]:>

domain2master

Please enter the master password again

  [Enter to accept the default]:>

domain2master

Using port 8989 for Admin.
Using port 1080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using port 1081 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Domain being created with profile:developer, as specified
  by variable AS_ADMIN_PROFILE in configuration file.
Security Store uses: JKS
2008-08-24 18:21:15.907 GMT Thread[main,5,main]
java.io.FileNotFoundException:
derby.log (Permission denied)
-------------------------------------------------
2008-03-24 18:21:16.216 GMT:
Booting Derby version The Apache Software Foundation
- Apache Derby - 10.2.2.1 -
(538595): instance c013800d-0118-e205-d50b-00000c0c0770
on database directory
/export/osso80adm/domains/ossodomain/lib/databases/ejbtimer

  Database Class Loader started - derby.database.classpath=’’
  Domain ossodomain created.
```

Note – Creating a non-root domain displays a `FileNotFoundException`. Please see Appendix G, “Known Issues and Limitations.”

7 **Verify that the non-root user domain was created with the correct permissions using the following sub-procedure.**

a. **Change to the `ossodomain` directory.**

```
# cd /export/osso80adm/domains/ossodomain
```

b. **List the contents of the directory.**

```
# ls -la

total 30
drwxr-xr-x  15 osso80adm staff   512 Mar 20 14:12 .
drwxr-xr-x   3 osso80adm staff   512 Mar 20 14:12 ..
drwxr-xr-x   2 osso80adm staff   512 Mar 20 14:12 addons
drwxr-xr-x   6 osso80adm staff   512 Mar 20 14:12 applications
drwxr-xr-x   3 osso80adm staff   512 Mar 20 14:12 autodeploy
drwxr-xr-x   2 osso80adm staff   512 Mar 20 14:12 bin
drwx------   3 osso80adm staff  1024 Mar 26 13:27 config
drwxr-xr-x   2 osso80adm staff   512 Mar 20 14:12 docroot
drwxr-xr-x   6 osso80adm staff   512 Mar 26 13:34 generated
drwxr-xr-x   3 osso80adm staff   512 Mar 20 14:12 imq
```

```
drwxr-xr-x   5 osso80adm staff   512 Mar 20 14:16 java-web-start
drwxr-xr-x   8 osso80adm staff   512 Mar 20 14:16 jbi
drwxr-xr-x   6 osso80adm staff   512 Mar 20 14:12 lib
drwxr-xr-x   2 osso80adm staff   512 Mar 26 13:26 logs
drwxr-xr-x   2 osso80adm staff   512 Mar 20 14:12 session-store
```

The files and directories are owned by osso80adm.

8 **Start ossodomain, the non-root user domain, using the following sub-procedure.**

a. **Switch to the non-root user.**

```
# su osso80adm
```

b. **Change to the bin directory.**

```
# cd /export/osso80adm/domains/ossodomain/bin
```

c. **Start ossodomain.**

```
# ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

9 **Verify that ossodomain has started with the following sub-procedure.**

a. **Access http://osso1.idp-example.com:8989/login.jsf from a web browser.**

b. **Log in to the Application Server console as the ossodomain administrator.**

Username	**domain2adm**
Password	**domain2pwd**

When the Application Server administration console is displayed, it is verification that the non-root user was able to start the domain server.

c. **Exit the console and close the browser.**

10 Create a request for a server certificate to secure communications between the soon-to-be-configured OpenSSO Enterprise load balancer and `ossodomain` using the following sub-procedure.

a. Generate a private/public key pair and reference it with the alias, `opensso-idp-1`.

`opensso-idp-1` will be used in a later step to retrieve the public key which is contained in a self-signed certificate.

```
# cd /export/osso80adm/domains/ossodomain/config
# keytool -genkey -noprompt -keyalg rsa -keypass domain2master
-alias opensso-idp-1 -keystore keystore.jks -dname
"CN=osso1.idp-example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US" -storepass domain2master
```

b. Verify that the key pair was successfully created and stored in the certificate store.

```
# keytool -list -v -keystore keystore.jks -storepass domain2master

 Alias name: opensso-idp-1
 Creation date: Aug 4, 2008
 Entry type: keyEntry
 Certificate chain length: 1
 Certificate[1]:
 Owner: CN=osso1.idp-example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
 Issuer: CN=osso1.idp-example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
 Serial number: 47f6a587
 Valid from: Fri Aug 04 15:02:47 PDT 2008 until: Thu Nov 03 15:02:47 PDT 2008
 Certificate fingerprints:
  MD5:  62:0E:5E:EB:8A:73:B2:F9:08:83:05:C5:DC:07:3C:E1
  SHA1: D4:9C:BA:25:4C:B5:71:20:CF:F3:18:46:AF:2E:7F:71:2A:4B:BD:B3

The certificate indicated by the alias "opensso-idp-1" is a
self-signed certificate.
```

Note – The output of this command may list more than one certificate based on the entries in the keystore.

c. Generate a server certificate request.

```
# keytool -certreq -alias opensso-idp-1 -keypass domain2master
-keystore keystore.jks -storepass domain2master file opensso-idp-1.csr
```

`opensso-idp-1.csr` is the server certificate request.

d. (Optional) Verify that `opensso-idp-1.csr` was created.

```
# ls -la opensso-idp-1.csr

 -rw-r--r--   1 osso80adm staff        715 Apr  4 15:04 opensso-idp-1.csr
```

e. Send `opensso-idp-1.csr` to the CA of your choice.

The CA issues and returns a certified certificate named `opensso-idp-1.cer`.

f. Import `ca.cer`, the CA root certificate.

The root certificate must be imported into two keystores (`keystore.jks` and `cacerts.jks`) with Application Server.

```
# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore keystore.jks -storepass domain2master

Owner: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore

# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore cacerts.jks -storepass domain2master

Owner: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore
```

g. **Replace the self-signed public key certificate (associated with the `s1as` alias) with the server certificate received from the CA.**

```
# keytool -import -file opensso-idp-1.cer -alias opensso-idp-1
-keystore keystore.jks -storepass domain2master

Certificate reply was installed in keystore
```

h. **(Optional) Verify that the self-signed public key certificate has been overwritten by the server certificate received from the CA.**

```
# keytool -list -v -keystore keystore.jks
-storepass domain2master

The certificate indicated by the alias "opensso-idp-1" is signed by CA.
```

i. **Change the certificate alias from the default `s1as` to the new `opensso-idp-1` in the `domain.xml` file for the `ossodomain` domain.**

The Application Server configuration file is `domain.xml`.

```
<http-listener acceptor-threads="1" address="0.0.0.0"
blocking-enabled="false" default-virtual-server="server" enabled="true"
family="inet" id="http-listener-2" port="1081" security-enabled="true"
server-name="" xpowered-by="true">
<ssl cert-nickname="opensso-idp-1" client-auth-enabled="false" ssl2-enabled="false"
ssl3-enabled="true" tls-enabled="true" tls-rollback-enabled="true"/>
```

Tip – Backup `domain.xml` before modifying it.

11 **Modify the JVM options in your web container's configuration file using the following sub-procedure.**

OpenSSO Enterprise is deployed with an embedded configuration data store (if desired). In order for the configuration data store to be created successfully, the following JVM options should be modified in the web container's configuration file. We will be modifying `domain.xml` again for this example.

Tip – Backup `domain.xml` before modifying it.

a. **Change to the `config` directory.**

```
# cd /export/osso80adm/domains/ossodomain/config
```

b. **Open `domain.xml` in a text editor and make the following changes:**

- Replace **`<jvm-options>-client</jvm-options>`** with **`<jvm-options>-server</jvm-options>`**.

- Replace **`<jvm-options>-Xmx512m</jvm-options>`** with **`<jvm-options>-Xmx1024m</jvm-options>`**.

c. **Save the file and close it.**

12 **Restart the `ossodomain` domain.**

```
# cd /export/osso80adm/domains/ossodomain/bin
# ./stopserv

Server was successfully stopped.

./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

13 **Verify that the certificate used for SSL communication is the root CA certificate.**

a. **Access `https://osso1.idp-example.com:1081/index.html` from a web browser.**

b. **View the details of the certificate in the security warning to ensure that it is Issued by "OpenSSLTestCA".**

After inspecting and accepting the certificate, you should see the default `index.html` page.

c. **Close the browser.**

14 **Log out of the `osso1.idp-example.com` host machine.**

▼ To Create a Non-Root User on the OpenSSO Enterprise 2 Host Machine

1 **Log in to the `osso2.idp-example.com` host machine as a root user.**

2 **Create a new user with `roleadd`.**

```
# roleadd -s /sbin/sh -m -g staff -d /export/osso80adm osso80adm
```

3 **(Optional) Verify that the user was created.**

```
# cat /etc/passwd

root:x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1::/:
...
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
osso80adm:x:223830:10::/export/osso80adm:/sbin/sh
```

4 **(Optional) Verify that the user's directory was created.**

```
# cd /export/osso80adm
# ls

local.cshrc    local.profile    local.login
```

5 **Create a password for the non-root user.**

```
# passwd osso80adm
New Password: nonroot2pwd
Re-ener new Pasword: nonroot2pwd

passwd: password successfully changed for osso80adm
```

⚠ **Caution –** If you do not perform this step, you will not be able to *switch user* (su) when logged in as the non-root user.

▼ To Install Application Server on the OpenSSO Enterprise 2 Host Machine

Before You Begin This procedure assumes you have just completed "To Create a Non-Root User on the OpenSSO Enterprise 2 Host Machine" on page 84 and are still logged into the osso2.idp-example.com host machine as a root user.

1 **Create a directory into which the Application Server bits can be downloaded and change into it.**

```
# mkdir /export/AS91
# cd /export/AS91
```

2 **Download the Sun Java System Application Server 9.1 Update 1 binary from the Sun Microsystems Product Download page to the /export/AS91 directory.**

3 **Grant the downloaded binary execute permission using the chmod command.**

```
# chmod +x sjsas-9_1_01-solaris-sparc.bin
```

4 Install the software.

```
# ./sjsas-9_1_01-solaris-sparc.bin -console
```

5 When prompted, provide the following information.

You are running the installation program for the Sun Java System Application Server. This program asks you to supply configuration preference settings that it uses to install the server. This installation program consists of one or more selections that provide you with information and let you enter preferences that determine how Sun Java System Application Server is installed and configured. When you are presented with the following question, the installation process pauses to allow you to read the information that has been presented When you are ready to continue, press Enter.	Press Enter to continue.
Have you read, and do you accept, all of the terms of the preceding Software License Agreement [no] {"<" goes back, "!" exits}?	Enter **yes**.
Installation Directory [/opt/SUNWappserver] {"<" goes back, "!" exits}	Enter **/opt/SUNWappserver91**
The specified directory "/opt/SUNWappserver91" does not exist. Do you want to create it now or choose another directory? 1. Create Directory 2. Choose New. Enter the number corresponding to your choice [1] {"<" goes back, "!" exits}	Enter **1** to create the directory.
The Sun Java System Application Server requires a Java 2 SDK. Please provide the path to a Java 2 SDK 5.0 or greater. [/usr/jdk/instances/jdk1.5.0] {"<" goes back, "!" exits}	Press Enter to accept the default value.
Supply the admin user’s password and override any of the other initial configuration settings as necessary. Admin User [admin] {"<" goes back, "!" exits}	Press Enter to accept the default value.

`Admin User’s Password (8 chars minimum):`
`Re-enter Password:`
	Enter **`domain1pwd`** and then re-enter **`domain1pwd`**.

`Do you want to store admin user name and`
`password in .asadminpass file in user’s home`
`directory [yes] {"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`Admin Port [4848] {"<" goes back, "!" exits}`
`HTTP Port [8080] {"<" goes back, "!" exits}`
`HTTPS Port [8181] {"<" goes back, "!" exits}`
	Press Enter to accept the three default values.

`Do you want to enable Updatecenter client`
`[yes] {"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`Do you want to upgrade from previous`
`Applicatin Server version [no]`
`{"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`The following items for the product Sun Java`
`System Application Server will be installed:`
 	
`Product: Sun Java System Application Server`
`Location: /opt/SUNWappserver91`
`Space Required: 161.61 MB`
`-------------------------------------------`
`Sun Java System message Queue 4.1`
`Application Server`
`Startup`
 	
`Ready To Install`
 	
`1. Install Now`
`2. Start Over`
`3. Exit Installation`
 	
`What would you like to do [1]`
`{"<" goes back, "!" exits}?`
	Press Enter to accept the default value and begin the installation process.

`- Installing Sun Java System Application`
`Server`
 	
`|-1%-----25%-----50%-----75%-----100%|`
 	
` - Installation Successful.`
	When installation is complete, an `Installation Successful` message is displayed:

<table>
<tr>
<td><pre>Next Steps:

1. Access the About Application Server 9.1 welcome
page at:
 file:///opt/SUNWappserver91/docs/about.html

2. Start the Application Server by executing:
  /opt/SUNWappserver91/bin/asadmin
  start-domain domain1

3. Start the Admin Console:
  http://osso2.idp-example.com:4848

Please press Enter/Return key to exit the
installation program. {"!" exits}</pre></td>
<td>Press Enter to exit the installation program.</td>
</tr>
</table>

6 Create a second Application Server domain for the non-root user.

The default domain created during the installation process is owned by root. We create a new domain for the non-root user `osso80adm` into which we will deploy OpenSSO Enterprise.

```
# cd /opt/SUNWappserver91/bin
# su osso80adm
# ./asadmin create-domain
--domaindir /export/osso80adm/domains
--adminport 8989 --user domain2adm --instanceport 1080
--domainproperties http.ssl.port=1081 ossodomain

 Please enter the admin password>

domain2pwd

Please enter the admin password again>

domain2pwd

Please enter the master password

  [Enter to accept the default]:>

domain2master

Please enter the master password again

  [Enter to accept the default]:>

domain2master

Using port 8989 for Admin.
Using port 1080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using port 1081 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Domain being created with profile:developer, as specified
  by variable AS_ADMIN_PROFILE in configuration file.
Security Store uses: JKS
2008-08-24 18:21:15.907 GMT Thread[main,5,main]
java.io.FileNotFoundException:
derby.log (Permission denied)
-------------------------------------------------
2008-03-24 18:21:16.216 GMT:
Booting Derby version The Apache Software Foundation
- Apache Derby - 10.2.2.1 -
(538595): instance c013800d-0118-e205-d50b-00000c0c0770
on database directory
/export/osso80adm/domains/ossodomain/lib/databases/ejbtimer

  Database Class Loader started - derby.database.classpath=’’
  Domain ossodomain created.
```

Note – The `FileNotFoundException` is a known issue. Please see Appendix G, “Known Issues and Limitations.”

7 **Verify that the non-root user domain was created with the correct permissions using the following sub-procedure.**

a. **Change to the `ossodomain` directory.**

```
# cd /export/osso80admin/domains/ossodomain
```

b. **List the contents of the directory.**

```
# ls -la

total 30
drwxr-xr-x  15 osso80adm staff   512 Mar 20 14:12 .
drwxr-xr-x   3 osso80adm staff   512 Mar 20 14:12 ..
drwxr-xr-x   2 osso80adm staff   512 Mar 20 14:12 addons
drwxr-xr-x   6 osso80adm staff   512 Mar 20 14:12 applications
drwxr-xr-x   3 osso80adm staff   512 Mar 20 14:12 autodeploy
drwxr-xr-x   2 osso80adm staff   512 Mar 20 14:12 bin
drwx------   3 osso80adm staff  1024 Mar 26 13:27 config
drwxr-xr-x   2 osso80adm staff   512 Mar 20 14:12 docroot
drwxr-xr-x   6 osso80adm staff   512 Mar 26 13:34 generated
drwxr-xr-x   3 osso80adm staff   512 Mar 20 14:12 imq
```

```
drwxr-xr-x   5 osso80adm staff   512 Mar 20 14:16 java-web-start
drwxr-xr-x   8 osso80adm staff   512 Mar 20 14:16 jbi
drwxr-xr-x   6 osso80adm staff   512 Mar 20 14:12 lib
drwxr-xr-x   2 osso80adm staff   512 Mar 26 13:26 logs
drwxr-xr-x   2 osso80adm staff   512 Mar 20 14:12 session-store
```

The files and directories are owned by `osso80adm`.

8 **Start `ossodomain`, the non-root user domain, using the following sub-procedure.**

a. **Switch to the non-root user.**

```
# su osso80adm
```

b. **Change to the `bin` directory.**

```
# cd /export/osso80adm/domains/ossodomain/bin
```

c. **Start `ossodomain`.**

```
# ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

9 **Verify that `ossodomain` has started with the following sub-procedure.**

a. **Access `http://osso2.idp-example.com:8989/login.jsf` from a web browser.**

b. **Log in to the Application Server console as the administrator.**

Username	**`domain2adm`**
Password	**`domain2pwd`**

When the Application Server administration console is displayed, it is verification that the non-root user was able to start the domain server.

c. **Exit the console and close the browser.**

10 Create a request for a server certificate to secure communications between the soon-to-be-configured OpenSSO Enterprise load balancer and `ossodomain` using the following sub-procedure.

a. Generate a private/public key pair and reference it with the alias, `opensso-idp-2`.

`opensso-idp-2` will be used in a later step to retrieve the public key which is contained in a self-signed certificate.

```
# cd /export/osso80adm/domains/ossodomain/config
# keytool -genkey -noprompt -keyalg rsa -keypass domain2master
-alias opensso-idp-2 -keystore keystore.jks -dname "CN=osso2.idp-example.com,
OU=OpenSSO, O=Sun Microsystems, L=Santa Clara, ST=California, C=US"
-storepass domain2master
```

b. Verify that the key pair was successfully created and stored in the certificate store.

```
# keytool -list -v -keystore keystore.jks -storepass domain2master

 Alias name: opensso-idp-2
 Creation date: Aug 4, 2008
 Entry type: keyEntry
 Certificate chain length: 1
 Certificate[1]:
 Owner: CN=osso2.idp-example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
 Issuer: CN=osso2.idp-example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
 Serial number: 47f6a587
 Valid from: Fri Aug 04 15:02:47 PDT 2008 until: Thu Nov 03 15:02:47 PDT 2008
 Certificate fingerprints:
  MD5:  62:0E:5E:EB:8A:73:B2:F9:08:83:05:C5:DC:07:3C:E1
  SHA1: D4:9C:BA:25:4C:B5:71:20:CF:F3:18:46:AF:2E:7F:71:2A:4B:BD:B3

The certificate indicated by the alias "opensso-idp-2" is a
self-signed certificate.
```

Note – The output of this command may list more than one certificate based on the entries in the keystore.

c. Generate a server certificate request.

```
# keytool -certreq -alias opensso-idp-2 -keypass domain2master
-keystore keystore.jks -storepass domain2master file opensso-idp-2.csr
```

`opensso-idp-2.csr` is the server certificate request.

d. (Optional) Verify that `opensso-idp-2.csr` was created.

```
# ls -la opensso-idp-2.csr

 -rw-r--r--   1 osso80adm staff        715 Apr  4 15:04 opensso-idp-2.csr
```

e. Send `opensso-idp-2.csr` to the CA of your choice.

The CA issues and returns a certified server certificate named `opensso-idp-2.cer`.

f. Import `ca.cer`, the CA root certificate, into the certificate store.

The root certificate must be imported into two keystores (`keystore.jks` and `cacerts.jks`) with Application Server.

```
# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore keystore.jks -storepass domain2master

Owner: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore

# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore cacerts.jks -storepass domain2master

Owner: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore
```

g. **Replace the self-signed public key certificate (associated with the `s1as` alias) with the server certificate received from the CA.**

```
# keytool -import -file opensso-idp-2.cer -alias opensso-idp-2
-keystore keystore.jks -storepass domain2master

Certificate reply was installed in keystore
```

h. **(Optional) Verify that the self-signed public key certificate has been overwritten by the server certificate received from the CA.**

```
# keytool -list -v -keystore keystore.jks
-storepass domain2master

The certificate indicated by the alias "opensso-idp-2" is signed by CA.
```

i. **Change the certificate alias from the default `s1as` to the new `opensso-idp-2` in the `domain.xml` file for the `ossodomain` domain.**

The Application Server configuration file is `domain.xml`.

```
<http-listener acceptor-threads="1" address="0.0.0.0"
blocking-enabled="false" default-virtual-server="server" enabled="true"
family="inet" id="http-listener-2" port="1081" security-enabled="true"
server-name="" xpowered-by="true">
<ssl cert-nickname="opensso-idp-2" client-auth-enabled="false" ssl2-enabled="false"
ssl3-enabled="true" tls-enabled="true" tls-rollback-enabled="true"/>
```

Tip – Backup `domain.xml` before modifying it.

---

11 **Modify the JVM options in your web container's configuration file using the following sub-procedure.**

OpenSSO Enterprise is deployed with an embedded configuration data store (if desired). In order for the configuration data store to be created successfully, the following JVM options should be modified in the web container's configuration file. We will be modifying `domain.xml` again for this example.

Tip – Backup `domain.xml` before modifying it.

---

a. **Change to the `config` directory.**

```
# cd /export/osso80adm/domains/ossodomain/config
```

b. **Open `domain.xml` in a text editor and make the following changes:**

- Replace **`<jvm-options>-client</jvm-options>`** with **`<jvm-options>-server</jvm-options>`**.

- Replace **`<jvm-options>-Xmx512m</jvm-options>`** with **`<jvm-options>-Xmx1024m</jvm-options>`**.

c. **Save the file and close it.**

12 **Restart the `ossodomain` domain.**

```
# cd /export/osso80adm/domains/ossodomain/bin
# ./stopserv

Server was successfully stopped.

./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

13 **Verify that the certificate used for SSL communication is the root CA certificate.**

a. **Access `https:///osso2.idp-example.com:1081/index.html` from a web browser.**

b. **View the details of the certificate in the security warning to ensure that it is Issued by "OpenSSLTestCA".**

After inspecting and accepting the certificate, you should see the default `index.html` page.

c. **Close the browser.**

14 **Log out of the `/osso2.idp-example.com` host machine.**

5.2 Configuring the OpenSSO Enterprise Load Balancer

The two instances of OpenSSO Enterprise are fronted by one load balancer (Load Balancer 2). Users will access OpenSSO Enterprise through the secure port 1081. Load Balancer 2 sends the user and agent requests to the server where the session originated. Secure Sockets Layer (SSL) is terminated and regenerated before a request is forwarded to the OpenSSO Enterprise servers to allow the load balancer to inspect the traffic for proper routing. Load Balancer 2 is capable of the following types of load balancing:

Cookie-based
The load balancer makes decisions based on client's cookies. The load balancer looks at the request and detects the presence of a cookie by a specific name. If the cookie is detected in the request, the load balancer routes the request to the specific server to which the cookie has been assigned. If the cookie is not detected in the request, the load balancer balances client requests among the available servers.

IP-based
This is similar to cookie-based load balancing, but the decision is based on the IP address of the client. The load balancer sends all requests from a specific IP address to the same server.

TCP
The load balancer mainstreams session affinity. This means that all requests related to a TCP session, are forwarded to the same server. In this deployment example, Load Balancer 2 forwards all requests from a single client to exactly the same server. When the session is started and maintained by one client, session affinity is guaranteed. This type of load-balancing is applicable to the TCP-based protocols.

This section assumes that you have already installed a load balancer. Before you begin, note the following:

- The load balancer hardware and software used in the lab facility for this deployment is BIG-IP® manufactured by F5 Networks. If you are using different load balancer software, see the documentation that comes with that product for detailed settings information.
- Contact your network administrator to obtain an available virtual IP address for the load balancer you want to configure.
- Know the IP address of the load balancer hardware, the URL for the load balancer login page, and a username and password for logging in to the load balancer application.
- Get the IP addresses for OpenSSO Enterprise 1 and OpenSSO Enterprise 2 by running the following command on each host machine:

```
# ifconfig -a
```

Use the following list of procedures as a checklist for completing the task.

1. “To Request a Certificate for OpenSSO Enterprise Load Balancer 2” on page 96
2. “To Install the Certificate Authority Root Certificate to OpenSSO Enterprise Load Balancer 2” on page 97
3. “To Install the Server Certificate to OpenSSO Enterprise Load Balancer 2” on page 97
4. “To Configure OpenSSO Enterprise Load Balancer 2” on page 98
5. “To Create an SSL Proxy for SSL Termination at the OpenSSO Enterprise Load Balancer 2” on page 101

▼ To Request a Certificate for OpenSSO Enterprise Load Balancer 2

You should already have a root certificate from the CA of your choice. Generate a request for a server certificate to send to the CA. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1. **Access `https://is-f5.example.com`, the BIG-IP load balancer login page, in a web browser.**

2. **Log in to the BIG-IP console as administrator.**

3. **Click *Configure your BIG-IP (R) using the Configuration Utility*.**

4. **In the left pane, click Proxies.**

5. **Click the Cert-Admin tab.**

6. **On the SSL Certificate Administration page, click *Generate New Key Pair/Certificate Request*.**

7. **In the Create Certificate Request page, provide the following information.**

Key Identifier:	**`lb2.idp-example.com`**
Organizational Unit Name:	**`Deployment`**
Domain Name:	**`lb2.idp-example.com`**
Challenge Password:	**`password`**
Retype Password:	**`password`**

8. **Click *Generate Key Pair/Certificate Request*.**

   On the SSL Certificate Request page, the request is generated in the Certificate Request field.

9. **Save the text contained in the Certificate Request field to a file named `lb-2.csr`.**

10. **Log out of the console and close the browser.**

11. **Send `lb-2.csr` to the CA of your choice.**

    The CA issues and returns a certified server certificate named `lb-2.cer`.

▼ To Install the Certificate Authority Root Certificate to OpenSSO Enterprise Load Balancer 2

You should already have a root certificate from the CA of your choice. Install the CA root certificate on Load Balancer 2 to ensure that a link between it and the CA can be maintained. Use the same root certificate that you imported in "4.4 Enabling Secure Communication for the Directory Server User Data Instances" on page 60. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1. **Access `https://is-f5.example.com`, the BIG-IP load balancer login page, in a web browser.**
2. **Log in to the BIG-IP console as administrator.**
3. **In the BIG-IP load balancer console, click Proxies.**
4. **Click the Cert-Admin tab.**
5. **Click Import.**
6. **In the Import Type field, choose Certificate, and click Continue.**
7. **Click Browse in the Certificate File field on the Install SSL Certificate page.**
8. **In the Choose File dialog, choose Browser.**
9. **Navigate to `ca.cer` and click Open.**
10. **In the Certificate Identifier field, enter `OpenSSL_CA_cert`.**
11. **Click Install Certificate.**
12. **On the Certificate OpenSSL_CA_Cert page, click Return to Certificate Administration.**

    The root certificate named `OpenSSL_CA_Cert` is now included in the Certificate ID list.

▼ To Install the Server Certificate to OpenSSO Enterprise Load Balancer 2

Before You Begin This procedure assumes you have received the server certificate requested in "To Request a Certificate for OpenSSO Enterprise Load Balancer 2" on page 96 and just completed "To Install the Certificate Authority Root Certificate to OpenSSO Enterprise Load Balancer 2" on page 97.

1 **In the BIG-IP load balancer console, click Proxies.**

2 **Click the Cert-Admin tab.**

The key `lb2.idp-example.com` is in the Key List.

3 **In the Certificate ID column, click Install for `lb2.idp-example.com`.**

4 **In the Certificate File field, click Browse.**

5 **In the Choose File dialog, navigate to `lb-2.cer`, the server certificate, and click Open.**

6 **Click Install Certificate.**

7 **On the Certificate lb2.idp-example.com page, click Return to Certificate Administration Information.**

Verify that the Certificate ID indicates `lb2.idp-example.com` on the SSL Certificate Administration page.

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

▼ To Configure OpenSSO Enterprise Load Balancer 2

1 **Access `https://is-f5.example.com`, the BIG-IP load balancer login page, in a web browser.**

2 **Log in to the BIG-IP console as administrator.**

3 **Click *Configure your BIG-IP (R) using the Configuration Utility*.**

4 **Create a Pool.**

A pool contains all the backend server instances.

a. **In the left pane, click Pools.**

b. **On the Pools tab, click Add.**

c. **In the Add Pool dialog, provide the following information.**

Pool Name	**`OpenSSO-IDP-Pool`**
Load Balancing Method	Round Robin
Resources	Add the IP addresses and port numbers for both OpenSSO Enterprise host machines.

Note – User port number `1081`.

---

d. **Click Done.**

5 **Add a Virtual Server.**

The virtual server presents an address to the outside world and, when users attempt to connect, it would forward the connection to the most appropriate real server.

Note – If you encounter JavaScript™ errors or otherwise cannot proceed to create a virtual server, try using Internet Explorer.

---

a. **In the left frame, click Virtual Servers.**

b. **On the Virtual Servers tab, click Add.**

c. **In the Add a Virtual Server dialog box, provide the following information:**

Address	Enter the IP address for `lb2.idp-example.com`
Service	**`1082`**

d. **Continue to click Next until you reach the Pool Selection dialog box.**

e. **In the Pool Selection dialog box, assign the `OpenSSO-IDP-Pool` Pool.**

f. **Click Done.**

6 **Add Monitors.**

OpenSSO Enterprise comes with a JSP file named `isAlive.jsp` that can be contacted to determine if the server is down. Since we have not yet deployed OpenSSO Enterprise, `isAlive.jsp` cannot be used. In the following sub procedure, create a custom monitor that periodically accesses the Application server instance(s). If desired, the monitor can be changed later to use `isAlive.jsp`.

a. **Click the Monitors tab**

b. **Click the Basic Associations tab**

c. **Find the IP address for `osso1.idp-example.com:1081` and `osso2.idp-example.com:1081`.**

d. **Mark the Add checkbox for `OSSO1` and `OSSO2`.**

e. **At the top of the Node column, choose the `tcp` monitor.**

f. **Click Apply.**

7 **Configure the load balancer for persistence.**

a. **In the left pane, click Pools.**

b. **Click the name of the pool you want to configure; in this case, `OpenSSO-IDP-Pool`.**

c. **Click the Persistence tab.**

d. **Under Persistence Type, select Passive HTTP Cookie.**

e. **Under Cookie Name, type `amlbcookie`.**

f. **Click Apply.**

8 **In the left pane, click BIGpipe.**

9 **In the BIGpipe command window, type the following:**

`makecookie` *ip-address*:*port*

ip-address is the IP address of the `osso1.idp-example.com` host machine and *port* is the same machine's port number; in this case, 1081.

10 **Press Enter to execute the command.**

Something similar to `Set-Cookie: BIGipServer[poolname]=692589248.22222.0000; path=/` is displayed. Save the numbered value (in this case, 692589248.22222.0000) for use in "To Create a Site on OpenSSO Enterprise 1" on page 112.

11 **In the left pane, click BIGpipe again.**

12 **In the BIGpipe command window, type the following:**

`makecookie` *ip-address*:*port*

ip-address is the IP address of the `osso2.idp-example.com` host machine and *port* is the same machine's port number; in this case, 1081.

13 **Press Enter to execute the command.**

Something similar to `Set-Cookie: BIGipServer[poolname]=692589248.33333.0000; path=/` is displayed. Save the numbered value (in this case, 692589248.33333.0000) for use in "To Create a Site on OpenSSO Enterprise 1" on page 112.

14 **Log out of the load balancer console.**

To Create an SSL Proxy for SSL Termination at the OpenSSO Enterprise Load Balancer 2

SSL communication is terminated at Load Balancer 2. The request is then re-encrypted and securely forwarded to OpenSSO Enterprise. When clients send an SSL-encrypted request to Load Balancer 2, it decrypts the request and re-encrypts it before sending it on to the OpenSSO Enterprise SSL port. Load Balancer 2 also encrypts the responses it receives back from OpenSSO Enterprise, and sends these encrypted responses back to the client. Towards this end create an *SSL proxy* for SSL termination and regeneration.

Before You Begin You should have a root certificate issued by a recognized CA.

1 **Access** `https://is-f5.example.com`**, the BIG-IP load balancer login page, in a web browser.**

2 **Log in to the BIG-IP console as administrator.**

3 **Click** *Configure your BIG-IP (R) using the Configuration Utility*.

4 **In the left pane, click Proxies.**

5 **Under the Proxies tab, click Add.**

6 **In the Add Proxy dialog, provide the following information.**

Proxy Type:	Check the SSL and ServerSSL checkbox.
Proxy Address:	The IP address of Load Balancer 2.
Proxy Service:	**`1081`** The secure port number
Destination Address:	The IP address of Load Balancer 2.
Destination Service:	**`1082`** The non-secure port number
Destination Target:	Choose **`Local Virtual Server`**.
SSL Certificate:	Choose **`lb2.idp-example.com`**.
SSL Key:	Choose **`lb2.idp-example.com`**.
Enable ARP:	Check this checkbox.

7 **Click Next.**

8 **On the page starting with "Insert HTTP Header String," change to Rewrite Redirects and choose Matching.**

9 **Click Next.**

10 **On the page starting with "Client Cipher List String", accept the defaults.**

11 **Click Next.**

12 **On the page starting with "Server Chain File," change to Server Trusted CA's File, select "OpenSSL_CA_Cert.crt" from the drop-down list.**

13 **Click Done.**

The new proxy server is added to the Proxy Server list.

14 **Log out of the load balancer console.**

15 **Access** `https://lb2.idp-example.com:1081/index.html` **from a web browser.**

If the Application Server index page is displayed, you can access it using the new proxy server port number and the load balancer is configured properly.

Tip – A message may be displayed indicating that the browser doesn't recognize the certificate issuer. If this happens, install the CA root certificate in the browser so that the browser recognizes the certificate issuer. See your browser's online help system for information on installing a root CA certificate.

---

16 **Close the browser.**

5.3 Deploying and Configuring OpenSSO Enterprise 1 and OpenSSO Enterprise 2

An OpenSSO Enterprise WAR will be deployed in the installed Application Server containers on both the OpenSSO Enterprise host machines. Additionally, you will configure the deployed applications. Use the following list of procedures as a checklist for completing the tasks.

1. "To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine" on page 103
2. "To Deploy the OpenSSO Enterprise WAR as OpenSSO Enterprise 1" on page 105

3. "To Copy the OpenSSO Enterprise WAR to the OpenSSO Enterprise 2 Host Machine" on page 106
4. "To Deploy the OpenSSO Enterprise WAR File as OpenSSO Enterprise 2" on page 107
5. "To Configure OpenSSO Enterprise 1" on page 108
6. "To Configure OpenSSO Enterprise 2" on page 110

▼ To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine

1 Log in to the `osso1.idp-example.com` host machine as root user.

2 Create a directory into which the OpenSSO Enterprise ZIP file can be downloaded and change into it.

```
# mkdir /export/OSSO_BITS
# cd /export/OSSO_BITS
```

3 Download the OpenSSO Enterprise ZIP file from `http://www.sun.com/download/`.

4 Unzip the downloaded file.

```
# unzip opensso_enterprise_80.zip
# cd /export/OSSO_BITS/opensso
# ls -al

total 66
        drwxr-xr-x  14 root     root         512 Jul 21 20:54 .
        drwxr-xr-x   3 root     root         512 Aug  5 16:49 ..
        -rw-r--r--   1 root     root         959 Jul 21 20:22 README
        drwxr-xr-x   6 root     root         512 Jul 21 20:58 deployable-war
        drwxr-xr-x   2 root     root         512 Jul 21 20:54 docs
        drwxr-xr-x   2 root     root         512 Jul 21 20:54 fedlet
        drwxr-xr-x   3 root     root         512 Jul 21 20:22 integrations
        drwxr-xr-x   2 root     root         512 Jul 21 20:54 ldif
        drwxr-xr-x   4 root     root         512 Jul 21 20:54 libraries
        -rw-r--r--   1 root     root       17003 Jul 21 20:22 license.txt
        drwxr-xr-x   2 root     root         512 Jul 21 20:54 migration
        drwxr-xr-x   2 root     root         512 Jul 21 20:54 patches
        drwxr-xr-x   2 root     root         512 Jul 21 20:54 samples
        drwxr-xr-x   3 root     root         512 Jul 21 20:58 tools
        drwxr-xr-x   8 root     root         512 Jul 21 20:32 upgrade
        drwxr-xr-x   2 root     root        2048 Jul 21 20:22 xml
```

5 Switch to the non-root user.

```
# su osso80adm
```

6 **Create a staging area in the non-root user directory into which the WAR will be exploded.**

```
# cd /export/osso80adm
# mkdir osso-staging
```

Tip – In the staging area, after exploding the WAR, you can modify the WAR contents to suit your needs, generate a new WAR, and deploy it on any number of remote host computers. Whenever you need to make changes to the WAR, you maintain the changes in this one staging area, and redeploy the modified WAR as many times as you want, on as many host machines as you need.

---

7 **Explode the WAR file.**

```
# cd osso-staging
# jar xvf /export/OSSO_BITS/opensso/deployable-war/opensso.war
```

8 **Make the following modifications to the `bootstrap.properties` file.**

By default, during the WAR deployment, OpenSSO Enterprise creates a bootstrap file in the user's home directory. The `bootstrap.properties` file points to the directory where all the OpenSSO Enterprise configurations will be created. With these modifications, OpenSSO Enterprise will create the bootstrap file in the directory you specify; in this case, `/export/osso80adm/config`. `bootstrap.properties` is located in `/export/osso80adm/osso-staging/WEB-INF/classes`.

- Uncomment the line that reads `#configuration.dir=`.
- Add the following value to the `configuration.dir=` property so it reads as follows.

  ```
  configuration.dir=/export/osso80adm/config
  ```

9 **Regenerate the WAR.**

```
# cd /export/osso80adm/osso-staging
# jar cvf ../opensso.war *
```

A new WAR file is created, including the modified `bootstrap.properties`.

10 **Verify that the new WAR was created in the proper location and with the appropriate permissions.**

```
# cd /export/osso80adm
# ls -al

total 130552
drwxr-xr-x   7 osso80adm staff        512 Aug 5 13:44 .
drwxr-xr-x  12 root      sys          512 Aug 5 11:11 ..
-rw-------   1 osso80adm staff        779 Aug 5 14:56 .asadmintruststore
drwx------   2 osso80adm staff        512 Aug 5 14:44 .gconf
```

```
drwx------   2 osso80adm staff       512 Aug 5 14:44 .gconfd
-rw-r--r--   1 osso80adm staff       144 Aug 5 17:02 .profile
drwx------   3 osso80adm staff       512 Aug 5 11:20 .sunw
drwxr-xr-x   3 osso80adm staff       512 Aug 5 14:55 domains
drwxr-xr-x  21 osso80adm staff      1024 Aug 5 13:43 osso-staging
-rw-r--r--   1 osso80adm staff  68884903 Aug 5 13:45 opensso.war
-rw-r--r--   1 osso80adm staff       136 Aug 5 17:02 local.cshrc
-rw-r--r--   1 osso80adm staff       157 Aug 5 17:02 local.login
-rw-r--r--   1 osso80adm staff       174 Aug 5 17:02 local.profile
```

Note – The opensso.war file is owned by osso80adm.

▼ To Deploy the OpenSSO Enterprise WAR as OpenSSO Enterprise 1

Before You Begin This procedure assumes you have just completed "To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine" on page 103 and are still logged into the osso1.idp-example.com host machine

1 **On the osso1.idp-example.com host machine, switch to the non-root user osso80adm.**

```
# su osso80adm
```

2 **Start the ossodomain domain.**

```
# cd /export/osso80adm/domains/ossodomain/bin
# ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

3 **Run asadm deploy to deploy the OpenSSO Enterprise WAR.**

```
# cd /opt/SUNWappserver91/bin
# ./asadm deploy --user domain2adm --host osso1.idp-example.com
--port=8989 --contextroot opensso --name opensso --target server
/export/osso80adm/opensso.war

Please enter the admin password> domain2pwd

Command deploy executed successfully.
```

4 **List the contents of the `j2ee-modules` directory to verify that the WAR file was successfully deployed.**

```
# cd /export/osso80adm/domains/ossodomain/applications/j2ee-modules
# ls -al

total 6
drwxr-xr-x   3 osso80adm staff      512 Aug 5 14:01 .
drwxr-xr-x   6 osso80adm staff      512 Aug 5 14:55 ..
drwxr-xr-x  21 osso80adm staff     1024 Aug 5 14:01 opensso
```

`opensso` exists in the directory and is owned by the non-root user `osso80adm`.

5 **Log out of the `osso1.idp-example.com` host machine.**

▼ To Copy the OpenSSO Enterprise WAR to the OpenSSO Enterprise 2 Host Machine

Before You Begin This procedure assumes you have completed "To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine" on page 103.

1 **Log in to the `osso2.idp-example.com` host machine as root user.**

2 **Switch to the non-root user `osso80adm`.**

```
# su osso80adm
```

3 **Change into the `osso80adm` directory.**

```
# cd /export/osso80adm
```

4 **Copy `opensso.war` from the `osso1.idp-example.com` host machine to the `osso80adm` directory.**

5 **Verify that the WAR file was copied into the proper location and with the appropriate permissions.**

```
# ls -al

total 130552
drwxr-xr-x   6 osso80adm staff        512 Aug 5 14:14 .
drwxr-xr-x   8 root      sys          512 Aug 5 10:54 ..
-rw-r--r--   1 osso80adm staff         70 Aug 5 14:13 .asadminpass
-rw-------   1 osso80adm staff        778 Aug 5 14:12 .asadmintruststore
drwx------   2 osso80adm staff        512 Aug 5 13:15 .gconf
drwx------   2 osso80adm staff        512 Aug 5 13:26 .gconfd
-rw-r--r--   1 osso80adm staff        144 Aug 5 15:00 .profile
drwx------   3 osso80adm staff        512 Aug 5 15:26 .sunw
```

```
drwxr-xr-x   3 osso80adm staff        512 Aug 5 14:12 domains
-rw-r--r--   1 osso80adm staff   68884903 Aug 5 14:14 opensso.war
-rw-r--r--   1 osso80adm staff        136 Aug 5 15:00 local.cshrc
-rw-r--r--   1 osso80adm staff        157 Aug 5 15:00 local.login
-rw-r--r--   1 osso80adm staff        174 Aug 5 15:00 local.profile
```

opensso.war is owned by osso80adm.

▼ To Deploy the OpenSSO Enterprise WAR File as OpenSSO Enterprise 2

Before You Begin This procedure assumes you have just completed "To Copy the OpenSSO Enterprise WAR to the OpenSSO Enterprise 2 Host Machine" on page 106 and are still logged into the osso2.idp-example.com host machine

1 On the osso2.idp-example.com host machine, switch to the non-root user osso80adm.

```
# su osso80adm
```

2 Start the ossodomain domain.

```
# cd /export/osso8/domains/ossodomain/bin
# ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

3 Run asadm deploy to deploy the OpenSSO Enterprise WAR file.

```
# cd /opt/SUNWappserver91/bin
# ./asadm deploy --user domain2adm --host osso2.idp-example.com
--port=8989 --contextroot opensso --name opensso --target server
/export/osso80adm/opensso.war

Please enter the admin password> domain2pwd

Command deploy executed successfully.
```

4 **List the contents of the `j2ee-modules` directory to verify that the WAR file was successfully deployed.**

```
# cd /export/osso80adm/domains/ossodomain/applications/j2ee-modules
# ls -al

total 6
drwxr-xr-x   3 osso80adm staff      512 Aug 5 14:01 .
drwxr-xr-x   6 osso80adm staff      512 Aug 5 14:55 ..
drwxr-xr-x  21 osso80adm staff     1024 Aug 5 14:01 opensso
```

`opensso` exists in the directory and is owned by the non-root user `osso80adm`.

5 **Log out of the `osso2.idp-example.com` host machine.**

▼ To Configure OpenSSO Enterprise 1

1 **Access `https://osso1.idp-example.com:1081/opensso` from a web browser.**

The OpenSSO Enterprise Configurator page is displayed for first time access.

2 **Select Create New Configuration under Custom Configuration on the Configurator page.**

The OpenSSO Enterprise Custom Configuration Wizard is displayed.

3 **Provide the following information for the Default User [amAdmin] in Step 1: General and click Next.**

Password	**`ossoadmin`**
Confirm	**`ossoadmin`**

4 **Accept the default values in Step 2: Server Settings and click Next**

5 **Do the following in Step 3: Configuration Store and click Next**

a. **Select First Instance.**

b. **Select Embedded (Open DS) as the configuration data store.**

c. **Accept the default values for the Port, Encryption Key, and Root Suffix fields.**

6 **Select Remote Directory in Step 4: User Store Settings, provide the following information and click Next**

SSL Enabled	Check the box.
Directory Name	**`lb1.idp-example.com`**

Port	**489**
Root Suffix	**dc=company,dc=com**
Password	**dsmanager**
Store Type	Select Generic LDAP.

7 **Select No in Step 5: Site Configuration and click Next.**

8 **Provide the following information for the Default Agent User [amldapuser] in Step 6: Default Agent User and click Next.**

Password	**agentuser**
Confirm	**agentuser**

9 **Click Create Configuration on the Summary page.**

The Configuration Complete page is displayed after configuration is completed.

10 **Click Proceed to Login on the Configuration Complete page.**

11 **Log in to the OpenSSO Enterprise console as the administrator.**

User Name:	**amadmin**
Password:	**ossoadmin**

If authentication succeeds and the OpenSSO Enterprise console is displayed, OpenSSO Enterprise has successfully accessed the embedded configuration data store.

12 **(Optional) To verify that the `config` directory and the supporting `bootstrap` directory have been created with the proper permissions, do the following.**

a. **Log in to the `osso1.idp-example.com` host machine as the root user.**

b. **Examine the file system.**

```
# cd /export/osso80adm
# ls -al

total 130556
drwxr-xr-x   8 osso80adm staff        512 Aug  6 19:32 .
drwxr-xr-x  14 root      sys          512 Aug  6 09:07 ..
-rw-r--r--   1 osso80adm staff         70 Mar 27 14:01 .asadminpass
-rw-------   1 osso80adm staff       1527 Aug  6 18:27 .asadmintruststore
drwx------   2 osso80adm staff        512 Mar 26 14:44 .gconf
drwx------   2 osso80adm staff        512 Mar 26 14:44 .gconfd
-rw-r--r--   1 osso80adm staff       1436 Apr  2 14:34 .keystore
```

```
-rw-r--r--   1 osso80adm staff        144 Mar 11 17:02 .profile
drwx------   3 osso80adm staff        512 Mar 24 11:20 .sunw
drwxr-xr-x   4 osso80adm staff        512 Aug  6 19:34 config
drwxr-xr-x   4 osso80adm staff        512 Aug  6 18:26 domains
drwxr-xr-x  21 osso80adm staff       1024 Aug  6 19:15 osso-staging
-rw-r--r--   1 osso80adm staff   68884903 Aug  6 19:17 opensso.war
-rw-r--r--   1 osso80adm staff        136 Mar 11 17:02 local.cshrc
-rw-r--r--   1 osso80adm staff        157 Mar 11 17:02 local.login
-rw-r--r--   1 osso80adm staff        174 Mar 11 17:02 local.profile
```

The `config` directory was created and is owned by non-root user `osso80adm`.

c. **Log out of the `osso1.idp-example.com` host machine.**

▼ To Configure OpenSSO Enterprise 2

1. **Access `https://osso2.idp-example.com:1081/opensso` from a web browser.**

   The OpenSSO Enterprise Configurator page is displayed for first time access.

2. **Select Create New Configuration under Custom Configuration on the Configurator page.**

   The OpenSSO Enterprise Custom Configuration Wizard is displayed.

3. **Provide the following information for the Default User [amAdmin] in Step 1: General and click Next.**

Password	**`ossoadmin`**
Confirm	**`ossoadmin`**

4. **Accept the default values in Step 2: Server Settings and click Next**

5. **Do the following in Step 3: Configuration Store and click Next**

   a. **Select Add to Existing Deployment as the configuration data store.**

   b. **Server URL: `https://osso2.idp-example.com:1081/opensso`**

   c. **Accept the default values for the ports.**

6. **Select No in Step 5: Site Configuration and click Next.**

7. **Click Create Configuration on the Summary page.**

   The Configuration Complete page is displayed after configuration is completed.

8 **Click Proceed to Login on the Configuration Complete page.**

9 **Log in to the OpenSSO Enterprise console as the administrator.**

User Name:	**amadmin**
Password:	**ossoadmin**

If authentication succeeds and the OpenSSO Enterprise console is displayed, OpenSSO Enterprise has successfully accessed the embedded configuration data store.

10 **(Optional) To verify that the `config` directory and the supporting `bootstrap` directory have been created with the proper permissions, do the following.**

a. **Log in to the `osso2.idp-example.com` host machine as the root user.**

b. **Examine the file system.**

```
# cd /export/osso80adm
# ls -al

total 130556
drwxr-xr-x   8 osso80adm staff        512 Aug  6 19:32 .
drwxr-xr-x  14 root      sys          512 Aug  6 09:07 ..
-rw-r--r--   1 osso80adm staff         70 Mar 27 14:01 .asadminpass
-rw-------   1 osso80adm staff       1527 Aug  6 18:27 .asadmintruststore
drwx------   2 osso80adm staff        512 Mar 26 14:44 .gconf
drwx------   2 osso80adm staff        512 Mar 26 14:44 .gconfd
-rw-r--r--   1 osso80adm staff       1436 Apr  2 14:34 .keystore
-rw-r--r--   1 osso80adm staff        144 Mar 11 17:02 .profile
drwx------   3 osso80adm staff        512 Mar 24 11:20 .sunw
drwxr-xr-x   4 osso80adm staff        512 Aug  6 19:34 config
drwxr-xr-x   4 osso80adm staff        512 Aug  6 18:26 domains
drwxr-xr-x  21 osso80adm staff       1024 Aug  6 19:15 osso-staging
-rw-r--r--   1 osso80adm staff   68884903 Aug  6 19:17 opensso.war
-rw-r--r--   1 osso80adm staff        136 Mar 11 17:02 local.cshrc
-rw-r--r--   1 osso80adm staff        157 Mar 11 17:02 local.login
-rw-r--r--   1 osso80adm staff        174 Mar 11 17:02 local.profile
```

The `config` directory was created and is owned by non-root user `osso80adm`.

c. **Log out of the `osso2.idp-example.com` host machine.**

5.4 Configuring the OpenSSO Enterprise Platform Service

The Platform Service provides centralized configuration management for an OpenSSO Enterprise deployment. In this procedure, you configure the two instances of OpenSSO Enterprise to work as a single unit. Once configured as a *site*, all client requests go through the configured load balancer. Use the following list of procedures as a checklist for completing this task.

1. "To Create a Site on OpenSSO Enterprise 1" on page 201
2. "To Verify that the OpenSSO Enterprise Site was Configured Properly" on page 204

▼ To Create a Site on OpenSSO Enterprise 1

It is **not** necessary to repeat this procedure on OpenSSO Enterprise 2.

1 **Access** `https://osso1.idp-example.com:1081/opensso/console` **in a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

3 **Under the Configuration tab, click Servers and Sites.**

The Servers and Sites page is displayed.

4 **Click New under Sites.**

The New Site properties page is displayed.

5 **Enter the following values for the load balancer and click OK.**

Name	**`External`**
Primary URL	**`https://lb2.idp-example.com:1081/opensso`**

A new site called External is displayed in the Sites list.

6 **Click on the** `https://osso1.idp-example.com:1081/opensso` **server entry under the Servers list.**

The Edit `https://osso1.idp-example.com:1081/opensso` page is displayed.

7 **Assign External from the Parent Site drop down list and click Save.**

8 **Click the Advanced tab.**

9 **Enter the number generated for the `osso1.idp-example.com` host machine as the value of the `com.iplanet.am.lbcookie.value` property and click Save.**

The number was generated using the `makecookie` command in "To Configure OpenSSO Enterprise Load Balancer 2" on page 98.

10 **Click Back to Server and Sites.**

11 **Click on the `https://osso2.idp-example.com:1081/opensso` server entry under the Servers list.**

The Edit `https://osso2.idp-example.com:1081/opensso` page is displayed.

12 **Assign External from the Parent Site drop down list and click Save.**

13 **Click the Advanced tab.**

14 **Enter the number generated for the `osso2.idp-example.com` host machine as the value of the `com.iplanet.am.lbcookie.value` property and click Save.**

The number was generated using the `makecookie` command in "To Configure OpenSSO Enterprise Load Balancer 2" on page 98.

15 **Click Back to Server and Sites.**

Note – You should see External under the Site Name column for both servers.

---

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

17 **As a root user, log in to the `osso1.idp-example.com` host machine.**

18 **Restart the web container for the changes to take effect.**

```
# su osso80adm
# cd /export/osso80adm/domains/ossodomain/bin
# ./stopserv; ./startserv

Server was successfully stopped.

admin username:  domain2adm

admin password:  domain2pwd

master password: domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

19 **As a root user, log in to the `osso2.idp-example.com` host machine.**

20 **Restart the web container for the changes to take effect.**

```
# su osso80adm
# cd /export/osso80adm/domains/ossodomain/bin
# ./stopserv; ./startserv

Server was successfully stopped.

admin username:  domain2adm

admin password:  domain2pwd

master password: domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

21 **Log out of both OpenSSO Enterprise host machines.**

▼ To Verify that the OpenSSO Enterprise Site was Configured Properly

1 **Access the load balancer at `https://lb2.idp-example.com:1081/opensso/UI/Login`.**

If an error message is displayed indicating that the browser cannot connect to either `osso1.idp-example.com` or `osso2.idp-example.com`, the site configuration is not correct. If the site configuration is correct, all browser interactions will occur as expected.

2 **When the OpenSSO Enterprise login page is displayed, verify that the browser URL still contains the Primary Site URL configured for the load balancer.**

If it does not contain the Site URL, the site configuration is incorrect. If the site configuration is correct, all browser interactions will occur through the secure Site URL.

3 **Log in to the OpenSSO Enterprise console as the administrator.**

User Name:	**`amadmin`**
Password:	**`ossoadmin`**

A successful login occurs when the site configuration is correct.

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

5.5 Configuring OpenSSO Enterprise for SAML v2

Configure OpenSSO Enterprise on the identity provider side to recognize the Directory Server LDAP schema previously modified for SAML v2 attributes.

▼ To Configure OpenSSO Enterprise for the Modified LDAP Schema

Before You Begin This procedure assumes you have completed "4.3 Modifying the Directory Server Schema" on page 59.

1. **Access** `https://lb2.idp-example.com:1081/opensso/console` **from a web browser.**

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

   Username **`amadmin`**

   Password **`ossoadmin`**

   The Common Tasks tab is displayed.

3. **Click the Access Control tab and** `/ (Top-level Realm)` **on the Access Control page.**

4. **Click the Data Stores tab.**

5. **Under the Data Stores tab, click** `embedded`**.**

   The Generic LDAPv3 page is displayed.

6. **Add the following values to properties on the Generic LDAPv3 page.**

   - Type **`sunFMSAML2NameIdentifier`** in the New Value box of the LDAP User Object Class property and click Add.
   - Add the following values to the LDAP User Attribute property.
     - Type **`sun-fm-saml2-nameid-infokey`** in the New Value box and click Add.
     - Type **`sun-fm-saml2-nameid-info`** in the New Value box and click Add.

7. **Click Save on the Generic LDAPv3 page.**

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

CHAPTER 6

Configuring OpenSSO Enterprise Realms for User Authentication

This chapter contains instructions on configuring OpenSSO Enterprise to use the external user data store (set up in Chapter 4, "Installing Sun Java System Directory Server and Creating Instances for User Data") for authentication credentials. This is done by modifying the top-level realm or, alternately, configuring a sub realm for the external users and creating an authentication chain. Choose either of the sections listed to configure OpenSSO Enterprise for user authentication.

- "6.1 Modifying the Top-Level Realm for Test Users" on page 117
- "6.2 Creating and Configuring a Sub Realm for Test Users" on page 119

Caution – Do **not** do both.

6.1 Modifying the Top-Level Realm for Test Users

At this point in the deployment, the root realm (by default, `/ (Top Level Realm)`) is configured to authenticate special OpenSSO Enterprise accounts (for example, `amadmin` and agents) against the embedded configuration data store. Since the external user data store is an instance of Directory Server and not part of the embedded configuration data store, we modify the configuration details of the top-level realm to include the user data stores schema, allowing OpenSSO Enterprise to recognize users in the external user data store. Use the following list of procedures as a checklist for completing this task.

1. "To Modify the Top-Level Realm for User Authentication" on page 118
2. "To Verify that a User Can Successfully Authenticate" on page 119

▼ To Modify the Top-Level Realm for User Authentication

1 **Access** `https://osso1.idp-example.com:1081/opensso/console` **in a web browser.**

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

User Name:	**`amadmin`**
Password:	**`ossoadmin`**

3 **Click the Access Control tab.**

4 **Click** `/ (Top Level Realm)`**, the root realm, under the Access Control tab.**

5 **Click the Data Stores tab.**

The `embedded` data store link is displayed.

6 **Click** `embedded`**.**

The `Generic LDAPv3` properties page is displayed.

7 **On the** `Generic LDAPv3` **properties page, set the following attribute values and click Save.**

LDAP People Container Naming Attribute
Enter `ou`.

LDAP Groups Container Value
Enter `Groups`.

LDAP Groups Container Naming Attribute
Enter `ou`.

LDAP People Container Value
Enter `users`.

Note – If this field is empty, the search for user entries will start from the root suffix.

8 **Click Back to Data Stores.**

9 **(Optional) Click the Subjects tab to verify that the test users are now displayed.**

`idpuser` is displayed under Users (as well as others created during OpenSSO Enterprise configuration).

10 **Click the Authentication tab.**

11 **Click the Advanced Properties link under General.**

The Core Realm Attributes page is displayed.

12 **Change the value of User Profile to Ignored.**

This new value specifies that a user profile is not required by the Authentication Service in order to issue a token after successful authentication. This modification is specific to this deployment example because the OpenSSO Enterprise schema and the Directory Server schema have not been mapped.

13 **Click Save.**

14 **Click Back to Authentication.**

15 **Click Back to Access Control.**

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

▼ To Verify that a User Can Successfully Authenticate

You should be able to log in successfully as the test user.

1 **Access `https://osso1.idp-example.com:1081/opensso/UI/Login` in a web browser.**

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

User Name: **`idpuser`**

Password: **`idpuser`**

You should be able to log in successfully and see a page with a message that reads *You're logged in*. Since the User Profile attribute was previously set to `Ignored`, the user's profile is not displayed after a successful login. If the login is not successful, watch the Directory Server access log to troubleshoot the problem.

6.2 Creating and Configuring a Sub Realm for Test Users

At this point in the deployment, `/ (Top Level Realm)`, the root realm, is configured to authenticate special OpenSSO Enterprise accounts (for example, `amadmin` and agents) against the embedded configuration data store. Since the external user data store is an instance of Directory Server and not part of the embedded configuration data store, we create a sub realm and modify the configuration details to include the external user data stores schema, allowing OpenSSO Enterprise to recognize users in the Directory Server instances. The sub realm creates a demarcation between OpenSSO Enterprise configuration and administrative data and the user data. Use the following list of procedures as a checklist for completing this task.

1. "To Create a Sub Realm" on page 120
2. "To Change the User Profile Configuration for the Sub Realm" on page 121
3. "To Modify the Sub Realm for User Authentication" on page 121
4. "To Verify That the Sub Realm Can Access the External User Data Store" on page 122
5. "To Verify That the Sub Realm Subjects Can Successfully Authenticate" on page 123

▼ To Create a Sub Realm

When a sub realm is created it inherits configuration data (including which user data store to access) from the root realm (by default, `/ (Top Level Realm)`) and uses said data to authenticate users. The user data store can be modified per sub realm. In this deployment, we use the inherited `Generic LDAPv3` data store.

1 Access `https://osso1.idp-example.com:1081/opensso/console` **from a web browser.**

2 Log in to the OpenSSO Enterprise console as the administrator.

User Name: **`amadmin`**

Password: **`ossoadmin`**

3 Click the Access Control tab.

4 Click New to create a new realm.

The New Realm page is displayed.

5 Set the following attribute values on the New Realm page.

Name

Enter **`users`**.

Realm/DNS Aliases

Enter `users` in the New Value field and click Add.

6 Click OK.

The `users` realm is listed as a sub realm of `/ (Top Level Realm)`, the root realm.

▼ To Change the User Profile Configuration for the Sub Realm

Before You Begin This procedure assumes you have just completed "To Create a Sub Realm" on page 120 and are still logged in to the OpenSSO Enterprise console.

1. **Under the Access Control tab, click the `users` realm.**

2. **Click the Authentication tab.**

3. **Click the Advanced Properties link under General.**

   The Core Realm Attributes page is displayed.

4. **Change the value of User Profile to Ignored.**

   This new value specifies that a user profile is not required by the Authentication Service in order to issue a token after successful authentication.

5. **Click Save.**

6. **Click Back to Access Control.**

▼ To Modify the Sub Realm for User Authentication

Before You Begin This procedure assumes you have just completed "To Change the User Profile Configuration for the Sub Realm" on page 121 and are still logged in to the OpenSSO Enterprise console.

1. **Click `users`, the sub realm, under the Access Control tab.**

2. **Click the Data Stores tab.**

   The `embedded` data store link is displayed.

3. **Click `embedded`.**

   The `Generic LDAPv3` properties page is displayed.

4. **On the `Generic LDAPv3` properties page, set the following attribute values and click Save.**

   LDAP People Container Naming Attribute
   Enter `ou`.

   LDAP Groups Container Value
   Enter `Groups`.

   LDAP Groups Container Naming Attribute
   Enter `ou`.

LDAP People Container Value

Enter `users`.

Note – If this field is empty, the search for user entries will start from the root suffix.

---

5 **Click Back to Data Stores.**

6 **(Optional) Click the Subjects tab to verify that the test users are now displayed.**

`idpuser` is displayed under Users (as well as others created during OpenSSO Enterprise configuration).

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

▼ To Verify That the Sub Realm Can Access the External User Data Store

This optional procedure is to verify the modifications.

1 **Access** `https://osso1.idp-example.com:1081/opensso/console` **from a web browser.**

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

User Name: **`amadmin`**

Password: **`ossoadmin`**

3 **Click on the Access Control tab**

4 **Click on the** `users` **sub realm.**

5 **Click on the Subjects tab.**

`spuser` is displayed under Users.

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

▼ To Verify That the Sub Realm Subjects Can Successfully Authenticate

1 **Access** `https://osso1.idp-example.com:1081/opensso/UI/Login?realm=users` **from a web browser.**

The parameter `realm=users` specifies the realm to use for authentication. At this point, a user can log in against Directory Server only if the `realm` parameter is defined in the URL.

2 **Log in to OpenSSO Enterprise with as a test user.**

User Name	**`idpuser`**
Password	**`idpuser`**

You should be able to log in successfully and see a page with a message that reads *You're logged in*. Since the User Profile attribute was set to `Ignored`, the user's profile is not displayed after a successful login. If the login is not successful, watch the Directory Server access log to troubleshoot the problem.

PART III

Building the Service Provider Environment

This third part of *Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0* provides the instructions for installing and configuring OpenSSO Enterprise, Sun Java System Directory Server, applicable web containers and policy agents to function as the service provider. Best results will be obtained by executing the tasks in the exact sequence in which they are presented. This part contains the following chapters:

- Chapter 7, “Installing Sun Java System Directory Server and Creating Instances for User Data”
- Chapter 8, “Deploying and Configuring OpenSSO Enterprise”
- Chapter 9, “Configuring OpenSSO Enterprise Realms for User Authentication”
- Chapter 10, “Configuring the Service Provider Protected Resource Host Machine”

Caution – If deviating from the task sequence or details described, refer to the relevant product documentation for information or necessary requirements.

CHAPTER 7

Installing Sun Java System Directory Server and Creating Instances for User Data

This chapter contains instructions for installing Sun Java™ System Directory Server and creating the instances in which Sun OpenSSO Enterprise user data will be stored. Additionally, the procedure for enabling multi-master replication between the two instances and the procedure for configuring the user data load balancer are included. This chapter contains the following sections:

- "7.1 Installing and Configuring Directory Server 1 and Directory Server 2" on page 127
- "7.2 Enabling Multi-Master Replication of the User Data Instances" on page 137
- "7.3 Modifying the Directory Server Schema" on page 144
- "7.4 Enabling Secure Communication for the Directory Server User Data Instances" on page 146
- "7.5 Configuring the Directory Server Load Balancer" on page 149

Note – If you have an existing user data store, you can go directly to the instructions in Chapter 8, "Deploying and Configuring OpenSSO Enterprise."

7.1 Installing and Configuring Directory Server 1 and Directory Server 2

This section contains the instructions for installing Directory Server on two different host machines and creating the directory instances named `sp-users` in which the user data will be stored. Use the following list of procedures as a checklist for completing the task.

1. "To Download the Directory Server Bits and Required Patches to the Directory Server Host Machines" on page 128
2. "To Patch the Directory Server Host Machines" on page 130
3. "To Install Directory Server 1" on page 131
4. "To Create a User Data Instance on Directory Server 1" on page 132
5. "To Create a Base Suffix for the User Data Instance on Directory Server 1" on page 133

6. “To Install Directory Server 2” on page 134
7. “To Create a User Data Instance on Directory Server 2” on page 135
8. “To Create a Base Suffix for the User Data Instance on Directory Server 2” on page 136

▼ To Download the Directory Server Bits and Required Patches to the Directory Server Host Machines

Use this procedure to download the Directory Server Enterprise Edition (EE) 6.3 bits and the required system patches to both the Directory Server 1 host machine (`ds1.sp-example.com`) and the Directory Server 2 host machine (`ds2.sp-example.com`).

1 **Access** `http://www.sun.com/software/products/directory_srvr_ee/get.jsp` **from a web browser and click Download Now.**

2 **Provide the following information in the** ***Select product configuration*** **section and click View Downloads.**

Step 1: Select Component	**`Directory Server Enterprise Edition 6.x`**
Step 2: Select Version	**`6.3`**
Step 3: Select Delivery Type	**`Compress Archive (ZIP)`**
Step 4: Select Platform	Choose the platform you are using.

The Selection Results page will be displayed with links to the download sites for the Directory Server bits and required patches.

Note – The patch numbers generated for download on the Selection Results page are based on your input. Check the most recent Directory Server Enterprise Edition 6.1 Release Notes to determine if you need to install other patches based on your machine's architecture and operating system. In this deployment, the Release Notes indicate that based on the hardware and operating system being used, patch 118855, patch 127112, patch 119964, patch 125379, and patch 119255 are required.

---

3 **Log into the** `ds1.sp-example.com` **host machine as a root user.**

4 **Run** `patchadd` **to see if the patches are already installed.**

See the `patchadd` man page for more information.

```
# /usr/sbin/patchadd -p | grep 118855
```

No results are returned which indicates that the patch is not yet installed on the system.

```
# /usr/sbin/patchadd -p | grep 127112
```

No results are returned which indicates that the patch is not yet installed on the system.

```
# /usr/sbin/patchadd -p | grep 119964
```

No results are returned which indicates that the patch is not yet installed on the system.

```
# /usr/sbin/patchadd -p | grep 125379
```

No results are returned which indicates that the patch is not yet installed on the system.

```
# /usr/sbin/patchadd -p | grep 119255
```

No results are returned which indicates that the patch is not yet installed on the system.

Note – If the necessary patches are already installed on your machine, proceed to step 7.

---

5 **Make a directory for the patch downloads and change into it.**

```
# mkdir /export/patches
# cd /export/patches
```

6 **Download the patches.**

You can click on the patch links from the Selection Results page or search for patches directly at `http://sunsolve.sun.com`. If searching directly, navigate to the PatchFinder page and enter the patch number. For each patch you are downloading, click the HTTP link beside the heading *Download Signed Patch (xxx bytes)*.

Note – Signed patches are downloaded as JAR files. Unsigned patches are downloaded as ZIP files. In this step, ZIP files are downloaded.

---

7 **Make a directory for the Directory Server download and change into it.**

```
# mkdir /export/DS63
# cd /export/DS63
```

8 **Download the *Base Full Install* of Directory Server EE 6.3 — Zip Distribution, Multi-Language, (DS/DPS/DE/ISW/DSRK) bits.**

Note – No Directory Server Administration Console is installed with these bits. This deployment example uses the command line to configure the software.

---

9 **Log out of the `ds1.sp-example.com` host machine.**

10 **Repeat this same procedure on the `ds2.sp-example.com` host machine.**

▼ To Patch the Directory Server Host Machines

If necessary, use this procedure to patch both the `ds1.sp-example.com` host machine and the `ds2.sp-example.com` host machine.

1 **Log in to the `ds1.sp-example.com` host machine as a root user.**

2 **Change into the directory that contains the downloaded patch files.**

```
# cd /export/patches
```

3 **Unzip the patch files.**

```
# unzip 118855.zip
# unzip 127112.zip
# unzip 119964.zip
# unzip 125379.zip
# unzip 119255.zip
```

4 **Install the patches.**

```
# /usr/sbin/patchadd /export/patches/118855
# /usr/sbin/patchadd /export/patches/127112
# /usr/sbin/patchadd /export/patches/119964
# /usr/sbin/patchadd /export/patches/125379
# /usr/sbin/patchadd /export/patches/119255
```

Tip – You can use the `-M` option to install all patches at once. See the `patchadd` man page for more information.

5 **Reboot your machine, if requested.**

6 **After installation is complete, verify that each patch was added successfully.**

```
# /usr/sbin/patchadd -p | grep 118855
```

A series of patch numbers are displayed, and the patch 118855 is present.

```
# /usr/sbin/patchadd -p | grep 127112
```

A series of patch numbers are displayed, and the patch 127112 is present.

```
# /usr/sbin/patchadd -p | grep 119964
```

A series of patch numbers are displayed, and the patch 119964 is present.

```
# /usr/sbin/patchadd -p | grep 125379
```

A series of patch numbers are displayed, and the patch 125379 is present.

```
# /usr/sbin/patchadd -p | grep 119255
```

A series of patch numbers are displayed, and the patch 119255 is present.

7 **Log out of the `ds1.sp-example.com` host machine.**

8 **Repeat this same procedure on the `ds2.sp-example.com` host machine.**

▼ To Install Directory Server 1

Before You Begin This procedures assumes "To Download the Directory Server Bits and Required Patches to the Directory Server Host Machines" on page 128 and "To Patch the Directory Server Host Machines" on page 130 have been completed.

1 **Log in to the `ds1.sp-example.com` host machine as a root user.**

2 **(Optional) Resolve the following issues, if necessary.**

- The `LD_LIBRARY_PATH` environment variable should *not* be set to the default setting. Change the value to *empty* as in the following example:

  ```
  # setenv LD_LIBRARY_PATH
  ```

- The `JAVA_HOME` environment variable should be set appropriately for your system architecture as in the following example:

  ```
  # setenv JAVA_HOME /usr/jdk/jdk1.5.0_09
  ```

3 **Unzip the Directory Server ZIP file.**

```
# cd /export/DS63
# ls

DSEE.6.3.Solaris10-X86_AMD64-full.tar.gz

# gunzip DSEE.6.3.Solaris10-X86_AMD64-full.tar.gz
```

4 **Untar the resulting `.tar` file.**

```
# tar xvf DSEE.6.1.Solaris10-X86_AMD64-full.tar
```

The `DSEE_ZIP_Distribution` directory is the result of the decompression.

5 **Change into** `DSEE_ZIP_Distribution` **and run** `dsee_deploy install` **to install Directory Server.**

```
# cd DSEE_ZIP_Distribution
# ./dsee_deploy install -i /var/opt/mps/serverroot
```

The Licensing Agreement is displayed. At each `Type return to continue` prompt, press Return to continue.

6 **When** `Do you accept the license terms?` **is displayed, enter** `yes` **to continue.**

Once you accept the license terms, the Directory Server binaries will be installed in the `/var/opt/mps/serverroot/ds6` directory.

▼ To Create a User Data Instance on Directory Server 1

Use this procedure to create a Directory Server instance named `sp-users` for storing user data. The instance uses port 1489 for LDAP and port 1736 for LDAPS.

Before You Begin This procedure assumes you have just completed "To Install Directory Server 1" on page 131 and are still logged into the `ds1.sp-example.com` host machine as a root user.

1 **Change to the Directory Server** `bin` **directory.**

```
# cd /var/opt/mps/serverroot/ds6/bin
```

2 **Run** `dsadm create` **to create a user data instance called** `sp-users`**.**

```
# ./dsadm create -p 1489 -P 1736 /var/opt/mps/sp-users

Choose the Directory Manager password: dsmanager

Confirm the Directory Manager password: dsmanager

use 'dsadm start /var/opt/mps/am-users' to start the instance
```

3 **Run** `dsadm start` **to start the instance.**

```
# ./dsadm start /var/opt/mps/sp-users

Directory Server instance '/var/opt/mps/sp-users' started: pid=11347
```

4 **Run** `netstat` **to verify that the new instance is up and running.**

```
# netstat -an | grep 1489

.1489        *.*        0        0  49152        0 LISTEN
.1489        *.*        0        0  49152        0 LISTEN
```

```
# netstat -an | grep 1736

.1736        *.*        0        0  49152        0 LISTEN
.1736        *.*        0        0  49152        0 LISTEN
```

5 **Run `ldapsearch` to verify that you can read the root Directory Server entry of the new instance.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds1.sp-example.com
-p 1489 -b "" -s base "(objectclass=*)"

version: 1
dn:
objectClass: top
...
supportedLDAPVersion: 3
vendorName: Sun Microsystems, Inc.
vendorVersion: Sun-Java(tm)-System-Directory/6.3
...
```

▼ To Create a Base Suffix for the User Data Instance on Directory Server 1

Use this procedure to create the base suffix in which the user entries will be stored.

Before You Begin This procedure assumes you have just completed "To Create a User Data Instance on Directory Server 1" on page 132 and are still logged into the `ds1.sp-example.com` host machine as a root user.

1 **Run `dsconf create-suffix` to create a base suffix.**

```
# ./dsconf create-suffix -p 1489 -B dbExample
-L /var/opt/mps/sp-users/db/exampleDS o=spusers.com
```

2 **Provide the appropriate information when prompted.**

```
Certificate "CN=ds1, CN=1736, CN=directory Server, O=Sun Microsystems"
presented by the server is not trusted.

Type "Y" to accept, "y" to accept just once, "n" to refuse, "d" for more details: Y

Enter "cn=Directory Manager" password: dsmanager
```

Tip – When you type an uppercase **Y**, you are not asked for the certificate again in the next steps.

3 **Run `dsconf list-suffixes` to verify that the base suffix was successfully created.**

```
# ./dsconf list-suffixes -p 1489

Enter "cn=Directory Manager" password: dsmanager

o=spusers.com
```

If the base suffix was successfully created, `o=spusers.com` is returned. You can also see `sp-users` in a command line list of directory instances.

```
# cd /var/opt/mps
# ls

sp-users serverroot
```

4 **Log out of the `ds1.sp-example.com` host machine.**

▼ To Install Directory Server 2

Before You Begin This procedures assumes "To Download the Directory Server Bits and Required Patches to the Directory Server Host Machines" on page 128 and "To Patch the Directory Server Host Machines" on page 130 have been completed.

1 **Log in to the `ds2.sp-example.com` host machine as a root user.**

2 **(Optional) Resolve the following issues, if necessary.**

- The `LD_LIBRARY_PATH` environment variable should *not* be set to the default setting. Change the value to *empty* as in the following example:

  ```
  # setenv LD_LIBRARY_PATH
  ```

- The `JAVA_HOME` environment variable should be set appropriately for your system architecture as in the following example:

  ```
  # setenv JAVA_HOME /usr/jdk/jdk1.5.0_09
  ```

3 **Unzip the Directory Server ZIP file.**

```
# cd /export/DS63
# ls
```

```
DSEE.6.3.Solaris-Sparc-full.tar.gz

# gunzip DSEE.6.3.Solaris-Sparc-full.tar.gz
```

4 **Untar the resulting .tar file.**

```
# tar xvf DSEE.6.3.Solaris-Sparc-full.tar
```

The DSEE_ZIP_Distribution directory is the result of the decompression.

5 **Change into DSEE_ZIP_Distribution and run dsee_deploy install to install Directory Server.**

```
# cd DSEE_ZIP_Distribution
# ./dsee_deploy install -i /var/opt/mps/serverroot
```

The Licensing Agreement is displayed. At each Type return to continue prompt, press Return to continue.

6 **When Do you accept the license terms? is displayed, enter yes to continue.**

Once you accept the license terms, the Directory Server binaries will be installed in the /var/opt/mps/serverroot/ds6 directory.

▼ To Create a User Data Instance on Directory Server 2

Use this procedure to create a Directory Server instance named am-users for storing user data. The instance uses port 1489 for LDAP and port 1736 for LDAPS.

Before You Begin This procedure assumes you have just completed "To Install Directory Server 2" on page 134 and are still logged into the ds2.sp-example.com host machine as a root user.

1 **Change to the Directory Server bin directory.**

```
# cd /var/opt/mps/serverroot/ds6/bin
```

2 **Run dsadm create to create a user data instance called am-users.**

```
# ./dsadm create -p 1489 -P 1736 /var/opt/mps/sp-users

Choose the Directory Manager password: dsmanager

Confirm the Directory Manager password: dsmanager

use 'dsadm start /var/opt/mps/am-users' to start the instance
```

3 **Run `dsadm start` to start the instance.**

```
# ./dsadm start /var/opt/mps/sp-users

Directory Server instance '/var/opt/mps/sp-users' started: pid=7191
```

4 **Run `netstat` to verify that the new instance is up and running.**

```
# netstat -an | grep 1489

.1489        *.*        0        0  49152        0 LISTEN
.1489        *.*        0        0  49152        0 LISTEN
```

5 **Run `ldapsearch` to verify that you can read the root Directory Server entry of the new instance.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds2.sp-example.com
-p 1489 -b "" -s base "(objectclass=*)"

version: 1
dn:
objectClass: top
...
supportedLDAPVersion: 3
vendorName: Sun Microsystems, Inc.
vendorVersion: Sun-Java(tm)-System-Directory/6.3
...
```

▼ To Create a Base Suffix for the User Data Instance on Directory Server 2

Use this procedure to create the base suffix in which the user entries will be stored.

Before You Begin This procedure assumes you have just completed "To Create a User Data Instance on Directory Server 2" on page 135 and are still logged into the `ds2.sp-example.com` host machine as a root user.

1 **Run `dsconf create-suffix` to create a base suffix.**

```
# ./dsconf create-suffix -p 1489 -B dbExample
-L /var/opt/mps/am-users/db/exampleDS o=spusers.com
```

2 **Provide the appropriate information when prompted.**

```
Certificate "CN=ds2, CN=1736, CN=directory Server, O=Sun Microsystems"
presented by the server is not trusted.
```

```
Type "Y" to accept, "y" to accept just once, "n" to refuse, "d" for more details: Y

Enter "cn=Directory Manager" password: dsmanager
```

Tip – When you type an uppercase **Y**, you are not asked for the certificate again in the next steps.

3 **Run `dsconf list-suffixes` to verify that the base suffix was successfully created.**

```
# ./dsconf list-suffixes -p 1489

Enter "cn=Directory Manager" password: dsmanager

o=siroeusers.com
```

If the base suffix was successfully created, `o=spusers.com` is returned. You can also see `sp-users` in a command line list of directory instances.

```
# cd /var/opt/mps
# ls

sp-users serverroot
```

4 **Log out of the `ds2.sp-example.com` host machine.**

7.2 Enabling Multi-Master Replication of the User Data Instances

This section contains the instructions to enable multi-master replication (MMR) between two Directory Server instances, each configured as a *master*. This includes creating replication agreements between the masters and initializing the second directory master with the data and schema from the first directory master. The previously created `sp-users` user data instances will serve as the two master instances. Use the following list of procedures as a checklist for completing the task.

1. "To Enable Multi-Master Replication for User Data Instance on Directory Server 1" on page 138
2. "To Enable Multi-Master Replication for User Data Instance on Directory Server 2" on page 139
3. "To Change the Default Replication Manager Password for Each User Data Instance" on page 140
4. "To Create Replication Agreements for Each User Data Instance" on page 141
5. "To Initialize the Replication Agreements" on page 142
6. "To Verify Successful User Data Replication" on page 143

▼ To Enable Multi-Master Replication for User Data Instance on Directory Server 1

1 **Log in to the `ds1.sp-example.com` host machine as a root user.**

2 **(Optional) Run `dsconf list-suffixes` to verify that the user data instance is not already enabled for replication.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf list-suffixes -p 1489 -v

Enter "cn=Directory Manager" password: dsmanager
...
o=spusers.com     1        not-replicated        N/A        N/A        29    0

The "list-suffixes" operation succeeded on "ds1.sp-example.com:1489"
```

The base suffix of the user data instance is not replicated.

3 **Run `dsconf enable-repl` to enable replication of the user data instance.**

```
# ./dsconf enable-repl -h ds1.sp-example.com -p 1489
-d 11 master o=spusers.com

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf create-repl-agmt" to create replication agreements on
"o=spusers.com".
```

The `-d` option takes as input a randomly chosen identifier to represent the Directory Server 1 user data instance; in this case, `11 master` indicates that the user data instance is a master and not a replica. The base suffix is specified as `o=spusers.com`.

4 **Run `dsconf list-suffixes` again to verify that the instance is now enabled for replication.**

```
# ./dsconf list-suffixes -p 1489 -v

Enter "cn=Directory Manager" password: dsmanager
...
o=siroeusers.com     1        master(11)        N/A        N/A        29    0

The "list-suffixes" operation succeeded on "ds1.sp-example.com:1489"
```

The base suffix of the instance is `master(11)` indicating that the master was successfully enabled.

5 **Log out of the `ds1.sp-example.com` host machine.**

▼ To Enable Multi-Master Replication for User Data Instance on Directory Server 2

1 **Log in to the** `ds2.sp-example.com` **host machine as a root user.**

2 **(Optional) Run** `dsconf list-suffixes` **to verify that the user data instance is not already enabled for replication.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf list-suffixes -p 1489 -v

Enter "cn=Directory Manager" password: dsmanager
...
o=spusers.com     1        not-replicated        N/A        N/A        29    0

The "list-suffixes" operation succeeded on "ds2.sp-example.com:1489"
```

The base suffix of the user data instance is not replicated.

3 **Run** `dsconf enable-repl` **to enable replication of the user data instance.**

```
# ./dsconf enable-repl -h ds2.sp-example.com -p 1489
-d 22 master o=spusers.com

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf create-repl-agmt" to create replication agreements on
"o=spusers.com".
```

The `-d` option takes as input a randomly chosen identifier to represent the Directory Server 2 user data instance; in this case, `22 master` indicates that the user data instance is a master and not a replica. The base suffix is specified as `o=spusers.com`.

4 **Run** `dsconf list-suffixes` **again to verify that the instance is now enabled for replication.**

```
# ./dsconf list-suffixes -p 1489 -v

Enter "cn=Directory Manager" password: dsmanager
...
o=spusers.com     1        master(22)        N/A        N/A        29        0

The "list-suffixes" operation succeeded on "ds2.sp-example.com:1489"
```

The base suffix of the instance is `master(22)` indicating that the master was successfully enabled.

5 **Log out of the** `ds2.sp-example.com` **host machine.**

▼ To Change the Default Replication Manager Password for Each User Data Instance

The *replication manager* is the user that data suppliers use to bind to the data consumer when sending replication updates. (In MMR the data consumer refers to whichever master happens to be the consumer for a particular operation.) It is recommended to change the default password created during the process of enabling replication.

1 **Log in to the `ds1.sp-example.com` host machine as a root user.**

2 **Create a temporary file that contains the new replication manager password.**

This file will be read once, and the password stored for future use.

```
# cd /var/opt/mps/serverroot/ds6/bin
# echo replmanager > pwd.txt
```

3 **Verify that the file was successfully created.**

```
# cat pwd.txt

replmanager
```

4 **Run `dsconf set-server-prop` to set the replication manager password using `pwd.txt` as input.**

```
# ./dsconf set-server-prop -h ds1.sp-example.com -p 1489
def-repl-manager-pwd-file:pwd.txt

Enter "cn=Directory Manager" password: dsmanager
```

5 **Remove the `pwd.txt` file.**

6 **Log out of the `ds1.sp-example.com` host machine.**

7 **Log in to the `ds2.sp-example.com` host machine as a root user.**

8 **Create a temporary file that contains the new replication manager password.**

This file will be read once, and the password stored for future use.

```
# cd /var/opt/mps/serverroot/ds6/bin
# echo replmanager > pwd.txt
```

9 **Verify that the file was successfully created.**

```
# cat pwd.txt

replmanager
```

10 **Run** `dsconf set-server-prop` **to set the replication manager password using** `pwd.txt` **as input.**

```
# ./dsconf set-server-prop -h ds2.sp-example.com -p 1489
def-repl-manager-pwd-file:pwd.txt

Enter "cn=Directory Manager" password: dsmanager
```

11 **Remove the** `pwd.txt` **file.**

12 **Log out of the** `ds2.sp-example.com` **host machine.**

▼ To Create Replication Agreements for Each User Data Instance

A *replication agreement* is a set of parameters on a supplier that controls how updates are sent to a given consumer. In this deployment, the agreement simply makes the user data instances aware of each other.

1 **Log in to the** `ds1.sp-example.com` **host machine as a root user.**

2 **Run** `dsconf create-repl-agmt` **to create the replication agreement.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf create-repl-agmt -h ds1.sp-example.com
 -p 1489 o=spusers.com ds2.sp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf init-repl-dest o=spusers.com ds1.sp-example.com:1489"
to start replication of "o=spusers.com" data.
```

3 **Run** `dsconf list-repl-agmts` **to verify that the replication agreement was successfully created.**

```
# ./dsconf list-repl-agmts -p 1489

Enter "cn=Directory Manager" password: dsmanager

o=spusers.com ds2.sp-example.com:1489
```

This response indicates that the Directory Server 1 base suffix will be replicated to Directory Server 2.

4 **Log out of the** `ds1.sp-example.com` **host machine.**

5 **Log in to the** `ds2.sp-example.com` **host machine as a root user.**

6 **Run** `dsconf create-repl-agmt` **to create the replication agreement.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf create-repl-agmt -h ds2.sp-example.com -p 1489
o=spusers.com ds1.sp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf init-repl-dest o=spusers.com ds1.sp-example.com:1489"
to start replication of "o=spusers.com" data.
```

7 **Run** `dsconf list-repl-agmts` **to verify that the replication agreement was successfully created.**

```
# ./dsconf list-repl-agmts -p 1489

Enter "cn=Directory Manager" password: dsmanager

o=spusers.com ds1.sp-example.com:1489
```

This response indicates that the Directory Server 2 base suffix will be replicated to Directory Server 1.

8 **Log out of the** `ds2.sp-example.com` **host machine.**

▼ To Initialize the Replication Agreements

Use this procedure to initialize the user data instance on Directory Server 1. The previously created agreements will replicate the data to Directory Server 2.

Note – Initialization is **not** required on both instances when configuring for MMR.

---

1 **Log in to the** `ds1.sp-example.com` **host machine as a root user.**

2 **Run** `dsconf show-repl-agmt-status` **to verify that the replication agreements have not yet been initialized.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsconf show-repl-agmt-status -h ds1.sp-example.com
 -p 1489 o=spusers.com ds2.sp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Configuration Status         : OK
Authentication Status        : OK
Initialization Status        : NOT OK
```

```
Status:                        : Dest. Not Initialized
```

3. **Run `dsconf init-repl-dest` to initialize the replication agreements.**

```
# ./dsconf init-repl-dest -h ds1.sp-example.com
 -p 1489 o=spusers.com ds2.sp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Started initialization of "ds2.sp-example.com:1489"; Sep 13, 2008 9:58:08 AM
Sent 2 entries.
Completed initialization of "ds2.sp-example.com:1489"; Sep 13, 2008 9:58:12 AM
```

4. **Run `dsconf show-repl-agmt-status` again to verify that the replication agreements are now initialized.**

```
# ./dsconf show-repl-agmt-status -h ds1.sp-example.com
 -p 1489 o=spusers.com ds2.sp-example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Configuration Status         : OK
Authentication Status        : OK
Initialization Status        : OK

Status:                        : Enabled
Last Update Date                :  Sep 13, 2008 9:58:17 AM
```

▼ To Verify Successful User Data Replication

Before You Begin This procedure assumes you have just completed "To Initialize the Replication Agreements" on page 142 and are still logged into the `ds2.sp-example.com` host machine as a root user.

1. **Prepare an LDIF file with the following contents and save it in the `/tmp` directory as `people.ldif`.**

```
dn: ou=People,o=spusers.com
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries
```

2. **Run `ldapmodify` on the `ds1.sp-example.com` host machine using `people.ldif` as input.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapmodify -a -h ds1.sp-example.com -p 1489
-f /tmp/people.ldif -D cn=Directory Manager,cn=Administrators,cn=config
```

```
-w dsmanager

adding new entry ou=People,o=spusers.com
```

3 **After the entry is created, log in to the `ds2.sp-example.com` host machine as a root user.**

4 **Run `ldapsearch` on Directory Server 2 to verify that `ou=People` was successfully replicated.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -b "o=spusers.com" -p 1489
-D "cn=Directory Manager" -w dsmanager
"objectclass=organizationalUnit"

version: 1
dn: ou=People,o=spusers.com
objectClass: top
objectClass: organizationalUnit
ou: People
description Container for user entries
```

5 **Now run `ldapdelete` on Directory Server 2 to delete `ou=People`.**

```
# ./ldapdelete -h ds2.sp-example.com -p 1489
 -D "cn=Directory Manager" -w dsmanager
"ou=People,o=spusers.com"
```

6 **Now, as a root user on Directory Server 1, run `ldapsearch` to verify that the deletion was replicated.**

```
# ./ldapsearch -b "o=spusers.com"
-p 1489 -D "cn=Directory Manager" -w dsmanager
"objectclass=organizationalUnit"
```

The search will return no results as the delete was successfully replicated.

7 **Log out of both Directory Server host machines.**

7.3 Modifying the Directory Server Schema

This deployment will be used to test SAML v2 communications. Towards this end, modify the LDAP schema used by the Directory Server user data instances on the service provider side to recognize and store SAML v2 attributes.

▼ To Modify the Directory Server LDAP Schema for SAML v2 User Data

1. **Log in to the `ds2.sp-example.com` host machine as a root user.**

2. **Create an LDIF file with the following information and save it as `/tmp/saml.ldif`.**

   This file includes SAML v2 LDAP attributes.

```
dn: CN=schema
changetype:modify
add:attributeTypes
attributeTypes: ( 1.3.6.1.4.1.42.2.27.9.1.500
NAME ’sun-fm-saml2-nameid-infokey’
DESC ’SAML 2.0 Name Identifier Information Key’
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN
’Sun Java System Access Management’ )

attributeTypes: ( 1.3.6.1.4.1.42.2.27.9.1.501
NAME ’sun-fm-saml2-nameid-info’
DESC ’SAML 2.0 Name Identifier Information’
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN
’Sun Java System Access Management’ )
-
add:objectClasses
objectClasses: ( 1.3.6.1.4.1.42.2.27.9.2.200
NAME ’sunFMSAML2NameIdentifier’
DESC ’SAML 2.0 name identifier objectclass’
SUP top AUXILIARY MAY
( sun-fm-saml2-nameid-infokey $ sun-fm-saml2-nameid-info )
X-ORIGIN ’Sun Java System Access Management’ )
```

3. **Run `ldapmodify` on the `ds1.sp-example.com` host machine using `/tmp/saml.ldif` as input.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ldapmodify -a -h ds2.sp-example.com -p 1489
-D "cn=Directory Manager" -w dsmanager -f /tmp/saml.ldif

modifying entry CN=schema
```

4. **Log out of the `ds1.idp-example.com` host machine.**

7.4 Enabling Secure Communication for the Directory Server User Data Instances

By default, when an instance of Directory Server is created (in this case, `sp-users`), its SSL port is secured with a self-signed certificate named `defaultCert`. A *self-signed certificate* contains a public and private key; the public key is signed by the private key. The `sp-users` instances, though, need to use a server certificate signed by a certificate authority (CA) to allow for secure communication between the instances and the soon-to-be-installed user data load balancer. This entails installing the server certificate signed by the CA and the root certificate confirming the signature of the CA on both Directory Server host machines. Use the following list of procedures as a checklist for completing this task.

1. "To Install a Root Certificate and a Server Certificate on Directory Server 1" on page 146
2. "To Install a Root Certificate and a Server Certificate on Directory Server 2" on page 148

▼ To Install a Root Certificate and a Server Certificate on Directory Server 1

Before You Begin You should already have a root certificate from the CA of your choice. Send server certificate requests to the same CA. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1 **Log in to the `ds1.sp-example.com` host machine as a root user.**

2 **Generate a request for a server certificate signed by a CA.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsadm request-cert -S "CN=ds1.sp-example.com,
OU=OpenSSO Enterprise, O=Sun Microsystems, L=Santa Clara
 ST=California, C=US" -F ascii -o ds-1.csr /var/opt/mps/sp-users
```

`ds-1.csr` is the certificate request.

3 **Send `ds-1.csr` to the CA of your choice.**

The CA issues and returns a certified server certificate named `ds-1.cer`.

4 **Add `ds-1.cer`, the CA-signed server certificate, to the certificate store.**

```
# ./dsadm add-cert /var/opt/mps/sp-users server-cert ds-1.cer
```

5 **Add `ca.cer`, the CA root certificate, to the certificate store.**

```
# ./dsadm add-cert --ca /var/opt/mps/sp-users opensslCA ca.cer
```

6 (Optional) Verify that the CA root certificate was successfully added.

```
# ./dsadm list-certs -C /var/opt/mps/sp-users | grep opensslCA

opensslCA
2008/02/06 00:00  2017/02/06 00:00  n
CN=Certificate Manager,OU=opensso,O=Identity,C=US
Same as issuer
```

7 Configure the Directory Server instance to use the imported certificates.

```
# ./dsconf set-server-prop -h ds1.sp-example.com
-p 1489 ssl-rsa-cert-name:server-cert

Enter "cn=Directory Manager" password: dsmanager

Before setting SSL configuration, export Directory Server data.

Do you want to continue [y/n] ?  y

Directory Server must be restarted for changes to take effect.
```

8 Restart the Directory Server instance.

Directory Server needs to be restarted to use the new certificates.

```
# ./dsadm stop /var/opt/mps/sp-users

Directory Server instance ’/var/opt/mps/sp-users’ stopped

# ./dsadm start /var/opt/mps/sp-users

Directory Server instance ’/var/opt/mps/sp-users’ started: pid=11591
```

9 Run `ldapsearch` on Directory Server 1 to verify that the directory entries can be accessed through the secure port.

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds1.sp-example.com -p 1736
-Z -P /var/opt/mps/sp-users/alias/slapd-cert8.db
-b "" -s base "(objectclass=*)"

version: 1
dn:
objectClass:top
namingContexts: o=spusers.com
supportedExtension: 2.16.840.1.113730.3.5.7
:
supportedSSLCiphers: SSL-CK_RC4_128_EXPORT40_WITH_MD5
supportedSSLCiphers: SSL-CK_RC2_128_CBC_EXPORT40_WITH_MD5
```

This confirms that the Directory Server instance can be accessed through the secure port.

10 **Log out of the `ds1.sp-example.com` host machine.**

▼ To Install a Root Certificate and a Server Certificate on Directory Server 2

Before You Begin You should already have a root certificate from the CA of your choice. Send any server certificate requests to the same CA. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1 **Log in to the `ds2.sp-example.com` host machine as a root user.**

2 **Generate a request for a server certificate signed by a CA.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsadm request-cert -S "CN=ds2.sp-example.com,
OU=OpenSSO Enterprise, O=Sun Microsystems, L=Santa Clara
 ST=California, C=US" -F ascii -o ds-2.csr /var/opt/mps/sp-users
```

`ds-2.csr` is the certificate request.

3 **Send `ds-2.csr` to the CA of your choice.**

The CA issues and returns a certified server certificate named `ds-2.cer`.

4 **Add `ds-2.cer`, the CA-signed server certificate, to the certificate store.**

```
# ./dsadm add-cert /var/opt/mps/sp-users server-cert ds-2.cer
```

5 **Add `ca.cer`, the CA root certificate, to the certificate store.**

```
# ./dsadm add-cert --ca /var/opt/mps/sp-users opensslCA ca.cer
```

6 **(Optional) Verify that the CA root certificate was successfully added.**

```
# ./dsadm list-certs -C /var/opt/mps/sp-users | grep opensslCA

opensslCA
2008/02/06 00:00  2017/02/06 00:00  n
CN=Certificate Manager,OU=opensso,O=Identity,C=us
Same as issuer
```

7 **Configure the Directory Server instance to use the imported certificates.**

```
# ./dsconf set-server-prop -h ds2.sp-example.com
-p 1489 ssl-rsa-cert-name:server-cert
```

```
Enter "cn=Directory Manager" password: dsmanager

Before setting SSL configuration, export Directory Server data.

Do you want to continue [y/n] ?  y

Directory Server must be restarted for changes to take effect.
```

8 **Restart the Directory Server instance.**

Directory Server needs to be restarted to use the new certificates.

```
# ./dsadm stop /var/opt/mps/sp-users

Directory Server instance ’/var/opt/mps/sp-users’ stopped

# ./dsadm start /var/opt/mps/sp-users

Directory Server instance ’/var/opt/mps/sp-users’ started: pid=7311
```

9 **Run `ldapsearch` on Directory Server 2 to verify that the directory entries can be accessed through the secure port.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds2.sp-example.com -p 1736
-Z -P /var/opt/mps/sp-users/alias/slapd-cert8.db
-b "" -s base "(objectclass=*)"

version: 1
dn:
objectClass:top
namingContexts: o=spusers.com
supportedExtension: 2.16.840.1.113730.3.5.7
:
supportedSSLCiphers: SSL-CK_RC4_128_EXPORT40_WITH_MD5
supportedSSLCiphers: SSL-CK_RC2_128_CBC_EXPORT40_WITH_MD5
```

This confirms that the Directory Server instance can be accessed through the secure port.

10 **Log out of the `ds2.sp-example.com` host machine.**

7.5 Configuring the Directory Server Load Balancer

Load Balancer 1 (`lb1.sp-example.com`) is configured in front of the Directory Server user data instances on the service provider side. This section assumes that you have already installed the load balancer. Before beginning, note the following:

- The load balancer hardware and software used in the lab facility for this deployment is BIG-IP® manufactured by F5 Networks. If you are using different load balancer software, see the documentation that comes with that product for detailed settings information.
- Contact your network administrator to obtain an available virtual IP address for the load balancer you want to configure.
- Know the IP address of the load balancer hardware, the URL for the load balancer login page, and a username and password for logging in to the load balancer application.
- Get the IP addresses for Directory Server 1 and Directory Server 2 by running the following command on each host machine:

  ```
  # ifconfig -a
  ```

Use the following list of procedures as a checklist for completing the task.

1. "To Import the Root Certificate to the User Data Load Balancer" on page 150
2. "To Configure Directory Server Load Balancer 1" on page 151

▼ To Import the Root Certificate to the User Data Load Balancer

Install the CA root certificate on the user data load balancer to ensure that a link between the load balancer can be maintained with the CA. Use the same root certificate that you imported in "7.4 Enabling Secure Communication for the Directory Server User Data Instances" on page 146. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1 **Access `https://is-f5.siroe.com`, the BIG-IP load balancer login page, in a web browser.**

2 **Log in to the load balancer as administrator.**

3 **Click Proxies.**

4 **Click the Cert-Admin tab.**

5 **Click Import.**

6 **In the Import Type field, choose Certificate and click Continue.**

7 **Click Browse in the Certificate File field on the Install SSL Certificate page.**

8 **Choose Browser in the Choose File dialog box.**

9 **Navigate to `ca.cer` and click Open.**

10 **Enter `opensslCA` in the Certificate Identifier field.**

11 **Click Install Certificate.**

The Certificate opensslCA page is displayed.

12 **Click Return to Certificate Administration on the Certificate opensslCA page.**

opensslCA, the root certificate, is now included in the Certificate ID list.

▼ To Configure Directory Server Load Balancer 1

Before You Begin This procedure assumes that you have just completed "To Import the Root Certificate to the User Data Load Balancer" on page 150 and are still logged into the load balancer console.

1 **Click** ***Configure your BIG-IP (R) using the Configuration Utility*****.**

2 **Create a Pool.**

A pool contains all the backend server instances.

a. **In the left pane, click Pools.**

b. **On the Pools tab, click Add.**

c. **In the Add Pool dialog, provide the following information:**

Pool Name	**`DirectoryServerSP-UserData-Pool`**
Load Balancing Method	Round Robin
Resources	Add the IP address and port number of both Directory Server host machines.

Note – Use port number 1489.

d. **Click Done.**

3 **Add a Virtual Server.**

The virtual server presents an address to the outside world and, when users attempt to connect, it would forward the connection to the most appropriate real server.

Tip – If you encounter JavaScript™ errors or otherwise cannot proceed to create a virtual server, try using Internet Explorer.

---

a. **In the left frame, click Virtual Servers.**

b. **Click Add on the Virtual Servers tab.**

c. **In the Add a Virtual Server dialog box, provide the following information:**

Address	Enter the IP address for `lb1.sp-example.com`
Service	**489**

d. **Continue to click Next until you reach the Pool Selection dialog box.**

e. **Assign `DirectoryServerSP-UserData-Pool` to the virtual server in the Pool Selection dialog box.**

f. **Click Done.**

4 **Add Monitors**

Monitors are required for the load balancer to detect the backend server failures.

a. **In the left frame, click Monitors.**

b. **Click the Basic Associations tab.**

c. **Add an LDAP monitor for the Directory Server 1 node.**

In the Node column, locate the IP address and port number previously entered for Directory Server 1 and select the Add checkbox.

d. **Add an LDAP monitor for the Directory Server 2 node.**

In the Node column, locate the IP address and port number previously entered for Directory Server 2 and select the Add checkbox.

e. **At the top of the Node column, in the drop-down list, choose `ldap-tcp`.**

f. **Click Apply.**

5 **Configure the load balancer for simple persistence.**

With simple persistence, all requests sent *within a specified interval* are processed by the same Directory Server instance, ensuring complete replication of entries. For example, when a request requires information to be written to Directory Server 1, that information must also be replicated to Directory Server 2. As the replication takes time to complete, if a related request is directed by the load balancer to Directory Server 2 during the replication process itself, the request may fail as the entry might only be partially created. When properly configured, simple persistence ensures that both requests are routed to Directory Server 1 and processed in consecutive order; the first request is finished before the second request begins processing. Simple persistence ensures that within the specified interval, no errors or delays occur due to replication time or redirects when retrieving data. Simple persistence tracks connections based only on the client IP address.

a. **In the left frame, click Pools.**

b. **Click the name of the pool you want to configure.**

   In this example, `DirectoryServerSP-UserData-Pool`.

c. **Click the Persistence tab.**

d. **Under Persistence Type, select Simple.**

e. **Enter 300 seconds for the Timeout interval.**

f. **Click Apply.**

6 **Verify the Directory Server load balancer configuration using the following sub-procedure.**

a. **Log in as a root user on each Directory Server host machine.**

b. **On each host machine, use the `tail` command to monitor the Directory Server access log.**

```
# cd /var/opt/mps/sp-users/logs
# tail -f access
```

   You should see connections to the load balancer IP address opening and closing. For example:

```
[12/July/2008:13:10:20-0700] conn=69755 op=-1 msgId=-1 - closed
[12/July/2008:13:10:25-0700] conn=69756 op=-1 msgId=-1
- fd=27 slot=27 LDAP connection from IP_address to IP_address
[12/July/2008:13:10:25-0700] conn=69756 op=0 msgId=0
- RESULT err=80 tag=120 nentries=0 etime=0
[12/July/2008:13:10:25-0700] conn=69756 op=-1 msgId=-1
- closing from IP_address
```

c. **Execute the following LDAP search against the Directory Server load balancer from Directory Server 1.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h lb1.sp-example.com -p 489 -Z
-P /var/opt/mps/sp-users/alias/slapd-cert8.db
-b "o=spusers.com" -D "cn=directory manager"
-w dsmanager "(objectclass=*)"

version: 1
dn: o=spusers.com
objectClass: top
objectClass: organization
o: spusers.com
```

Make sure the returned entries display in the access log on only one Directory Server host machine.

d. **Run `dsadm stop` to stop Directory Server 1.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsadm stop /var/opt/mps/sp-users
```

e. **Perform the (same) LDAP search against the Directory Server load balancer from Directory Server 2.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h lb1.sp-example.com -p 489 -Z
-P /var/opt/mps/sp-users/alias/slapd-cert8.db
-b "o=spusers.com" -D "cn=directory manager"
-w dsmanager "(objectclass=*)"

version: 1
dn: o=spusers.com
objectClass: top
objectClass: organization
o: spusers.com
```

Make sure the returned entries display in the access log on only Directory Server 2.

Note – You may encounter the following error message:

```
ldap_simple_bind: Cant’ connect to the LDAP
server — Connection refused
```

This means that the load balancer may not fully detect that Directory Server 1 is stopped. In this case, you may have started the search too soon based on the polling interval setting. For example, if the polling interval is set to 10 seconds, you should wait ten seconds to start the search. You can reset the timeout properties to a lower value using the load balancer console.

a. **Click the Monitors tab.**

b. **Click the ldap-tcp monitor name.**

c. **In the Interval field, set the value to 5.**

   This tells the load balancer to poll the server every 5 seconds.

d. **In the Timeout field, set the value to 16.**

e. **Click Apply and repeat the LDAP search.**

See your load balancer documentation for more information on the timeout property.

f. **Start Directory Server 1.**

```
# ./dsadm start /var/opt/mps/sp-users
```

g. **Stop Directory Server 2.**

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsadm stop /var/opt/mps/sp-users
```

h. **Perform the (same) LDAP search against the Directory Server load balancer from Directory Server 1 to confirm that the request is forwarded to the running Directory Server 1.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
 ./ldapsearch -h lb1.sp-example.com -p 489 -Z
-P /var/opt/mps/am-users/alias/slapd-cert8.db
-b "o=spusers.com" -D "cn=directory manager"
-w dsmanager "(objectclass=*)"

version: 1
dn: o=spusers.com
objectClass: top
objectClass: organization
o: spusers.com
```

Make sure the returned entries display in the access log on only Directory Server 1.

i. **Start Directory Server 2.**

```
# ./dsadm start /var/opt/mps/sp-users
```

j. **Log out of both Directory Server host machines and the load balancer console.**

7.6 Creating a Test User

Create a user entry in the replicated Directory Server user data instances for `spuser`.

Note – If you are using an existing user data store, create the appropriate users in it and move on to Chapter 9, "Configuring OpenSSO Enterprise Realms for User Authentication."

▼ To Import Test User Data into the Replicated Directory Server Instances

Create an LDIF file for the test user and import the file into `ds1.sp-example.com`. The test user data will then be replicated to `ds2.sp-example.com`.

1 **Log in to the `ds1.sp-example.com` host machine as a root user.**

2 **Create an LDIF file with the following entries.**

```
dn: ou=users,o=spusers.com
objectclass: top
objectclass: organizationalUnit
ou: users
description: Container for user entries

dn: ou=Groups,o=spusers.com
objectClass: top
objectClass: organizationalUnit
ou: Groups
description: Container for group entries

dn: uid=spuser,ou=users,o=spusers.com
uid: spuser
givenName: sp
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetadmin
objectClass: inetorgperson
```

```
objectClass: inetUser
sn: user
cn: sp user
userPassword: spuser
inetUserStatus: Active
```

3 **Save the file as `sp-users.ldif` in the `/tmp` directory.**

4 **Import the LDIF file into Directory Server 1 using `ldapmodify`.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapmodify -h ds1.sp-example.com -p 1489
 -D "cn=Directory Manager" -w dsmanager
 -a -f /tmp/sp-users.ldif

adding new entry ou=users,o=spusers.com

adding new entry ou=Groups,o=spusers.com

adding new entry uid=spuser,ou=users,o=spusers.com
```

5 **Verify that the new users were imported using `ldapsearch`.**

```
# ./ldapsearch -h ds1.sp-example.com
 -b "o=spusers.com" -p 1489 -D "cn=Directory Manager"
 -w dsmanager "uid=spuser"

version: 1
dn: uid=spuser,ou=users,o=spusers.com
uid: spuser
givenName: sp
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetadmin
objectClass: inetorgperson
objectClass: inetUser
sn: user
cn: sp user
userPassword:
 {SSHA}H5LpB+QLZMoL9SiXzY/DokHKXRclELVy7w25AA==
inetUserStatus: Active
```

6 **Log out of the `ds1.sp-example.com` host machine.**

7 **(Optional) Verify that the entries were replicated to Directory Server 2 by logging in as a root user to the `ds2.idp-example.com` host machine and using `ldapsearch`.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds2.sp-example.com
 -b "o=spusers.com" -p 1489 -D "cn=Directory Manager"
 -w dsmanager ""

version: 1
dn: o=spusers.com
objectClass: top
objectClass: domain
dc: company

dn: ou=users,o=spusers.com
objectClass: top
objectClass: organizationalUnit
ou: users
description: Container for user entries

dn: ou=Groups,o=spusers.com
objectClass: top
objectClass: organizationalUnit
ou: Groups
description: Container for group entries

dn: uid=spuser,ou=users,o=spusers.com
uid: spuser
givenName: sp
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetadmin
objectClass: inetorgperson
objectClass: inetUser
sn: user
cn: sp user
userPassword:
 {SSHA}H5LpB+QLZMoL9SiXzY/DokHKXRclELVy7w25AA==
inetUserStatus: Active
```

8 **Log out of the `ds2.sp-example.com` host machine.**

CHAPTER 8

Deploying and Configuring OpenSSO Enterprise

This chapter includes instructions on how to deploy and configure two instances of Sun OpenSSO Enterprise 8.0 on the service provider side. It begins with the installation of Sun Java^{TM} System Application Server onto each host machine, followed by the deployment and configuration of the OpenSSO Enterprise WAR. This chapter contains the following sections:

8.1 Installing the Application Server Web Containers

In this section, we create a non-root user with the `roleadd` command in the Solaris Operating Environment on each OpenSSO Enterprise host machine and install Sun Java System Application Server 9.1 Update 1 using the non-root user. Use the following list of procedures as a checklist for completing the task.

1. "To Patch the OpenSSO Enterprise Host Machines" on page 160
2. "To Create a Non-Root User on the OpenSSO Enterprise 1 Host Machine" on page 160
3. "To Install Application Server on the OpenSSO Enterprise 1 Host Machine" on page 161
4. "To Create a Non-Root User on the OpenSSO Enterprise 2 Host Machine" on page 172
5. "To Install Application Server on the OpenSSO Enterprise 2 Host Machine" on page 173

Note – We use `roleadd` rather than `useradd` for security reasons; `roleadd` disables the ability of the user to log in.

▼ To Patch the OpenSSO Enterprise Host Machines

On our lab machines, the required Application Server patch is 117461–08. Results for your machine might be different. Read the latest documentation for your web container to determine if you need to install patches and, if so, what they might be. You can search for patches directly at http://sunsolve.sun.com. Navigate to the PatchFinder page, enter the patch number, click Find Patch, and download the appropriate patch for the OpenSSO Enterprise 1 host machine (osso1.sp-example.com) and the OpenSSO Enterprise 2 host machine (osso2.sp-example.com).

1 **Log in to the** `osso1.sp-example.com` **host machine as a root user.**

2 **Run** `patchadd` **to see if the patch is already installed.**

```
# patchadd -p | grep 117461-08
```

A series of patch numbers are displayed, and patch 117461–08 is present so there is no need to install any patches at this time.

3 **Log out of the** `osso1.sp-example.com` **host machine.**

4 **Log in to the** `osso2.sp-example.com` **host machine as a root user.**

5 **Run** `patchadd` **to see if the patch is already installed.**

```
# patchadd -p | grep 117461-08
```

A series of patch numbers are displayed, and patch 117461–08 is present so there is no need to install any patches at this time.

6 **Log out of the** `osso1.sp-example.com` **host machine.**

▼ To Create a Non-Root User on the OpenSSO Enterprise 1 Host Machine

1 **Log in to the** `osso1.sp-example.com` **host machine as a root user.**

2 **Create a new user with** `roleadd`**.**

```
# roleadd -s /sbin/sh -m -g staff -d /export/osso80adm osso80adm
```

3 **(Optional) Verify that the user was created.**

```
# cat /etc/passwd

root:x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1::/:
```

```
...
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
osso80adm:x:223830:10::/export/osso80adm:/sbin/sh
```

4. **(Optional) Verify that the user's directory was created.**

```
# cd /export/osso80adm
# ls

local.cshrc    local.profile    local.login
```

5. **Create a password for the non-root user.**

```
# passwd osso80adm
New Password: nonroot1pwd
Re-ener new Pasword: nonroot1pwd

passwd: password successfully changed for osso80adm
```

Caution – If you do not perform this step, you will not be able to *switch user* (`su`) when logged in as the non-root user.

▼ To Install Application Server on the OpenSSO Enterprise 1 Host Machine

Install Application Server and the appropriate CA root and CA-signed server certificates.

Before You Begin This procedure assumes you have just completed "To Create a Non-Root User on the OpenSSO Enterprise 1 Host Machine" on page 160 and are still logged into the `osso1.sp-example.com` host machine as a root user.

1. **Create a directory into which the Application Server bits can be downloaded and change into it.**

```
# mkdir /export/AS91
# cd /export/AS91
```

2. **Download the Sun Java System Application Server 9.1 Update 2 binary from the Sun Microsystems Product Download page to the `/export/AS91` directory.**

3. **Grant the downloaded binary execute permission using the `chmod` command.**

```
# chmod +x sjsas-9_1_02-solaris-sparc-ml.bin
```

4. **Install the software.**

```
# ./sjsas-9_1_02-solaris-sparc-ml.bin -console
```

5 When prompted, provide the following information.

`You are running the installation program for the Sun Java System Application Server. This program asks you to supply configuration preference settings that it uses to install the server.`
`This installation program consists of one or more selections that provide you with information and let you enter preferences that determine how Sun Java System Application Server is installed and configured.`
`When you are presented with the following question, the installation process pauses to allow you to read the information that has been presented When you are ready to continue, press Enter.`
	Press Enter to continue.

`Some questions require more detailed information that you are required to type. The question may have a default value that is displayed inside of brackets []. For example, the following question has a default answer of yes:`
`Are you sure? [yes]`
`If you want to accept the default answer, press only the Enter key (which on some keyboards is labeled Return).`
`If you want to provide a different answer, type it at the command prompt and then press Enter.`
	Press Enter to continue.

`Welcome to the Sun Java System Application Server Installation program.`	Press Enter to continue.
`Before you install this product, you must read and accept the entire Software License Agreement under which this product is licensed for your use.`	Press Enter to display the Software License Agreement.

This Agreement, including any terms contained in your Entitlement, is the entire agreement between you and Sun relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party. Please contact Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95054 if you have questions. If you have read and accept all the terms of the entire Software License Agreement, answer ’yes’, and the installation will continue. If you do not accept all the terms of the Software License Agreement, answer ’no’, and the installation program will end without installing the product. Have you read, and do you accept, all of the terms of the preceding Software License Agreement [no] {"<" goes back, "!" exits}?	Type **yes** and press Enter.
The Sun Java System Application Server components will be installed in the following directory, which is referred to as the "Installation Directory".To use this directory, press only the Enter key. To use a different directory, type in the full path of the directory to use followed by pressing the Enter key. Installation Directory [/opt/SUNWappserver] {"<" goes back, "!" exits}	Enter **/opt/SUNWappserver91**

`The directory "/opt/SUNWappserver91"`
`does not exist. Do you want to create it now or`
`choose another directory?`
 	
`1. Create Directory`
`2. Choose New.`
 	
`Enter the number corresponding to your choice [1]`
`{"<" goes back, "!" exits}`
	Press Enter to accept the default value.

`The Sun Java System Application Server`
`requires a Java 2 SDK. Please provide the path to`
`a Java 2 SDK 5.0 or greater. [/usr/jdk/instances/jdk1.5.0]`
`{"<" goes back, "!" exits}`
	Press Enter to accept the default value.

`Supply the admin user’s password and override`
`any of the other initial configuration settings as`
`necessary.`
 	
`Admin User [admin] {"<" goes back, "!" exits}`
	Press Enter to accept the default value.

`Admin User’s Password (8 chars minimum):`
`Re-enter Password:`
	Enter **`domain1pwd`** and then re-enter **`domain1pwd`**.

`Do you want to store admin user name and`
`password in .asadminpass file in user’s home`
`directory [yes] {"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`Admin Port [4848] {"<" goes back, "!" exits}`
`HTTP Port [8080] {"<" goes back, "!" exits}`
`HTTPS Port [8181] {"<" goes back, "!" exits}`
	Press Enter to accept the three default values.

`Do you want to enable Updatecenter client`
`[yes] {"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`Do you want to upgrade from previous`
`Applicatin Server version [no]`
`{"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

The following items for the product Sun Java System Application Server will be installed: Product: Sun Java System Application Server Location: /opt/SUNWappserver91 Space Required: 185.42 MB --- Sun Java System Message Queue 4.1 Application Server Startup Ready To Install 1. Install Now 2. Start Over 3. Exit Installation What would you like to do [1] {"<" goes back, "!" exits}?	Press Enter to accept the default value and begin the installation process.
- Installing Sun Java System Application Server |-1%-----25%-----50%-----75%-----100%| - Installation Successful.	When installation is complete, an `Installation Successful` message is displayed:
Next Steps: 1. Access the About Application Server 9.1 welcome page at: file:///opt/SUNWappserver91/docs/about.html 2. Start the Application Server by executing: /opt/SUNWappserver91/bin/asadmin start-domain domain1 3. Start the Admin Console: http://osso1.sp-example.com:4848 Please press Enter/Return key to exit the installation program. {"!" exits}	Press Enter to exit the installation program.

6 Create a second Application Server domain for the non-root user.

The default domain created during the installation process is owned by root. We create a new domain for `osso80adm`, the non-root user, into which we will deploy OpenSSO Enterprise.

```
# cd /opt/SUNWappserver91/bin
# su osso80adm
# ./asadmin create-domain
```

```
--domaindir /export/osso80adm/domains
--adminport 8989 --user domain2adm --instanceport 1080
--domainproperties http.ssl.port=1081 ossodomain

 Please enter the admin password>

domain2pwd

Please enter the admin password again>

domain2pwd

Please enter the master password

  [Enter to accept the default]:>

domain2master

Please enter the master password again

  [Enter to accept the default]:>

domain2master

Using port 8989 for Admin.
Using port 1080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using port 1081 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Domain being created with profile:developer, as specified
  by variable AS_ADMIN_PROFILE in configuration file.
Security Store uses: JKS
2008-09-14 18:21:15.907 GMT Thread[main,5,main]
java.io.FileNotFoundException:
derby.log (Permission denied)
-------------------------------------------------
2008-09-14 18:21:16.216 GMT:
Booting Derby version The Apache Software Foundation
- Apache Derby - 10.2.2.1 -
(538595): instance c013800d-0118-e205-d50b-00000c0c0770
on database directory
/export/osso80adm/domains/ossodomain/lib/databases/ejbtimer

  Database Class Loader started - derby.database.classpath=’’
  Domain ossodomain created.
```

Note – Creating a non-root domain displays a `FileNotFoundException`. Please see Appendix G, “Known Issues and Limitations.”

7 **Verify that the non-root user domain was created with the correct permissions using the following sub-procedure.**

a. **Change to the** `ossodomain` **directory.**

```
# cd /export/osso80adm/domains/ossodomain
```

b. **List the contents of the directory.**

```
# ls -la

total 30
drwxr-xr-x  15 osso80adm staff   512 Sep 14 16:43 .
drwxr-xr-x   3 osso80adm staff   512 Sep 14 16:43 ..
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 addons
drwxr-xr-x   6 osso80adm staff   512 Sep 14 16:43 applications
drwxr-xr-x   3 osso80adm staff   512 Sep 14 16:43 autodeploy
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 bin
drwx------   3 osso80adm staff  1024 Sep 14 16:43 config
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 docroot
drwxr-xr-x   6 osso80adm staff   512 Sep 14 16:43 generated
drwxr-xr-x   3 osso80adm staff   512 Sep 14 16:43 imq
drwxr-xr-x   5 osso80adm staff   512 Sep 14 16:43 java-web-start
drwxr-xr-x   8 osso80adm staff   512 Sep 14 16:43 jbi
drwxr-xr-x   6 osso80adm staff   512 Sep 14 16:43 lib
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 logs
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 session-store
```

The files and directories are owned by `osso80adm`.

8 **Start** `ossodomain`**, the non-root user domain, using the following sub-procedure.**

a. **Change to the non-root user domain** `bin` **directory.**

```
# cd /export/osso80adm/domains/ossodomain/bin
```

b. **Start** `ossodomain`**.**

```
# ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master
```

```
Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

9 **Verify that `ossodomain` has started with the following sub-procedure.**

a. **Access `http://osso1.sp-example.com:8989/login.jsf` from a web browser.**

b. **Log in to the Application Server console as the `ossodomain` administrator.**

Username	**domain2adm**
Password	**domain2pwd**

When the Application Server administration console is displayed, it is verification that the non-root user was able to start the domain server.

c. **Exit the console and close the browser.**

10 **Create a request for a CA-signed server certificate to secure communications between the soon-to-be-configured OpenSSO Enterprise load balancer and `ossodomain` using the following sub-procedure.**

a. **Generate a private/public key pair and reference it with the alias, `opensso-sp-1`.**

`opensso-sp-1` will be used in a later step to retrieve the public key which is contained in a self-signed certificate.

```
# cd /export/osso80adm/domains/ossodomain/config
# keytool -genkey -noprompt -keyalg rsa -keypass domain2master
-alias opensso-sp-1 -keystore keystore.jks -dname "CN=osso1.sp-example.com,
OU=OpenSSO, O=Sun Microsystems, L=Santa Clara, ST=California, C=US"
-storepass domain2master
```

b. **Verify that the key pair was successfully created and stored in the certificate store.**

```
# keytool -list -v -keystore keystore.jks -storepass domain2master

 Keystore type: jks
 Keystore provider: SUN

 Your keystore contains two entries.
...
 Alias name: opensso-sp-1
 Creation date: Sep 14, 2008
 Entry type: keyEntry
 Certificate chain length: 1
 Certificate[1]:
 Owner: CN=osso1.sp-example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
 Issuer: CN=osso-osso1.sp-example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
 Serial number: 48cdb299
 Valid from: Sun Sep 14 15:02:47 PDT 2008 until: Sat Dec 13 15:02:47 PDT 2008
 Certificate fingerprints:
  MD5:  14:0F:88:BC:C8:6F:2C:8B:F0:A2:C2:F1:AF:FC:93:F1:
  SHA1: 9D:22:05:14:51:21:33:CB:06:36:25:FE:0A:B6:DF:45:EE:B1:19:86:
```

Note – The output of this command may list more than one certificate based on the entries in the keystore.

c. **Generate a CA-signed server certificate request.**

```
# keytool -certreq -alias opensso-sp-1 -keypass domain2master
-keystore keystore.jks -storepass domain2master file opensso-sp-1.csr
```

opensso-sp-1.csr is the server certificate request.

d. **(Optional) Verify that opensso-sp-1.csr was created.**

```
# ls -la opensso-sp-1.csr

 -rw-r--r--   1 osso80adm staff       715 Sep 14 15:04 opensso-sp-1.csr
```

e. **Send osso-sp-1.csr to the CA of your choice.**

The CA issues and returns a certified certificate named opensso-sp-1.cer.

f. **Import ca.cer, the CA root certificate.**

The root certificate must be imported into two keystores (keystore.jks and cacerts.jks) with Application Server. Use the same root certificate that you imported in “7.4 Enabling Secure Communication for the Directory Server User Data Instances” on page 146. For more information, see “3.3 Obtaining Secure Socket Layer Certificates” on page 36.

```
# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore keystore.jks -storepass domain2master

Owner: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes
```

```
Certificate was added to keystore

# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore cacerts.jks -storepass ossomaster

Owner: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore
```

g. **Replace the self-signed public key certificate (associated with the `s1as` alias) with the CA-signed server certificate.**

```
# keytool -import -file opensso-sp-1.cer -alias opensso-sp-1
-keystore keystore.jks -storepass domain2master

Certificate reply was installed in keystore
```

h. **(Optional) Verify that the self-signed public key certificate has been overwritten by the server certificate received from the CA.**

```
# keytool -list -v -keystore keystore.jks
-storepass domain2master

The certificate indicated by the alias "osso-sp-1" is signed by CA.
```

i. **Change the certificate alias from the default `s1as` to the new `opensso-sp-1` in the `domain.xml` file for the `ossodomain` domain.**

The Application Server configuration file is `domain.xml`.

```
<http-listener acceptor-threads="1" address="0.0.0.0"
blocking-enabled="false" default-virtual-server="server" enabled="true"
family="inet" id="http-listener-2" port="1081" security-enabled="true"
server-name="" xpowered-by="true">
<ssl cert-nickname="opensso-sp-1" client-auth-enabled="false" ssl2-enabled="false"
ssl3-enabled="true" tls-enabled="true" tls-rollback-enabled="true"/>
```

Tip – Backup `domain.xml` before modifying it.

---

11 **Modify the JVM options in your web container's configuration file using the following sub-procedure.**

OpenSSO Enterprise is deployed with an embedded configuration data store (if desired). In order for the configuration data store to be created successfully, the following JVM options should be modified in the web container's configuration file. We will be modifying `domain.xml` again for this example.

Tip – Backup `domain.xml` before modifying it.

---

a. **Change to the `config` directory.**

```
# cd /export/osso80adm/domains/ossodomain/config
```

b. **Open `domain.xml` in a text editor and make the following changes:**

- Replace **`<jvm-options>-client</jvm-options>`** with **`<jvm-options>-server</jvm-options>`**.
- Replace **`<jvm-options>-Xmx512m</jvm-options>`** with **`<jvm-options>-Xmx1024m</jvm-options>`**.

c. **Save the file and close it.**

12 **Restart the `ossodomain` domain.**

```
# cd /export/osso80adm/domains/ossodomain/bin
# ./stopserv

Server was successfully stopped.

./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

13 **Verify that the certificate used for SSL communication is the root CA certificate.**

a. **Access `https://osso1.sp-example.com:1081/index.html` from a web browser.**

b. **View the details of the certificate in the security warning to ensure that it is Issued by “OpenSSLTestCA”.**

After inspecting and accepting the certificate, you should see the default `index.html` page.

c. **Close the browser.**

14 **Log out of the `osso1.sp-example.com` host machine.**

To Create a Non-Root User on the OpenSSO Enterprise 2 Host Machine

1 **Log in to the `osso2.sp-example.com` host machine as a root user.**

2 **Create a new user with `roleadd`.**

```
# roleadd -s /sbin/sh -m -g staff -d /export/osso80adm osso80adm
```

3 **(Optional) Verify that the user was created.**

```
# cat /etc/passwd

root:x:0:0:Super-User:/:/sbin/sh
daemon:x:1:1::/:
...
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
osso80adm:x:223830:10::/export/osso80adm:/sbin/sh
```

4 **(Optional) Verify that the user's directory was created.**

```
# cd /export/osso80adm
# ls

local.cshrc    local.profile    local.login
```

5 **Create a password for the non-root user.**

```
# passwd osso80adm
New Password: nonroot2pwd
Re-ener new Pasword: nonroot2pwd

passwd: password successfully changed for osso80adm
```

Caution – If you do not perform this step, you will not be able to *switch user* (`su`) when logged in as the non-root user.

▼ To Install Application Server on the OpenSSO Enterprise 2 Host Machine

Install Application Server and the appropriate CA root and CA-signed server certificates.

Before You Begin This procedure assumes you have just completed "To Create a Non-Root User on the OpenSSO Enterprise 2 Host Machine" on page 172 and are still logged into the `osso2.sp-example.com` host machine as a root user.

1 **Create a directory into which the Application Server bits can be downloaded and change into it.**

```
# mkdir /export/AS91
# cd /export/AS91
```

2 **Download the Sun Java System Application Server 9.1 Update 2 binary from the Sun Microsystems Product Download page to the `AS91` directory of the `osso2.sp-example.com` host machine.**

3 **Grant the downloaded binary execute permission using the `chmod` command.**

```
# chmod +x sjsas-9_1_02-solaris-sparc-ml.bin
```

4 **Install the software.**

```
# ./sjsas-9_1_02-solaris-sparc-ml.bin -console
```

5 **When prompted, provide the following information.**

`You are running the installation program for the Sun Java System Application Server. This program asks you to supply configuration preference settings that it uses to install the server.`
`This installation program consists of one or more selections that provide you with information and let you enter preferences that determine how Sun Java System Application Server is installed and configured.`
`When you are presented with the following question, the installation process pauses to allow you to read the information that has been presented When you are ready to continue, press Enter.`
	Press Enter to continue.

Some questions require more detailed information that you are required to type. The question may have a default value that is displayed inside of brackets []. For example, the following question has a default answer of yes: Are you sure? [yes] If you want to accept the default answer, press only the Enter key (which on some keyboards is labeled Return). If you want to provide a different answer, type it at the command prompt and then press Enter.	Press Enter to continue.
Welcome to the Sun Java System Application Server Installation program.	Press Enter to continue.
Before you install this product, you must read and accept the entire Software License Agreement under which this product is licensed for your use.	Press Enter to display the Software License Agreement.

This Agreement, including any terms contained in your Entitlement, is the entire agreement between you and Sun relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party. Please contact Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95054 if you have questions. If you have read and accept all the terms of the entire Software License Agreement, answer ’yes’, and the installation will continue. If you do not accept all the terms of the Software License Agreement, answer ’no’, and the installation program will end without installing the product. Have you read, and do you accept, all of the terms of the preceding Software License Agreement [no] {"<" goes back, "!" exits}?	Type **yes** and press Enter.
The Sun Java System Application Server components will be installed in the following directory, which is referred to as the "Installation Directory".To use this directory, press only the Enter key. To use a different directory, type in the full path of the directory to use followed by pressing the Enter key. Installation Directory [/opt/SUNWappserver] {"<" goes back, "!" exits}	Enter **/opt/SUNWappserver91**

`The directory "/opt/SUNWappserver91"`
`does not exist. Do you want to create it now or`
`choose another directory?`
 	
`1. Create Directory`
`2. Choose New.`
 	
`Enter the number corresponding to your choice [1]`
`{"<" goes back, "!" exits}`
	Press Enter to accept the default value.

`The Sun Java System Application Server`
`requires a Java 2 SDK. Please provide the path to`
`a Java 2 SDK 5.0 or greater. [/usr/jdk/instances/jdk1.5.0]`
`{"<" goes back, "!" exits}`
	Press Enter to accept the default value.

`Supply the admin user's password and override`
`any of the other initial configuration settings as`
`necessary.`
 	
`Admin User [admin] {"<" goes back, "!" exits}`
	Press Enter to accept the default value.

`Admin User's Password (8 chars minimum):`
`Re-enter Password:`
	Enter **`domain1pwd`** and then re-enter **`domain1pwd`**.

`Do you want to store admin user name and`
`password in .asadminpass file in user's home`
`directory [yes] {"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`Admin Port [4848] {"<" goes back, "!" exits}`
`HTTP Port [8080] {"<" goes back, "!" exits}`
`HTTPS Port [8181] {"<" goes back, "!" exits}`
	Press Enter to accept the three default values.

`Do you want to enable Updatecenter client`
`[yes] {"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

`Do you want to upgrade from previous`
`Applicatin Server version [no]`
`{"<" goes back, "!" exits}?`
	Press Enter to accept the default value.

```
The following items for the product Sun Java
System Application Server will be installed:

Product: Sun Java System Application Server
Location: /opt/SUNWappserver91
Space Required: 185.42 MB
-------------------------------------------
Sun Java System Message Queue 4.1
Application Server
Startup

Ready To Install

1. Install Now
2. Start Over
3. Exit Installation

What would you like to do [1]
{"<" goes back, "!" exits}?
```

Press Enter to accept the default value and begin the installation process.

```
- Installing Sun Java System Application
Server

|-1%-----25%-----50%-----75%-----100%|

 - Installation Successful.
```

When installation is complete, an `Installation Successful` message is displayed:

```
Next Steps:

1. Access the About Application Server 9.1 welcome
page at:
 file:///opt/SUNWappserver91/docs/about.html

2. Start the Application Server by executing:
  /opt/SUNWappserver91/bin/asadmin
  start-domain domain1

3. Start the Admin Console:
  http://osso2.sp-example.com:4848

Please press Enter/Return key to exit the
installation program. {"!" exits}
```

Press Enter to exit the installation program.

6 Create a second Application Server domain for the non-root user.

The default domain created during the installation process is owned by root. We create a new domain for `osso80adm`, the non-root user, into which we will deploy OpenSSO Enterprise.

```
# cd /opt/SUNWappserver91/bin
# su osso80adm
# ./asadmin create-domain
```

```
--domaindir /export/osso80adm/domains
--adminport 8989 --user domain2adm --instanceport 1080
--domainproperties http.ssl.port=1081 ossodomain

 Please enter the admin password>

domain2pwd

Please enter the admin password again>

domain2pwd

Please enter the master password

  [Enter to accept the default]:>

domain2master

Please enter the master password again

  [Enter to accept the default]:>

domain2master

Using port 8989 for Admin.
Using port 1080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using port 1081 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Domain being created with profile:developer, as specified
  by variable AS_ADMIN_PROFILE in configuration file.
Security Store uses: JKS
2008-09-14 18:21:15.907 GMT Thread[main,5,main]
java.io.FileNotFoundException:
derby.log (Permission denied)
-------------------------------------------------
2008-09-14 18:21:16.216 GMT:
Booting Derby version The Apache Software Foundation
- Apache Derby - 10.2.2.1 -
(538595): instance c013800d-0118-e205-d50b-00000c0c0770
on database directory
/export/osso80adm/domains/ossodomain/lib/databases/ejbtimer

  Database Class Loader started - derby.database.classpath=’’
  Domain ossodomain created.
```

Note – Creating a non-root domain displays a `FileNotFoundException`. Please see Appendix G, "Known Issues and Limitations."

7 **Verify that the non-root user domain was created with the correct permissions using the following sub-procedure.**

a. **Change to the `ossodomain` directory.**

```
# cd /export/osso80adm/domains/ossodomain
```

b. **List the contents of the directory.**

```
# ls -la

total 30
drwxr-xr-x  15 osso80adm staff   512 Sep 14 16:43 .
drwxr-xr-x   3 osso80adm staff   512 Sep 14 16:43 ..
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 addons
drwxr-xr-x   6 osso80adm staff   512 Sep 14 16:43 applications
drwxr-xr-x   3 osso80adm staff   512 Sep 14 16:43 autodeploy
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 bin
drwx------   3 osso80adm staff  1024 Sep 14 16:43 config
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 docroot
drwxr-xr-x   6 osso80adm staff   512 Sep 14 16:43 generated
drwxr-xr-x   3 osso80adm staff   512 Sep 14 16:43 imq
drwxr-xr-x   5 osso80adm staff   512 Sep 14 16:43 java-web-start
drwxr-xr-x   8 osso80adm staff   512 Sep 14 16:43 jbi
drwxr-xr-x   6 osso80adm staff   512 Sep 14 16:43 lib
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 logs
drwxr-xr-x   2 osso80adm staff   512 Sep 14 16:43 session-store
```

The files and directories are owned by `osso80adm`.

8 **Start `ossodomain`, the non-root user domain, using the following sub-procedure.**

a. **Change to the non-root user domain `bin` directory.**

```
# cd /export/osso80adm/domains/ossodomain/bin
```

b. **Start `ossodomain`.**

```
# ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master
```

```
Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

9 Verify that `ossodomain` has started with the following sub-procedure.

a. Access `http://osso2.sp-example.com:8989/login.jsf` from a web browser.

b. Log in to the Application Server console as the `ossodomain` administrator.

Username	**domain2adm**
Password	**domain2pwd**

When the Application Server administration console is displayed, it is verification that the non-root user was able to start the domain server.

c. Exit the console and close the browser.

10 Create a request for a CA-signed server certificate to secure communications between the soon-to-be-configured OpenSSO Enterprise load balancer and `ossodomain` using the following sub-procedure.

a. Generate a private/public key pair and reference it with the alias, `opensso-sp-2`.

`opensso-sp-2` will be used in a later step to retrieve the public key which is contained in a self-signed certificate.

```
# cd /export/osso80adm/domains/ossodomain/config
# keytool -genkey -noprompt -keyalg rsa -keypass domain2master
-alias opensso-sp-2 -keystore keystore.jks -dname "CN=osso2.sp-example.com,
OU=OpenSSO, O=Sun Microsystems, L=Santa Clara, ST=California, C=US"
-storepass domain2master
```

b. Verify that the key pair was successfully created and stored in the certificate store.

```
# keytool -list -v -keystore keystore.jks -storepass domain2master

 Keystore type: jks
 Keystore provider: SUN

 Your keystore contains two entries.
...
...
 Alias name: opensso-sp-2
 Creation date: Sep 14, 2008
 Entry type: keyEntry
 Certificate chain length: 1
 Certificate[1]:
 Owner: CN=osso2.sp-example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
```

```
 Issuer: CN=osso2.sp-example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
 Serial number: 48cdb299
 Valid from: Sun Sep 14 15:02:47 PDT 2008 until: Sat Dec 13 15:02:47 PDT 2008
 Certificate fingerprints:
  MD5:  14:0F:88:BC:C8:6F:2C:8B:F0:A2:C2:F1:AF:FC:93:F1:
  SHA1: 9D:22:05:14:51:21:33:CB:06:36:25:FE:0A:B6:DF:45:EE:B1:19:86:
```

Note – The output of this command may list more than one certificate based on the entries in the keystore.

c. Generate a CA-signed server certificate request.

```
# keytool -certreq -alias opensso-sp-2 -keypass domain2master
-keystore keystore.jks -storepass domain2master file opensso-sp-2.csr
```

opensso-sp-2.csr is the server certificate request.

d. (Optional) Verify that opensso-sp-2.csr was created.

```
# ls -la opensso-sp-2.csr

 -rw-r--r--   1 osso80adm staff        715 Sep 14 15:04 opensso-sp-2.csr
```

e. Send opensso-sp-2.csr to the CA of your choice.

The CA issues and returns a certified certificate named opensso-sp-2.cer.

f. Import ca.cer, the CA root certificate.

The root certificate must be imported into two keystores (keystore.jks and cacerts.jks) with Application Server. Use the same root certificate that you imported in "7.4 Enabling Secure Communication for the Directory Server User Data Instances" on page 146. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

```
# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore keystore.jks -storepass domain2master

Owner: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes
```

```
Certificate was added to keystore

# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore cacerts.jks -storepass domain2master

Owner: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore
```

g. **Replace the self-signed public key certificate (associated with the `s1as` alias) with the CA-signed server certificate.**

```
# keytool -import -file opensso-sp-2.cer -alias opensso-sp-2
-keystore keystore.jks -storepass domain2master

Certificate reply was installed in keystore
```

h. **(Optional) Verify that the self-signed public key certificate has been overwritten by the CA-signed server certificate.**

```
# keytool -list -v -keystore keystore.jks
-storepass domain2master

The certificate indicated by the alias "opensso-sp-2" is signed by CA.
```

i. **Change the certificate alias from the default `s1as` to the new `opensso-sp-2` in the `domain.xml` file for the `ossodomain` domain.**

The Application Server configuration file is `domain.xml`.

```
<http-listener acceptor-threads="1" address="0.0.0.0"
blocking-enabled="false" default-virtual-server="server" enabled="true"
family="inet" id="http-listener-2" port="1081" security-enabled="true"
server-name="" xpowered-by="true">
<ssl cert-nickname="opensso-sp-2" client-auth-enabled="false" ssl2-enabled="false"
ssl3-enabled="true" tls-enabled="true" tls-rollback-enabled="true"/>
```

Tip – Backup `domain.xml` before modifying it.

---

11 **Modify the JVM options in your web container's configuration file using the following sub-procedure.**

OpenSSO Enterprise is deployed with an embedded configuration data store (if desired). In order for the configuration data store to be created successfully, the following JVM options should be modified in the web container's configuration file. We will be modifying `domain.xml` again for this example.

Tip – Backup `domain.xml` before modifying it.

---

a. **Change to the `config` directory.**

```
# cd /export/osso80adm/domains/ossodomain/config
```

b. **Open `domain.xml` in a text editor and make the following changes:**

- Replace **`<jvm-options>-client</jvm-options>`** with **`<jvm-options>-server</jvm-options>`**.
- Replace **`<jvm-options>-Xmx512m</jvm-options>`** with **`<jvm-options>-Xmx1024m</jvm-options>`**.

c. **Save the file and close it.**

12 **Restart the `ossodomain` domain.**

```
# cd /export/osso80adm/domains/ossodomain/bin
# ./stopserv

Server was successfully stopped.

./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

13 **Verify that the certificate used for SSL communication is the root CA certificate.**

a. **Access `https://osso2.sp-example.com:1081/index.html` from a web browser.**

b. **View the details of the certificate in the security warning to ensure that it is Issued by "OpenSSLTestCA".**

After inspecting and accepting the certificate, you should see the default `index.html` page.

c. **Close the browser.**

14 **Log out of the `osso2.sp-example.com` host machine.**

8.2 Configuring the OpenSSO Enterprise Load Balancer

The two instances of OpenSSO Enterprise are fronted by one load balancer (Load Balancer 2). Users will access OpenSSO Enterprise through the secure port 1081. Load Balancer 2 sends the user and agent requests to the server where the session originated. Secure Sockets Layer (SSL) is terminated and regenerated before a request is forwarded to the OpenSSO Enterprise servers to allow the load balancer to inspect the traffic for proper routing. Load Balancer 2 is capable of the following types of load balancing:

Cookie-based	The load balancer makes decisions based on client's cookies. The load balancer looks at the request and detects the presence of a cookie by a specific name. If the cookie is detected in the request, the load balancer routes the request to the specific server to which the cookie has been assigned. If the cookie is not detected in the request, the load balancer balances client requests among the available servers.
IP-based	This is similar to cookie-based load balancing, but the decision is based on the IP address of the client. The load balancer sends all requests from a specific IP address to the same server.
TCP	The load balancer mainstreams session affinity. This means that all requests related to a TCP session, are forwarded to the same server. In this deployment example, Load Balancer 2 forwards all requests from a single client to exactly the same server. When the session is started and maintained by one client, session affinity is guaranteed. This type of load-balancing is applicable to the TCP-based protocols.

This section assumes that you have already installed a load balancer. Before you begin, note the following:

- The load balancer hardware and software used in the lab facility for this deployment is BIG-IP® manufactured by F5 Networks. If you are using different load balancer software, see the documentation that comes with that product for detailed settings information.
- Contact your network administrator to obtain an available virtual IP address for the load balancer you want to configure.
- Know the IP address of the load balancer hardware, the URL for the load balancer login page, and a username and password for logging in to the load balancer application.

- Get the IP addresses for OpenSSO Enterprise 1 and OpenSSO Enterprise 2 by running the following command on each host machine:

  ```
  # ifconfig -a
  ```

Use the following list of procedures as a checklist for completing the task.

1. "To Request a Certificate for OpenSSO Enterprise Load Balancer 2" on page 185
2. "To Install a CA Root Certificate to OpenSSO Enterprise Load Balancer 2" on page 186
3. "To Install the Server Certificate to OpenSSO Enterprise Load Balancer 2" on page 187
4. "To Configure OpenSSO Enterprise Load Balancer 2" on page 187
5. "To Create an SSL Proxy for SSL Termination at the OpenSSO Enterprise Load Balancer 2" on page 190

▼ To Request a Certificate for OpenSSO Enterprise Load Balancer 2

You should already have a root certificate from the CA of your choice. Generate a request for a server certificate to send to the CA. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1. **Access** `https://is-f5.siroe.com`**, the BIG-IP load balancer login page, in a web browser.**

2. **Log in to the BIG-IP console as the administrator.**

3. **Click** *Configure your BIG-IP (R) using the Configuration Utility*.

4. **In the left pane, click Proxies.**

5. **Click the Cert-Admin tab.**

6. **On the SSL Certificate Administration page, click** *Generate New Key Pair/Certificate Request*.

7. **In the Create Certificate Request page, provide the following information.**

Key Identifier:	**`lb2.sp-example.com`**
Organizational Unit Name:	**`Deployment`**
Domain Name:	**`lb2.sp-example.com`**
Challenge Password:	**`password`**
Retype Password:	**`password`**

8 **Click** ***Generate Key Pair/Certificate Request*****.**

On the SSL Certificate Request page, the request is generated in the Certificate Request field.

9 **Save the text contained in the Certificate Request field to a file named `lb-2.csr`.**

10 **Log out of the console and close the browser.**

11 **Send `lb-2.csr` to the CA of your choice.**

The CA issues and returns a signed server certificate named `lb-2.cer`.

▼ To Install a CA Root Certificate to OpenSSO Enterprise Load Balancer 2

Install the CA root certificate on Load Balancer 2 to ensure that a link between it and the CA can be maintained. Use the same root certificate that you imported in "7.4 Enabling Secure Communication for the Directory Server User Data Instances" on page 146. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1 **Access `https://is-f5.example.com`, the BIG-IP load balancer login page, in a web browser.**

2 **Log in to the BIG-IP console as the administrator.**

3 **In the BIG-IP load balancer console, click Proxies.**

4 **Click the Cert-Admin tab.**

5 **Click Import.**

6 **In the Import Type field, choose Certificate, and click Continue.**

7 **Click Browse in the Certificate File field on the Install SSL Certificate page.**

8 **In the Choose File dialog, choose Browser.**

9 **Navigate to `ca.cer` and click Open.**

10 **In the Certificate Identifier field, enter `openSSLCA`.**

11 **Click Install Certificate.**

12 **On the Certificate openSSLCA page, click Return to Certificate Administration.**

The root certificate named `openSSLCA` is now included in the Certificate ID list.

▼ To Install the Server Certificate to OpenSSO Enterprise Load Balancer 2

Before You Begin This procedure assumes you have received the CA-signed server certificate requested in "To Request a Certificate for OpenSSO Enterprise Load Balancer 2" on page 185, just completed "To Install a CA Root Certificate to OpenSSO Enterprise Load Balancer 2" on page 186, and are still logged into the load balancer console.

1 **In the BIG-IP load balancer console, click Proxies.**

2 **Click the Cert-Admin tab.**

The key `lb2.sp-example.com` is in the Key List.

3 **In the Certificate ID column, click Install for `lb2.sp-example.com`.**

4 **In the Certificate File field, click Browse.**

5 **In the Choose File dialog, navigate to `lb-2.cer`, the CA-signed server certificate, and click Open.**

6 **Click Install Certificate.**

7 **On the Certificate lb2.sp-example.com page, click Return to Certificate Administration Information.**

Verify that the Certificate ID indicates `lb2.sp-example.com` on the SSL Certificate Administration page.

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

▼ To Configure OpenSSO Enterprise Load Balancer 2

1 **Access `https://is-f5.example.com`, the BIG-IP load balancer login page, in a web browser.**

2 **Log in to the BIG-IP console as the administrator.**

3 **Click** ***Configure your BIG-IP (R) using the Configuration Utility*****.**

4 Create a Pool.

A pool contains all the backend server instances.

a. In the left pane, click Pools.

b. On the Pools tab, click Add.

c. In the Add Pool dialog, provide the following information.

Pool Name	**`OpenSSO-SP-Pool`**
Load Balancing Method	Round Robin
Resources	Add the IP addresses and port numbers for both OpenSSO Enterprise host machines. **Note –** Use port number 1081.

d. Click Done.

5 Add a Virtual Server.

The virtual server presents an address to the outside world and, when users attempt to connect, it would forward the connection to the most appropriate real server.

Note – If you encounter JavaScript™ errors or otherwise cannot proceed to create a virtual server, try using Internet Explorer.

a. In the left frame, click Virtual Servers.

b. On the Virtual Servers tab, click Add.

c. In the Add a Virtual Server dialog box, provide the following information:

Address	Enter the IP address for `lb2.sp-example.com`
Service	**`1082`**

d. Continue to click Next until you reach the Pool Selection dialog box.

e. In the Pool Selection dialog box, assign the `OpenSSO-SP-Pool` Pool.

f. Click Done.

6 **Add Monitors.**

OpenSSO Enterprise comes with a JSP file named `isAlive.jsp` that can be contacted to determine if the server is down. Since we have not yet deployed OpenSSO Enterprise, `isAlive.jsp` cannot be used. In the following sub procedure, create a custom monitor that periodically accesses the Application Server instance(s). If desired, the monitor can be changed later to use `isAlive.jsp`.

a. **Click the Monitors tab**

b. **Click the Basic Associations tab**

c. **Find the IP address for `osso1.sp-example.com:1080` and `osso2.sp-example.com:1080`.**

d. **Mark the Add checkbox that corresponds to the IP address for both `osso1.sp-example.com:1080` and `osso2.sp-example.com:1080`.**

e. **At the top of the Node column, choose the `tcp` monitor.**

f. **Click Apply.**

7 **Configure the load balancer for persistence.**

a. **In the left pane, click Pools.**

b. **Click the name of the pool you want to configure; in this case, `OpenSSO-SP-Pool`.**

c. **Click the Persistence tab.**

d. **Under Persistence Type, select Passive HTTP Cookie.**

e. **Under Cookie Name, enter `amlbcookie`.**

f. **Click Apply.**

8 **In the left pane, click BIGpipe.**

9 **In the BIGpipe command window, type the following:**

```
makecookie ip-address:port
```

ip-address is the IP address of the `osso1.sp-example.com` host machine and *port* is the same machine's port number; in this case, 1081.

10 **Press Enter to execute the command.**

Something similar to `Set-Cookie: BIGipServer[poolname]=692589248.36895.0000; path=/` is displayed. Save the numbered value (in this case, 692589248.88888.0000) for use in "To Create a Site on OpenSSO Enterprise 1" on page 201.

11 **In the left pane, click BIGpipe again.**

12 **In the BIGpipe command window, type the following:**

```
makecookie ip-address:port
```

ip-address is the IP address of the `osso2.sp-example.com` host machine and *port* is the same machine's port number; in this case, 1081.

13 **Press Enter to execute the command.**

Something similar to `Set-Cookie: BIGipServer[poolname]=692589248.12345.0000; path=/` is displayed. Save the numbered value (in this case, 692589248.99999.0000) for use in "To Create a Site on OpenSSO Enterprise 1" on page 201.

14 **Log out of the load balancer console.**

▼ To Create an SSL Proxy for SSL Termination at the OpenSSO Enterprise Load Balancer 2

SSL communication is terminated at Load Balancer 2. The request is then re-encrypted and securely forwarded to OpenSSO Enterprise. When clients send an SSL-encrypted request to Load Balancer 2, it decrypts the request and re-encrypts it before sending it on to the OpenSSO Enterprise SSL port. Load Balancer 2 also encrypts the responses it receives back from OpenSSO Enterprise, and sends these encrypted responses back to the client. Towards this end create an *SSL proxy* for SSL termination and regeneration.

Before You Begin Use the same root certificate that you imported in "7.4 Enabling Secure Communication for the Directory Server User Data Instances" on page 146. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 36.

1 **Access** `https://is-f5.example.com`**, the BIG-IP load balancer login page, in a web browser.**

2 **Log in to the BIG-IP console as the administrator.**

3 **Click** ***Configure your BIG-IP (R) using the Configuration Utility*****.**

4 **In the left pane, click Proxies.**

5 **Under the Proxies tab, click Add.**

6 **In the Add Proxy dialog, provide the following information.**

Proxy Type:	Check the SSL and ServerSSL checkbox.
Proxy Address:	The IP address of Load Balancer 2.
Proxy Service:	`1081` The secure port number
Destination Address:	The IP address of Load Balancer 2.
Destination Service:	`1082` The non-secure port number
Destination Target:	Choose `Local Virtual Server`.
SSL Certificate:	Choose `lb2.sp-example.com`.
SSL Key:	Choose `lb2.sp-example.com`.
Enable ARP:	Check this checkbox.

7 **Click Next.**

8 **On the page starting with "Insert HTTP Header String," change to Rewrite Redirects and choose Matching.**

9 **Click Next.**

10 **On the page starting with "Server Chain File," change to Server Trusted CA's File, select "ca.cer" from the drop-down list.**

11 **Click Done.**

The new proxy server is added to the Proxy Server list.

12 **Log out of the load balancer console.**

13 **Access `https://lb2.sp-example.com:1081/index.html` from a web browser.**

If the Application Server index page is displayed, you can access it using the new proxy server port number and the load balancer is configured properly.

Tip – A message may be displayed indicating that the browser doesn't recognize the certificate issuer. If this happens, install the CA root certificate in the browser so that the browser recognizes the certificate issuer. See your browser's online help system for information on installing a root CA certificate.

14 Close the browser.

8.3 Deploying and Configuring OpenSSO Enterprise 1 and OpenSSO Enterprise 2

An OpenSSO Enterprise WAR will be deployed in the installed Application Server containers on both OpenSSO Enterprise host machines. Additionally, you will configure the deployed applications. Use the following list of procedures as a checklist for completing the tasks.

1. "To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine" on page 192
2. "To Deploy the OpenSSO Enterprise WAR as OpenSSO Enterprise 1" on page 194
3. "To Copy the OpenSSO Enterprise WAR to the OpenSSO Enterprise 2 Host Machine" on page 196
4. "To Deploy the OpenSSO Enterprise WAR File as OpenSSO Enterprise 2" on page 197
5. "To Configure OpenSSO Enterprise 1" on page 198
6. "To Configure OpenSSO Enterprise 2" on page 200

▼ To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine

1 As a root user, log in to the `osso1.sp-example.com` **host machine.**

2 Create a directory into which the OpenSSO Enterprise ZIP file can be downloaded and change into it.

```
# mkdir /export/OSSO_BITS
# cd /export/OSSO_BITS
```

3 Download the OpenSSO Enterprise ZIP file from `http://www.sun.com/download/`**.**

4 Unzip the downloaded file.

```
# unzip opensso_enterprise_80.zip
# cd /export/OSSO_BITS/opensso
# ls -al

total 68
drwxr-xr-x  14 root     root     512 Sep 8 11:13 ./
drwxrwxr-x   3 root     root     512 Sep 15 13:06 ../
-rw-r--r--   1 root     root    1349 Sep 8 10:58 README
drwxr-xr-x   6 root     root     512 Sep 8 11:15 deployable-war/
drwxr-xr-x   2 root     root     512 Sep 8 11:13 docs/
drwxr-xr-x   2 root     root     512 Sep 8 11:13 fedlet/
drwxr-xr-x   5 root     root     512 Sep 8 11:11 integrations/
drwxr-xr-x   2 root     root     512 Sep 8 11:13 ldif/
drwxr-xr-x   4 root     root     512 Sep 8 11:13 libraries/
-rw-r--r--   1 root     root   17003 Sep 8 10:58 license.txt
drwxr-xr-x   2 root     root     512 Sep 8 11:13 migration/
drwxr-xr-x   2 root     root     512 Sep 8 11:13 patches/
drwxr-xr-x   2 root     root     512 Sep 8 11:13 samples/
drwxr-xr-x   2 root     root     512 Sep 8 11:14 tools/
drwxr-xr-x   8 root     root     512 Sep 8 11:13 upgrade/
drwxr-xr-x   2 root     root    2048 Sep 8 11:11 xml/
```

5 Switch to the non-root user.

```
# su osso80adm
```

6 Create a staging area in the non-root user directory into which the WAR will be exploded.

```
# cd /export/osso80adm
# mkdir osso-staging
```

Tip – In the staging area, after exploding the WAR, you can modify the WAR contents to suit your needs, generate a new WAR, and deploy it on any number of remote host computers. Whenever you need to make changes to the WAR, you maintain the changes in this one staging area, and redeploy the modified WAR as many times as you want, on as many host machines as you need.

7 Explode the WAR file.

```
# cd osso-staging
# jar xvf /export/OSSO_BITS/opensso/deployable-war/opensso.war
```

8 Make the following modifications to the `bootstrap.properties` file.

By default, during the WAR deployment, OpenSSO Enterprise creates a bootstrap file in the user's home directory. The `bootstrap.properties` file points to the directory where all the OpenSSO Enterprise configurations will be created. With these modifications, OpenSSO Enterprise will create the bootstrap file in the directory you specify; in this case, `/export/osso80adm/config`. `bootstrap.properties` is located in `/export/osso80adm/osso-staging/WEB-INF/classes`.

- Uncomment the line that reads #configuration.dir=.
- Add the following value to the configuration.dir= property so it reads as follows.

```
configuration.dir=/export/osso80adm/config
```

9 **Regenerate the WAR.**

```
# cd /export/osso80adm/osso-staging
# jar cvf ../opensso.war *
```

A new WAR file is created, including the modified bootstrap.properties.

10 **Verify that the new WAR was created in the proper location and with the appropriate permissions.**

```
# cd /export/osso80adm/osso-staging
# /bin/rm -rf *
# jar xvf ../opensso.war
# ls -al

total 498
drwxr-xr-x 7 osso80adm staff 512 Aug 5 13:44 .
drwxr-xr-x 12 root sys 512 Aug 5 11:11 ..
-rw------- 1 osso80adm staff 779 Aug 5 14:56 .asadmintruststore
drwx------ 2 osso80adm staff 512 Aug 5 14:44 .gconf
drwx------ 2 osso80adm staff 512 Aug 5 14:44 .gconfd
-rw-r--r-- 1 osso80adm staff 144 Aug 5 17:02 .profile
drwx------ 3 osso80adm staff 512 Aug 5 11:20 .sunw
drwxr-xr-x 3 osso80adm staff 512 Aug 5 14:55 domains
drwxr-xr-x 21 osso80adm staff 1024 Aug 5 13:43 osso-staging
-rw-r--r-- 1 osso80adm staff 68884903 Aug 5 13:45 opensso.war
-rw-r--r-- 1 osso80adm staff 136 Aug 5 17:02 local.cshrc
-rw-r--r-- 1 osso80adm staff 157 Aug 5 17:02 local.login
-rw-r--r-- 1 osso80adm staff 174 Aug 5 17:02 local.profile
```

Note – The opensso.war file is owned by osso80adm.

▼ To Deploy the OpenSSO Enterprise WAR as OpenSSO Enterprise 1

Before You Begin This procedure assumes you have just completed “To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine” on page 192 and are still logged into the osso1.sp-example.com host machine

1 **On the `osso1.sp-example.com` host machine, switch to the non-root user `osso80adm`.**

```
# /bin/su osso80adm
```

2 **Start the `ossodomain` domain.**

```
# cd /export/osso80adm/domains/ossodomain/bin
# ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

3 **Run `asadmin deploy` to deploy the OpenSSO Enterprise WAR.**

```
# cd /opt/SUNWappserver91/bin
# ./asadmin deploy --user domain2adm --host osso1.sp-example.com
--port=8989 --contextroot opensso --name opensso --target server
/export/osso80adm/opensso.war

Please enter the admin password> domain2pwd

Command deploy executed successfully.
```

4 **List the contents of the `j2ee-modules` directory to verify that the WAR file was successfully deployed.**

```
# cd /export/osso80adm/domains/ossodomain/applications/j2ee-modules
# ls -al

total 6
drwxr-xr-x   3 osso80adm staff      512 Aug 5 14:01 .
drwxr-xr-x   6 osso80adm staff      512 Aug 5 14:55 ..
drwxr-xr-x  21 osso80adm staff     1024 Aug 5 14:01 opensso
```

`opensso` exists in the directory and is owned by the non-root user `osso80adm`.

5 **Log out of the `osso1.sp-example.com` host machine.**

▼ To Copy the OpenSSO Enterprise WAR to the OpenSSO Enterprise 2 Host Machine

Before You Begin This procedure assumes you have completed "To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine" on page 192.

1 **As a root user, log in to the `osso2.sp-example.com` host machine.**

2 **Switch to the non-root user `osso80adm`.**

```
# /bin/su osso80adm
```

3 **Change into the `osso80adm` directory.**

```
# cd /export/osso80adm
```

4 **Copy `opensso.war` from the `osso1.sp-example.com` host machine to the `osso80adm` directory.**

5 **Verify that the WAR file was copied into the proper location and with the appropriate permissions.**

```
# ls -al

total 130552
drwxr-xr-x   6 osso80adm staff        512 Sep 5 14:14 .
drwxr-xr-x   8 root      sys          512 Sep 5 10:54 ..
-rw-r--r--   1 osso80adm staff         70 Sep 5 14:13 .asadminpass
-rw-------   1 osso80adm staff        778 Sep 5 14:12 .asadmintruststore
drwx------   2 osso80adm staff        512 Sep 5 13:15 .gconf
drwx------   2 osso80adm staff        512 Sep 5 13:26 .gconfd
-rw-r--r--   1 osso80adm staff        144 Sep 5 15:00 .profile
drwx------   3 osso80adm staff        512 Sep 5 15:26 .sunw
drwxr-xr-x   3 osso80adm staff        512 Sep 5 14:12 domains
-rw-r--r--   1 osso80adm staff   68884903 Sep 5 14:14 opensso.war
-rw-r--r--   1 osso80adm staff        136 Sep 5 15:00 local.cshrc
-rw-r--r--   1 osso80adm staff        157 Sep 5 15:00 local.login
-rw-r--r--   1 osso80adm staff        174 Sep 5 15:00 local.profile
```

`opensso.war` exists in the directory and is owned by `osso80adm`.

▼ To Deploy the OpenSSO Enterprise WAR File as OpenSSO Enterprise 2

Before You Begin This procedure assumes you have just completed "To Copy the OpenSSO Enterprise WAR to the OpenSSO Enterprise 2 Host Machine" on page 196 and are still logged into the osso2.sp-example.com host machine

1 On the osso2.sp-example.com host machine, switch to the non-root user osso80adm.

```
# /bin/su osso80adm
```

2 Start the ossodomain domain.

```
# cd /export/osso8/domains/ossodomain/bin
# ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

3 Run asadmin deploy to deploy the OpenSSO Enterprise WAR file.

```
# cd /opt/SUNWappserver91/bin
# ./asadmin deploy --user domain2adm --host osso2.sp-example.com
--port=8989 --contextroot opensso --name opensso --target server
/export/osso80adm/opensso.war

Please enter the admin password> domain2pwd

Command deploy executed successfully.
```

4 List the contents of the j2ee-modules directory to verify that the WAR file was successfully deployed.

```
# cd /export/osso80adm/domains/ossodomain/applications/j2ee-modules
# ls -al

total 6
drwxr-xr-x   3 osso80adm staff      512 Sep 5 14:01 .
drwxr-xr-x   6 osso80adm staff      512 Sep 5 14:55 ..
drwxr-xr-x  21 osso80adm staff     1024 Sep 5 14:01 opensso
```

opensso exists in the directory and is owned by the non-root user osso80adm.

5 **Log out of the `osso2.sp-example.com` host machine.**

▼ To Configure OpenSSO Enterprise 1

1 **Access `https://osso1.sp-example.com:1081/opensso` from a web browser.**

The OpenSSO Enterprise Configurator page is displayed for first time access.

2 **Select Create New Configuration under Custom Configuration on the Configurator page.**

The OpenSSO Enterprise Custom Configuration Wizard is displayed.

3 **Provide the following information for the Default User [amAdmin] in Step 1: General and click Next.**

Password	**`ossoadmin`**
Confirm	**`ossoadmin`**

4 **Accept the default values in Step 2: Server Settings and click Next**

5 **Do the following in Step 3: Configuration Store and click Next**

a. Select First Instance.

b. Select Embedded DS as the configuration data store.

c. Accept the default values for the Port, Encryption Key, and Root Suffix fields.

6 **Select Remote Directory in Step 4: User Store Settings, provide the following information and click Next**

SSL Enabled	Check the box.
Directory Name	**`lb2.sp-example.com`**
Port	**`489`**
Root Suffix	**`o=spusers.com`**
Password	**`dsmanager`**
Store Type	Select Generic LDAP.

7 **Select No in Step 5: Site Configuration and click Next.**

8 **Provide the following information for the Default Agent User [amldapuser] in Step 6: Default Agent User and click Next.**

Password	**agentuser**
Confirm	**agentuser**

9 **Click Create Configuration on the Summary page.**

The Configuration Complete page is displayed after configuration is completed.

10 **Click Proceed to Login on the Configuration Complete page.**

11 **Log in to the OpenSSO Enterprise console as the administrator.**

User Name:	**amadmin**
Password:	**ossoadmin**

If authentication succeeds and the OpenSSO Enterprise console is displayed, OpenSSO Enterprise has successfully accessed the embedded configuration data store.

12 **(Optional) To verify that the `config` directory and the supporting `bootstrap` directory have been created with the proper permissions, do the following.**

a. **As a root user, log in to the `osso1.sp-example.com` host machine.**

b. **Examine the file system.**

```
# cd /export/osso80adm
# ls -al

total 130556
drwxr-xr-x   8 osso80adm staff        512 Sep  6 19:32 .
drwxr-xr-x  14 root      sys          512 Sep  6 09:07 ..
-rw-r--r--   1 osso80adm staff         70 Sep 27 14:01 .asadminpass
-rw-------   1 osso80adm staff       1527 Sep  6 18:27 .asadmintruststore
-rw-r--r--   1 osso80adm staff        144 Sep 11 17:02 .profile
drwx------   3 osso80adm staff        512 Sep 24 11:20 .sunw
drwxr-xr-x   4 osso80adm staff        512 Sep  6 19:34 config
drwxr-xr-x   4 osso80adm staff        512 Sep  6 18:26 domains
-rw-r--r--   1 osso80adm staff        136 Sep 11 17:02 local.cshrc
-rw-r--r--   1 osso80adm staff        157 Sep 11 17:02 local.login
-rw-r--r--   1 osso80adm staff        174 Sep 11 17:02 local.profile
```

The `config` directory was created and is owned by non-root user `osso80adm`.

c. **Log out of the `osso1.sp-example.com` host machine.**

▼ To Configure OpenSSO Enterprise 2

1 **Access** `https://osso2.sp-example.com:1081/opensso` **from a web browser.**

The OpenSSO Enterprise Configurator page is displayed for first time access.

2 **Select Create New Configuration under Custom Configuration on the Configurator page.**

The OpenSSO Enterprise Custom Configuration Wizard is displayed.

3 **Provide the following information for the Default User [amAdmin] in Step 1: General and click Next.**

Password	**`ossoadmin`**
Confirm	**`ossoadmin`**

4 **Accept the default values in Step 2: Server Settings and click Next**

5 **Do the following in Step 3: Configuration Store and click Next**

a. **Select Add to Existing Deployment as the configuration data store.**

b. **Server URL: `https://osso2.sp-example.com:1081/opensso`**

6 **Select No in Step 5: Site Configuration and click Next.**

7 **Click Create Configuration on the Summary page.**

The Configuration Complete page is displayed after configuration is completed.

8 **Click Proceed to Login on the Configuration Complete page.**

9 **Log in to the OpenSSO Enterprise console as the administrator.**

User Name:	**`amadmin`**
Password:	**`ossoadmin`**

If authentication succeeds and the OpenSSO Enterprise console is displayed, OpenSSO Enterprise has successfully accessed the embedded configuration data store.

10 **(Optional) To verify that the** `config` **directory and the supporting** `bootstrap` **directory have been created with the proper permissions, do the following.**

a. **As a root user, log in to the** `osso2.sp-example.com` **host machine.**

b. **Examine the file system.**

```
# cd /export/osso80adm
# ls -al

total 130556
drwxr-xr-x   8 osso80adm staff        512 Aug  6 19:32 .
drwxr-xr-x  14 root      sys          512 Aug  6 09:07 ..
-rw-r--r--   1 osso80adm staff         70 Mar 27 14:01 .asadminpass
-rw-------   1 osso80adm staff       1527 Aug  6 18:27 .asadmintruststore
-rw-r--r--   1 osso80adm staff        144 Mar 11 17:02 .profile
drwx------   3 osso80adm staff        512 Mar 24 11:20 .sunw
drwxr-xr-x   4 osso80adm staff        512 Aug  6 19:34 config
drwxr-xr-x   4 osso80adm staff        512 Aug  6 18:26 domains
-rw-r--r--   1 osso80adm staff        136 Mar 11 17:02 local.cshrc
-rw-r--r--   1 osso80adm staff        157 Mar 11 17:02 local.login
-rw-r--r--   1 osso80adm staff        174 Mar 11 17:02 local.profile
```

The `config` directory was created and is owned by non-root user `osso80adm`.

c. **Log out of the** `osso2.sp-example.com` **host machine.**

8.4 Configuring the OpenSSO Enterprise Platform Service

The Platform Service provides centralized configuration management for an OpenSSO Enterprise deployment. In this procedure, you configure the two OpenSSO Enterprise servers to work as a single unit. Once configured as a *site*, all client requests go through the configured load balancer. Use the following list of procedures as a checklist for completing this task.

1. "To Create a Site on OpenSSO Enterprise 1" on page 201
2. "To Verify that the OpenSSO Enterprise Site was Configured Properly" on page 204

▼ To Create a Site on OpenSSO Enterprise 1

It is **not** necessary to repeat this procedure on OpenSSO Enterprise 2.

1 **Access** `https://osso1.sp-example.com:1081/opensso/console` **in a web browser.**

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

Username **`amadmin`**

Password **`ossoadmin`**

3 **Under the Configuration tab, click Servers and Sites.**

The Servers and Sites page is displayed.

4 **Click New under Sites.**

The New Site properties page is displayed.

5 **Enter the following values for the load balancer and click OK.**

Name	**sp-site**
Primary URL	**https://lb2.sp-example.com:1081/opensso**

A new site called `sp-site` is displayed in the Sites list.

6 **Click on the `https://osso1.sp-example.com:1081/opensso` server entry under the Servers list.**

The Edit `https://osso1.sp-example.com:1081/opensso` page is displayed.

7 **Assign `sp-site` from the Parent Site drop down list and click Save.**

8 **Click the Advanced tab.**

9 **Enter the number generated for the `osso1.sp-example.com` host machine as the value of the `com.iplanet.am.lbcookie.value` property and click Save.**

The number was generated using the `makecookie` command in "To Configure OpenSSO Enterprise Load Balancer 2" on page 187.

10 **Click Back to Server and Sites.**

11 **Click on the `https://osso2.sp-example.com:1081/opensso` server entry under the Servers list.**

The Edit `https://osso2.sp-example.com:1081/opensso` page is displayed.

12 **Assign `sp-site` from the Parent Site drop down list and click Save.**

13 **Click the Advanced tab.**

14 **Enter the number generated for the `osso2.sp-example.com` host machine as the value of the `com.iplanet.am.lbcookie.value` property and click Save.**

The number was generated using the `makecookie` command in "To Configure OpenSSO Enterprise Load Balancer 2" on page 187.

15 **Click Back to Server and Sites.**

Note – You should see sp-site under the Site Name column for both servers.

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

17 **As a root user, log in to the** `osso1.sp-example.com` **host machine.**

18 **Restart OpenSSO Enterprise for the changes to take effect.**

```
# /bin/su osso80adm
# cd /export/osso80adm/domains/ossodomain/bin
# ./stopserv; ./startserv

Server was successfully stopped.

admin username:  domain2adm

admin password:  domain2pwd

master password: domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

19 **As a root user, log in to the** `osso2.sp-example.com` **host machine.**

20 **Restart the web container for the changes to take effect.**

```
# /bin/su osso80adm
# cd /export/osso80adm/domains/ossodomain/bin
# ./stopserv; ./startserv

Server was successfully stopped.

admin username:  domain2adm

admin password:  domain2pwd

master password: domain2master

Redirecting output to /export/osso80adm/domains/ossodomain/logs/server.log
```

21 **Log out of both OpenSSO Enterprise host machines.**

▼ To Verify that the OpenSSO Enterprise Site was Configured Properly

1 **Access the load balancer at** `https://lb2.sp-example.com:1081/opensso/UI/Login`**.**

If an error message is displayed indicating that the browser cannot connect to either `osso1.sp-example.com` or `osso2.sp-example.com`, the site configuration is not correct. If the site configuration is correct, all browser interactions will occur as expected.

2 **When the OpenSSO Enterprise login page is displayed, verify that the browser URL still contains the Primary Site URL for the load balancer.**

If it does not contain the Site URL, the site configuration is incorrect. If the site configuration is correct, all browser interactions will occur through the secure Site URL.

3 **Log in to the OpenSSO Enterprise console as the administrator.**

User Name:	**`amadmin`**
Password:	**`ossoadmin`**

A successful login occurs when the site configuration is correct.

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

8.5 Configuring OpenSSO Enterprise for SAML v2

Configure OpenSSO Enterprise on the service provider side to recognize the Directory Server LDAP schema previously modified for SAML v2 attributes.

▼ To Configure OpenSSO Enterprise for the Modified LDAP Schema

Before You Begin This procedure assumes you have completed "7.3 Modifying the Directory Server Schema" on page 144.

1 **Access** `https://lb4.sp-example.com:1081/opensso/console` **from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

The Common Tasks tab is displayed.

3 **Click the Access Control tab and `/ (Top-level Realm)` on the Access Control page.**

4 **Click the Data Stores tab.**

5 **Under the Data Stores tab, click `embedded`.**

The Generic LDAPv3 page is displayed.

6 **Add the following values to properties on the Generic LDAPv3 page.**

- Type **`sunFMSAML2NameIdentifier`** in the New Value box of the LDAP User Object Class property and click Add.
- Add the following values to the LDAP User Attribute property.
  - Type **`sun-fm-saml2-nameid-infokey`** in the New Value box and click Add.
  - Type **`sun-fm-saml2-nameid-info`** in the New Value box and click Add.

7 **Click Save on the Generic LDAPv3 page.**

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

CHAPTER 9

Configuring OpenSSO Enterprise Realms for User Authentication

This chapter contains instructions on configuring OpenSSO Enterprise to use the external user data store (set up in Chapter 4, "Installing Sun Java System Directory Server and Creating Instances for User Data") for authentication credentials. This is done by modifying the top-level realm or, alternately, configuring a sub realm for the external users and creating an authentication chain. Choose either of the sections listed to configure OpenSSO Enterprise for user authentication.

- "6.1 Modifying the Top-Level Realm for Test Users" on page 117
- "6.2 Creating and Configuring a Sub Realm for Test Users" on page 119

⚠ **Caution –** Do **not** do both.

9.1 Modifying the Top-Level Realm for Test Users

At this point in the deployment, the root realm (by default, `/ (Top Level Realm)`) is configured to authenticate special OpenSSO Enterprise accounts (for example, `amadmin` and agents) against the embedded configuration data store. Since the external user data store is an instance of Directory Server and not part of the embedded configuration data store, we modify the configuration details of the top-level realm to include the user data stores schema, allowing OpenSSO Enterprise to recognize users in the external user data store. Use the following list of procedures as a checklist for completing this task.

1. "To Modify the Top-Level Realm for User Authentication" on page 118
2. "To Verify that a User Can Successfully Authenticate" on page 119

▼ To Modify the Top-Level Realm for User Authentication

1 **Access** `https://osso1.sp-example.com:1081/opensso/console` **in a web browser.**

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

User Name: **`amadmin`**

Password: **`ossoadmin`**

3 **Click the Access Control tab.**

4 **Click** `/ (Top Level Realm)`**, the root realm, under the Access Control tab.**

5 **Click the Data Stores tab.**

The `embedded` data store link is displayed.

6 **Click** `embedded`**.**

The `Generic LDAPv3` properties page is displayed.

7 **On the** `Generic LDAPv3` **properties page, set the following attribute values and click Save.**

LDAP People Container Naming Attribute
Enter `ou`.

LDAP Groups Container Value
Enter `Groups`.

LDAP Groups Container Naming Attribute
Enter `ou`.

LDAP People Container Value
Enter `users`.

Note – If this field is empty, the search for user entries will start from the root suffix.

8 **Click Back to Data Stores.**

9 **(Optional) Click the Subjects tab to verify that the test users are now displayed.**

`spuser` is displayed under Users (as well as others created during OpenSSO Enterprise configuration).

10 **Click the Authentication tab.**

11 **Click the Advanced Properties link under General.**

The Core Realm Attributes page is displayed.

12 **Change the value of User Profile to Ignored.**

This new value specifies that a user profile is not required by the Authentication Service in order to issue a token after successful authentication. This modification is specific to this deployment example because the OpenSSO Enterprise schema and the Directory Server schema have not been mapped.

13 **Click Save.**

14 **Click Back to Authentication.**

15 **Click Back to Access Control.**

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

▼ To Verify that a User Can Successfully Authenticate

You should be able to log in successfully as the test user.

1 **Access** `https://osso1.sp-example.com:1081/opensso/UI/Login` **in a web browser.**

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

User Name: **`spuser`**

Password: **`spuser`**

You should be able to log in successfully and see a page with a message that reads *You're logged in*. Since the User Profile attribute was previously set to `Ignored`, the user's profile is not displayed after a successful login. If the login is not successful, watch the Directory Server access log to troubleshoot the problem.

9.2 Creating and Configuring a Sub Realm for Test Users

At this point in the deployment, `/ (Top Level Realm)`, the root realm, is configured to authenticate special OpenSSO Enterprise accounts (for example, `amadmin` and agents) against the embedded configuration data store. Since the external user data store is an instance of Directory Server and not part of the embedded configuration data store, we create a sub realm and modify the configuration details to include the external user data stores schema, allowing OpenSSO Enterprise to recognize users in the Directory Server instances. The sub realm creates a demarcation between OpenSSO Enterprise configuration and administrative data and the user data. Use the following list of procedures as a checklist for completing this task.

1. "To Create a Sub Realm" on page 120
2. "To Change the User Profile Configuration for the Sub Realm" on page 121
3. "To Modify the Sub Realm for User Authentication" on page 121
4. "To Verify That the Sub Realm Can Access the External User Data Store" on page 122
5. "To Verify That the Sub Realm Subjects Can Successfully Authenticate" on page 123

▼ To Create a Sub Realm

When a sub realm is created it inherits configuration data (including which user data store to access) from the root realm (by default, `/ (Top Level Realm)`) and uses said data to authenticate users. The user data store can be modified per sub realm. In this deployment, we use the inherited `Generic LDAPv3` data store.

1 Access `https://osso1.sp-example.com:1081/opensso/console` **from a web browser.**

2 Log in to the OpenSSO Enterprise console as the administrator.

User Name: **`amadmin`**

Password: **`ossoadmin`**

3 Click the Access Control tab.

4 Click New to create a new realm.

The New Realm page is displayed.

5 Set the following attribute values on the New Realm page.

Name
Enter **`users`**.

Realm/DNS Aliases
Enter `users` in the New Value field and click Add.

6 Click OK.

The `users` realm is listed as a sub realm of `/ (Top Level Realm)`, the root realm.

▼ To Change the User Profile Configuration for the Sub Realm

Before You Begin This procedure assumes you have just completed "To Create a Sub Realm" on page 120 and are still logged in to the OpenSSO Enterprise console.

1 **Under the Access Control tab, click the `users` realm.**

2 **Click the Authentication tab.**

3 **Click the Advanced Properties link under General.**

The Core Realm Attributes page is displayed.

4 **Change the value of User Profile to Ignored.**

This new value specifies that a user profile is not required by the Authentication Service in order to issue a token after successful authentication.

5 **Click Save.**

6 **Click Back to Access Control.**

▼ To Modify the Sub Realm for User Authentication

Before You Begin This procedure assumes you have just completed "To Change the User Profile Configuration for the Sub Realm" on page 121 and are still logged in to the OpenSSO Enterprise console.

1 **Click `users`, the sub realm, under the Access Control tab.**

2 **Click the Data Stores tab.**

The `embedded` data store link is displayed.

3 **Click `embedded`.**

The `Generic LDAPv3` properties page is displayed.

4 **On the `Generic LDAPv3` properties page, set the following attribute values and click Save.**

LDAP People Container Naming Attribute
Enter `ou`.

LDAP Groups Container Value
Enter `Groups`.

LDAP Groups Container Naming Attribute
Enter `ou`.

LDAP People Container Value
Enter `users`.

Note – If this field is empty, the search for user entries will start from the root suffix.

5 **Click Back to Data Stores.**

6 **(Optional) Click the Subjects tab to verify that the test users are now displayed.**

`spuser` is displayed under Users (as well as others created during OpenSSO Enterprise configuration).

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

▼ To Verify That the Sub Realm Can Access the External User Data Store

This optional procedure is to verify the modifications.

1 **Access** `https://osso1.sp-example.com:1081/opensso/console` **from a web browser.**

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

User Name: **`amadmin`**

Password: **`ossoadmin`**

3 **Click on the Access Control tab**

4 **Click on the** `users` **sub realm.**

5 **Click on the Subjects tab.**

`idpuser` is displayed under Users.

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

▼ To Verify That the Sub Realm Subjects Can Successfully Authenticate

1 **Access `https://osso1.sp-example.com:1081/opensso/UI/Login?realm=users` from a web browser.**

The parameter `realm=users` specifies the realm to use for authentication. At this point, a user can log in against Directory Server only if the `realm` parameter is defined in the URL.

2 **Log in to OpenSSO Enterprise with as a test user.**

User Name	**`spuser`**
Password	**`spuser`**

You should be able to log in successfully and see a page with a message that reads *You're logged in*. Since the User Profile attribute was set to `Ignored`, the user's profile is not displayed after a successful login. If the login is not successful, watch the Directory Server access log to troubleshoot the problem.

CHAPTER 10

Configuring the Service Provider Protected Resource Host Machine

In this deployment, protected resources are hosted on one machine that contains two installed web containers (one Sun Java™ System Web Server and one BEA WebLogic Server application server) and the appropriate policy agent for each (a web policy agent and a J2EE policy agent, respectively). The policy agents are configured to access the OpenSSO Enterprise Load Balancer 4. This chapter contains the following sections:

- "10.1 Installing the J2EE Container and J2EE Policy Agent on Protected Resource 1" on page 215
- "10.2 Installing the Web Server and Web Policy Agent on Protected Resource 1" on page 233

10.1 Installing the J2EE Container and J2EE Policy Agent on Protected Resource 1

Download the BEA WebLogic Server bits to the Protected Resource 1 host machine (`pr1.sp-example.com`) and install the application server. Additionally, download, install and configure the appropriate J2EE policy agent. Use the following list of procedures as a checklist for completing this task.

1. "To Install BEA WebLogic Server on Protected Resource 1" on page 216
2. "To Configure BEA WebLogic Server on Protected Resource 1" on page 217
3. "To Import a Certificate Authority Root Certificate to Protected Resource 1" on page 221
4. "To Install the J2EE Policy Agent on Protected Resource 1" on page 222
5. "To Deploy and Start the J2EE Policy Agent Housekeeping Application" on page 226
6. "To Deploy the J2EE Policy Agent Sample Application" on page 228
7. "To Configure the J2EE Policy Agent to Bypass Application Server Administrator Authentication" on page 229
8. "To Enable the J2EE Policy Agent to Run in SSO Only Mode" on page 230
9. "To Configure the J2EE Policy Agent for SAML v2 Communication" on page 231

▼ To Install BEA WebLogic Server on Protected Resource 1

BEA WebLogic Server is the application server used as the J2EE web container on Protected Resource 1.

Before You Begin Ensure that your machine is properly patched. Refer to the BEA web site to make sure that your system has the recommended patches.

1 As a root user, log into the `pr1.sp-example.com` host machine.

2 Create a directory into which you can download the WebLogic Server bits and change into it.

```
# mkdir /export/BEAWL10
# cd /export/BEAWL10
```

3 Download the WebLogic Server bits from `http://commerce.bea.com/`.

For this deployment, we download the Solaris version.

```
# ls -al

total 294548
drwxr-xr-x   2 root     root         512 Aug  7 13:23 .
drwxr-xr-x   3 root     sys          512 Aug  7 13:16 ..
-rw-r--r--   1 root     root     656834948 Aug  7 13:24 server100_solaris32.bin
```

4 Run the installer.

```
# ./server100_solaris32.bin
```

5 When prompted, do the following:

`The Welcome screen is displayed.`	Click Next.
`Accept the License agreement`	Select Yes and click Next.
`Create a new BEA Home`	Type **`/usr/local/bea`** and click Next.
`Select "Custom"`	Click Next.

`Deselect the following:`
`- Workshop for WebLogic Platform`
	Click Next.

`Choose Product Installation Directories`	Type **`/usr/local/bea/weblogic10`** and click Next.
`Installation Complete`	Deselect `Run Quickstart` and click Done.

6 **(Optional) Verify that the application was correctly installed.**

```
# cd /usr/local/bea
# ls -al

total 90
drwxr-xr-x   7 root     root         512 Jul 15 11:59 .
drwxr-xr-x   4 root     root         512 Jul 15 11:58 ..
-rwxr-xr-x   1 root     root         826 Jul 15 11:59 UpdateLicense.sh
-rw-r--r--   1 root     root          14 Jul 15 11:59 beahomelist
drwxr-xr-x   6 root     root         512 Jul 15 11:59 jdk150_06
-rw-r--r--   1 root     root       12447 Jul 15 11:59 license.bea
drwxr-xr-x   2 root     root         512 Jul 15 11:59 logs
drwxr-xr-x   6 root     root        6656 Jul 15 11:58 modules
-rw-r--r--   1 root     root       15194 Jul 15 11:59 registry.dat
-rw-r--r--   1 root     root        1077 Jul 15 11:59 registry.xml
drwxr-xr-x   4 root     root         512 Jul 15 12:01 utils
drwxr-xr-x  10 root     root         512 Jul 15 11:59 weblogic10
```

▼ To Configure BEA WebLogic Server on Protected Resource 1

Before You Begin This procedure assumes you have just completed "To Install BEA WebLogic Server on Protected Resource 1" on page 216 and are still logged into the host machine as the root user.

1 **Run the WebLogic Server configuration script.**

```
# cd /usr/local/bea/weblogic10/common/bin
# ./config.sh
```

2 **When prompted, do the following:**

`Select "Create a new Weblogic domain"`	Click Next.
`Select "Generate a domain configured automatically to support the following BEA products:"`	Click Next.
`Configure Administrator Username and Password`	Enter the following and click Next. ■ Username: **weblogic** ■ Password: **bea10admin** ■ Confirm Password: **bea10admin**
`Select "Prduction Mode" and "BEA Supplied JDK's" (Sun SDK 1.5.0_06@/usr/local/bea/jdk150_06)`	Click Next.
`Customize Environment and Services Settings`	Select yes and click Next.

`Configure the Administration Server`	Accept the default values and click Next.
`Configure Managed Servers`	Select Add, enter the following values, and click Next. ■ Name: **`ApplicationServer-1`** ■ Listen Port: **`1081`**
`Configure Clusters`	Accept the default values and click Next.
`Configure Machines`	Select the Unix Machine tab, then select Add, type **`pr1`** and click Next.
`Assign Servers to Machines`	From the left panel select *AdminServer* and *ApplicationServer-1*. From the right panel select *pr-1*. Click - -> and then click Next.
`Review WebLogic Domain`	Click Next.
`Create WebLogic Domain`	Add the following and click Create. ■ Domain name: **`pr1`** ■ Domain Location: **`/usr/local/bea/user_projects/domains`** (default)
`Creating Domain`	Click Done.

3 **Start *AdminServer*, the WebLogic administration server.**

```
# cd /usr/local/bea/user_projects/domains/pr1
# ./startWebLogic.sh
```

When prompted, type the following credentials.

Username	**`weblogic`**
Password	**`bea10admin`**

4 **Run the `netstat` command to verify that the port is open and listening.**

```
# netstat -an | grep 7001

XXX.XX.XX.101.7001        *.*               0      0 49152      0 LISTEN
XXX.X.X.1.7001            *.*               0      0 49152      0 LISTEN
```

Note – You can also access the administration console by pointing a web browser to `http://pr1.sp-example.com:7001/console`.

5 **Change to the `AdminServer` directory.**

```
# cd /usr/local/bea/user_projects/domains/pr1/servers/AdminServer
```

6 **Create a security directory and change into it.**

```
# mkdir security
# cd security
```

7 **Create a `boot.properties` file for the WebLogic Server administration server administrator credentials.**

The administration server administrative user and password are stored in `boot.properties`. Application Server 1 uses this information during startup. WebLogic Server encrypts the file, so there is no security risk even if you enter the user name and password in clear text.

```
# cat > boot.properties
username=weblogic
password=bea10admin
```

Hit Control D to terminate the command

```
^D
```

8 **Restart WebLogic to encrypt the username and password in `boot.properties`.**

```
# cd /usr/local/bea/user_projects/domains/pr1/bin
# ./stopWebLogic.sh
# ./startWebLogic.sh
```

9 **Start the managed servers.**

```
# cd /usr/local/bea/user_projects/domains/pr1/bin
# ./startManagedWebLogic.sh ApplicationServer-1 t3://localhost:7001
```

You will be prompted for the administrative user credentials.

Username	**weblogic**
Password	**bea10admin**

10 **Change to the `ApplicationServer-1` directory.**

```
# cd /usr/local/bea/user_projects/domains/pr1/
  servers/ApplicationServer-1
```

11 **Create a security directory and change into it.**

```
# mkdir security
# cd security
```

12 **Create a `boot.properties` file for the WebLogic Server managed server administrator credentials.**

The managed server administrative user and password are stored in `boot.properties`. The Application Server 1 managed server uses this information during startup. WebLogic Server encrypts the file, so there is no security risk even if you enter the user name and password in clear text.

```
# cat > boot.properties
username=weblogic
password=bea10admin
```

Hit Control D to terminate the command

```
^D
```

13 **Restart the managed server.**

```
# cd /usr/local/bea/user_projects/domains/
  pr-1/bin
# ./stopManagedWebLogic.sh ApplicationServer-1
   t3://localhost:7001
# ./startManagedWebLogic.sh ApplicationServer-1
   t3://localhost:7001
```

14 **Run the `netstat` command to verify that the port is open and listening.**

```
# netstat -an | grep 1081

XXX.XX.XX.101.1081     *.*               0      0 49152      0 LISTEN
XXX.X.X.1.1081         *.*               0      0 49152      0 LISTEN
```

15 **Access `http://pr1.sp-example.com:7001/console` from a web browser.**

16 **Login to the BEA WebLogic Server as the administrator.**

Username	**`weblogic`**
Password	**`bea10admin`**

17 **Click `servers` under Domain Structure —>Environment.**

On the Summary of Servers page, verify that both *AdminServer (admin)* and *ApplicationServer-1* are running and OK.

18 **Log out of the console.**

19 **Log out of the `pr1.sp-example.com` host machine.**

▼ To Import a Certificate Authority Root Certificate to Protected Resource 1

The Certificate Authority (CA) root certificate enables the J2EE policy agent to trust the certificate from the OpenSSO Enterprise Load Balancer 2, and to establish trust with the certificate chain that is formed from the CA to the certificate.

Before You Begin Copy the same CA root certificate used in "To Install a CA Root Certificate to OpenSSO Enterprise Load Balancer 2" on page 186 to the `/export/software` directory on the `pr1.sp-example.com` host machine.

1 **As a root user, log into the `pr1.sp-example.com` host machine.**

2 **Change to the directory where `cacerts`, the certificate store is located.**

```
# cd /usr/local/bea/jdk150_06/jre/lib/security.
```

Tip – Backup `cacerts` before modifying it.

---

3 **Import `ca.cer`, the CA root certificate.**

```
# /usr/local/bea/jdk150_06/bin/keytool -import -trustcacerts
  -alias OpenSSLTestCA -file /export/software/ca.cer
  -keystore /usr/local/bea/jdk150_06/jre/lib/security/cacerts -storepass changeit

Owner: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=Sun,
O=Sun,L=Santa Clara, ST=California C=US
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=Sun,
O=Sun,L=Santa Clara, ST=California C=US
Serial number: 97dba0aa26db6386
Valid from: Tue Apr 18 07:66:19 PDT 2006 until: Tue Jan 13 06:55:19
PST 2009
Certificate fingerprints:
MD5: 9f:57:ED:B2:F2:88:B6:E8:0F:1E:08:72:CF:70:32:06
SHA1: 31:26:46:15:C5:12:5D:29:46:2A:60:A1:E5:9E:26:64:36:80:E4:70
Trust this certificate: [no] yes

Certificate was added to keystore.
```

4 **Verify that `ca.cer` was successfully imported.**

```
# /usr/local/bea/jdk150_06/bin/keytool -list
  -keystore /usr/local/bea/jdk150_06/jre/lib/security/cacerts
  -storepass changeit | grep -i openssl

OpenSSLTestCA, Sep 15, 2008, trustedCertEntry,
```

5 **Log out of the `pr1` host machine.**

▼ To Install the J2EE Policy Agent on Protected Resource 1

Before You Begin Set `JAVA_HOME` to `/usr/local/bea/jdk150_06`.

1 **As a root user, log into the `pr1.sp-example.com` host machine.**

2 **Stop the WebLogic Server 1 administration server and the WebLogic Server 1 managed instance.**

```
# cd /usr/local/bea/user_projects/domains/pr1/bin
# ./stopManagedWebLogic.sh ApplicationServer-1 t3://localhost:7001
# ./stopWebLogic.sh
```

3 **Create a directory into which you will download the J2EE Policy Agent bits and change into it.**

```
# mkdir /export/J2EEPA1
# cd /export/J2EEPA1
```

4 **Create a text file that contains a password for the Agent Profile created during installation.**

The J2EE Policy Agent installer requires this.

```
# cat > agent.pwd

j2eeagent1
```

Hit Control D to terminate the command

```
^D
```

5 **Download the J2EE policy agent bits for WebLogic Server from** `http://www.sun.com/download/index.jsp`**.**

```
# ls -al

total 18824
drwxr-xr-x   2 root     root         512 Jul 17 16:02 .
drwxr-xr-x   8 root     root         512 Jul 17 15:58 ..
-rw-r--r--   1 root     root          11 Jul 17 15:59 agent.pwd
-rw-r--r--   1 root     root           9 Jul 17 16:01 agentadm.pwd
-rw-r--r--   1 root     root     9623704 Jul 17 16:02 weblogic_v10_agent_3.zip
```

6 **Unzip the J2EE policy agent bits.**

```
# unzip weblogic_v10_agent_3.zip
```

7 Run the J2EE policy agent installer.

```
# cd /export/J2EEPA1/j2ee_agents/weblogic_v10_agent/bin
# chmod 755 agentadmin
# ./agentadmin --custom-install
```

8 When prompted, provide the following information.

The following information is to configure the J2EE Policy Agent against the OpenSSO Enterprise secure port.

`Please read the following License Agreement carefully:`	Press Enter to continue. Continue to press Enter until you reach the end of the License Agreement and the installer's Welcome page is displayed.
`Enter startup script location.`	Enter **`/usr/local/bea/user_projects/domains/pr1/bin/startwebLogic.sh`**
`Enter the WebLogic Server instance name: [AdminServer]`	Enter the name of the WebLogic Server instance secured by the agent **`ApplicationServer-1`**
`Enter the WebLogic home directory: [/usr/local/bea/wlserver_10.0]`	Enter **`/usr/local/bea/weblogic10`**.
`OpenSSO Enterprise URL`	Enter the URL where OpenSSO Enterprise is running (including the URI): **`https://lb4.sp-example.com:1081/opensso`**
`Is the agent being deployed on a Portal domain [false]`	Accept the default value.
`Agent URL:`	Enter the URL where the policy agent is running (including the URI): **`http://pr1.sp-example.com:1081/agentapp`**
`Enter the Encryption Key [+Yr3K4K1/lWFe4H17SBHMNIUzLNRut7H]:`	Accept the default value.
`Enter the Agent Profile Name:`	**`j2eeagent-1`**
`Enter the path to the password File:`	Enter **`/export/J2EEPA1/agent.pwd`**, path to the file that contains the password used for identifying the policy agent. **Note –** A warning message is displayed regarding the existence of the agent profile.

```
This Agent Profile does not exist in
OpenSSO Enterprise.
Will it be created by the installer? (Agent
Administrator name and password are required)
[true]:
```

Accept the default value to create the Agent Profile during installation.

```
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
Startup script location :
  /usr/local/bea/user_projects/domains/
  pr1/bin/startWebLogic.sh
WebLogic Server instance name :
  ApplicationServer-1
WebLogic home directory :
  /usr/local/bea/weblogic10
OpenSSO Server URL :
  https://lb4.sp-example.com:1081/opensso
Agent Installed on Portal domain : false
Agent URL :
  http://pr1.sp-example.com:1081/agentapp
Encryption Key :
  +Yr3K4K1/lWFe4H17SBHMNIUzLNRut7H
Agent Profile name : j2eeagent-1
Agent Profile Password file name :
  /export/J2EEPA1/agent.pwd

Verify your settings and decide from
the choices below:
1. Continue with Installation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]:
```

Accept the default value.

```
---------------------------------------------
SUMMARY OF AGENT INSTALLATION
-----------------------------
Agent instance name: Agent_001
Agent Bootstrap file location:
/export/J2EEPA1/j2ee_agents/
  weblogic_v10_agent/Agent_001/
  config/FAMAgentBootstrap.properties
Agent Configuration file location
/export/J2EEPA1/j2ee_agents/
  weblogic_v10_agent/Agent_001/
  config/FAMAgentConfiguration.properties
Agent Audit directory location:
/export/J2EEPA1/j2ee_agents/
  weblogic_v10_agent/Agent_001/logs/audit
Agent Debug directory location:
/export/J2EEPA1/j2ee_agents/
  weblogic_v10_agent/Agent_001/logs/debug

Install log file location:
/export/J2EEPA1/j2ee_agents/
  weblogic_v10_agent/installer-logs
  /audit/custom.log
```

Accept the default value.

When the installer is finished, a new file is in the bin directory called setAgentEnv_ApplicationServer-1.sh.

9 Modify the startup script `setDomainEnv.sh` to reference `setAgentEnv_ApplicationServer-1.sh` with the following sub procedure.

Tip – Backup `setDomainEnv.sh` before you modify it.

---

a. Change to the `bin` directory.

```
# cd /usr/local/bea/user_projects/domains/pr1/bin
```

b. Insert the following line at the end of `setDomainEnv.sh`.

```
. /usr/local/bea/user_projects/domains/pr1/
bin/setAgentEnv_ApplicationServer-1.sh
```

c. Save `setDomainEnv.sh` and close the file.

10 Change permissions for `setAgentEnv_ApplicationServer-1.sh`.

```
# chmod 755 setAgentEnv_ApplicationServer-1.sh
```

11 **Start the WebLogic Server administration server and managed instance.**

```
# ./startWebLogic.sh &
# ./startManagedWebLogic.sh ApplicationSever-1 t3://localhost:7001
```

Watch for startup errors.

12 **Verify that the J2EE Policy Agent 1 was successfully created in OpenSSO Enterprise using the following sub procedure.**

a. **Access** `https://lb4.sp-example.com:1081/opensso/console` **from a web browser.**

b. **Log in to the OpenSSO Enterprise console as the administrator.**

User Name: **amadmin**

Password: **ossoadmin**

c. **Under the Access Control tab, click** `/ (Top Level Realm)`**.**

d. **Click the Agents tab.**

e. **Click the J2EE tab.**

`j2eeagent-1` is displayed under the Agent table.

f. **Click** `j2eeagent-1`**.**

The `j2eeagent-1` properties page is displayed.

g. **Log out of the OpenSSO Enterprise console and close the browser.**

13 **Remove the password files.**

```
# cd /export/J2EEPA1
# rm agent.pwd
# rm agentadm.pwd
```

14 **Log out of the** `pr1.sp-example.com` **host machine.**

▼ To Deploy and Start the J2EE Policy Agent Housekeeping Application

The agent application is a housekeeping application bundled with the binaries and used by the agent for notifications and other internal functionality. This application must be deployed to the agent-protected web container using the same URI that was supplied during the agent installation process. For example, during the installation process, if you entered `/agentapp` as the deployment URI for the agent application, use that same context path in this procedure.

1 **Access `http://pr1.sp-example.com:7001/console` from a web browser.**

2 **Log in to the WebLogic Server console as the administrator.**

Username	**`weblogic`**
Password	**`bea10admin`**

3 **Under Domain Structure, click Deployments.**

4 **On the Summary of Deployments page, in the Change Center, click Lock & Edit.**

5 **Under Deployments, click Install.**

6 **On the Install Application Assistant page, click the `pr1.sp-example.com` link.**

7 **In the field named Location: `pr1.sp-example.com`, click the root directory.**

8 **Navigate to `/export/J2EEPA1/j2ee_agents/weblogic_v10_agent/etc`, the application directory.**

9 **Select `agentapp.war` and click Next.**

10 **In the Install Application Assistant page, choose *Install this deployment as an application* and click Next.**

11 **In the list of Servers, mark the checkbox for ApplicationServer-1 and click Next.**

12 **In the Optional Settings page, click Next.**

13 **Click Finish.**

14 **On the Settings for agentapp page, click Save.**

15 **In the Change Center, click Activate Changes.**

16 **On the Settings for agentapp page, click Deployments.**

17 **On the Summary of Deployments page, mark the `agentapp` checkbox and click Start > Servicing all requests.**

18 **On the Start Application Assistant page, click Yes.**

Tip – If you encounter a JavaScript^{TM} error, start the WebLogic Server instance and perform the steps again.

▼ To Deploy the J2EE Policy Agent Sample Application

1. **Access Application Server 1 at `http://pr1.sp-example.com:7001/console`.**

2. **Log in to the WebLogic Server console as the administrator.**

Username	**`weblogic`**
Password	**`bea10admin`**

3. **On the Change Center, click Lock & Edit.**

4. **Under Domain Structure, click Deployments.**

5. **Under Deployments, click Install.**

6. **On the Install Application Assistant page, click the `pr1.sp-example.com` link.**

7. **In the list for Location: `pr1.example.com`, click the root directory.**

8. **Navigate to the application directory (`/export/J2EEPA1/j2ee_agents/weblogic_v10_agent/sampleapp/dist`), select `agentsample.ear` and click Next.**

9. **In the Install Application Assistant page, choose *Install this deployment as an application* and click Next.**

10. **In the list of Servers, mark the checkbox for `ApplicationServer-1` and click Next.**

11. **On the Optional Settings page, click Next to accept the default settings.**

12. **On the Review Your Choices page, click Finish.**

    The Target Summary section indicates that the module `agentsample` will be installed on the target `ApplicationServer-1`.

13. **On the Settings for agentsample page, click Save.**

14. **On the Settings for agentsample page, click Activate Changes.**

15 **Under Domain Structure, click Deployments.**

16 **In the Deployments list, mark the checkbox for `agentsample` and click Start > Servicing All Requests.**

17 **On the Start Application Assistant page, click Yes.**

The state of the deployment changes from Prepared to Active.

18 **Log out of the Application Server 1 console.**

▼ To Configure the J2EE Policy Agent to Bypass Application Server Administrator Authentication

The J2EE policy agent can operate in *local* or *centralized* mode. The centralized option was selected during the custom installation of the agent. Centralized agent configuration stores agent configuration data in a data store managed by OpenSSO Enterprise. Since J2EE policy agents are configured in centralized mode, any configuration changes must be made using the OpenSSO Enterprise server. In this procedure, configure the agent to bypass authentication of the Application Server administrator.

1 **Access `https://lb4.sp-example.com:1081/opensso/console` from a web browser.**

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

User Name: **`amadmin`**

Password: **`ossoadmin`**

3 **Under the Access Control tab, click `/ (Top Level Realm)`.**

4 **Click the Agents tab.**

5 **Click the J2EE tab.**

`j2eeagent-1` is displayed under the Agent table.

6 **Click `j2eeagent-1`.**

The `j2eeagent-1` properties page is displayed.

7 **Click the Miscellaneous tab.**

The Miscellaneous properties page is displayed.

8 **Provide the user name of the Application Server administrator in the Bypass Principal List and click Add.**

Enter `weblogic` to ensure that the administrator will be authenticated against WebLogic itself and not OpenSSO Enterprise.

9 **Click Save.**

10 **Exit the console and close the browser.**

▼ To Enable the J2EE Policy Agent to Run in SSO Only Mode

1 **Access `https://lb4.sp-example.com:1081/opensso/console` from a web browser.**

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

User Name:	**`amadmin`**
Password:	**`ossoadmin`**

3 **Under the Access Control tab, click `/ (Top Level Realm)`.**

4 **Click the Agents tab.**

5 **Click the J2EE tab.**

`j2eeagent-1` is displayed under the Agent table.

6 **Click `j2eeagent-1`.**

The `j2eeagent-1` properties page is displayed.

7 **Click the General link on the `j2eeagent-1` properties page.**

8 **Remove the existing value of the Agent Filter Mode property.**

This value is displayed in the Current Values text box.

9 **Add the following values to the New Value text boxes and click Add.**

Map Key	**`agentsample`**
Corresponding Map Value	**`SSO_ONLY`**

10 **Click Save.**

11 **Log out of the OpenSSO Enterprise console and close the browser.**

12 **Log in to the** `pr1.sp-example.com` **host machine as root user.**

13 **Restart the WebLogic administration server and managed instance.**

```
# cd /usr/local/bea/user_projects/domains/pr1/bin
# ./stopManagedWebLogic.sh ApplicationServer-1 t3://localhost:7001
# ./stopWebLogic.sh
# ./startWebLogic.sh
# ./startManagedWebLogic.sh ApplicationServer-1 t3://localhost:7001
```

14 **Log out of the** `pr1.sp-example.com` **host machine.**

15 **Verify the configurations with the following sub procedure.**

a. **Close and reopen the browser application.**

b. **Access** `http://pr1.sp-example.com:1081/agentsample` **from a web browser.**

c. **Log in to the OpenSSO Enterprise console as the administrator.**

User Name:	**spuser**
Password:	**spuser**

The user is redirected to the service provider console for authentication.

d. **Close the browser.**

▼ To Configure the J2EE Policy Agent for SAML v2 Communication

1 **Access** `https://lb4.sp-example.com:1081/opensso/console` **from a web browser.**

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

User Name:	**amadmin**
Password:	**ossoadmin**

3 **Under the Access Control tab, click** `/ (Top Level Realm)`**.**

4 **Click the Agents tab.**

5 **Click the J2EE tab.**

`j2eeagent-1` is displayed under the Agent table.

6 **Click `j2eeagent-1`.**

The `j2eeagent-1` properties page is displayed.

7 **Click the OpenSSO Services tab.**

The Edit j2eeagent-1 page is displayed.

8 **Click the Login URL link on the Edit j2eeagent-1 page.**

9 **Remove the existing value of the OpenSSO Login URL property.**

This value is displayed in the Selected box.

10 **Enter `https://lb4.sp-example.com:1081/opensso/spssoinit?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1181/opensso` in the text box and click Add.**

This URL redirects the agent to the identity provider for authentication.

11 **Enter `https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSingleLogoutInit.jsp?idpEntityID=https://lb2.idp-example.com:1181/opensso` as a value of the OpenSSO Logout URL attribute and click Add.**

This URL redirects the agent to the identity provider for authentication.

12 **Click Save.**

13 **Click the Application tab.**

14 **Add the following values to the Application Logout URI text boxes and click Add.**

Map Key	**`agentsample`**
Corresponding Map Value	**`/agentsample/logout`**

15 **Click Save.**

16 **Log out of the OpenSSO Enterprise console and close the browser.**

17 **Log in to the `pr1.sp-example.com` host machine.**

18 **Restart the WebLogic administration server and managed instance.**

```
# cd /usr/local/bea/user_projects/domains/pr1/bin
# ./stopManagedWebLogic.sh ApplicationServer-1 t3://localhost:7001
# ./stopWebLogic.sh
# ./startWebLogic.sh
# ./startManagedWebLogic.sh ApplicationServer-1 t3://localhost:7001
```

19 **Log out of the** `pr1.sp-example.com` **host machine.**

20 **Verify the configurations with the following sub procedure.**

a. **Access** `http://pr1.sp-example.com:1081/agentsample` **from a web browser.**

The user is redirected to the OpenSSO Enterprise login page on the identity provider side.

b. **Log in to the OpenSSO Enterprise console as the administrator.**

User Name: **idpuser**

Password: **idpuser**

After successful authentication, single sign on is accomplished between the identity provider and the service provider.

21 **Access** `http://pr1.sp-example.com:1081/agentsample/logout` **from a web browser.**

The J2EE policy agent sample application welcome page is displayed. The user has successfully logged out of both the identity provider and the service provider.

10.2 Installing the Web Server and Web Policy Agent on Protected Resource 1

Download the Sun Java System Web Server bits to the Protected Resource 1 host machine (`pr1.sp-example.com`) and install it. Additionally, download, install and configure the appropriate web policy agent. Use the following list of procedures as a checklist for completing the task.

- "To Patch the Protected Resource 1 Host Machine" on page 234
- "To Install and Configure Sun Java System Web Server on Protected Resource 1" on page 235
- "To Import a Certificate Authority Root Certificate to Protected Resource 1" on page 239
- "To Install and Configure Web Policy Agent on Protected Resource 1" on page 240
- "To Enable the Web Policy Agent to Run in SSO Only Mode" on page 243
- "To Configure the Web Policy Agent for SAML v2 Communication" on page 244

▼ To Patch the Protected Resource 1 Host Machine

Sun Java System Web Server is the second web container used on the `pr1.sp-example.com` host machine.

Before You Begin Read the latest version of the Web Server 7.0 Release Notes to determine if you need to install patches on your host machine. In this case, the Release Notes indicate that based on the hardware and operating system being used, patch 119963–08, patch 120011–14, and patch 117461–08 are required.

1 **As a root user, log into the** `pr1.sp-example.com` **host machine.**

2 **Run** `patchadd` **to see if the patch is installed.**

```
# patchadd -p | grep 117461–08
```

A list of patch numbers is displayed. On our lab machine, the required patch 117461–08 is present so there is no need to install it.

```
# patchadd -p | grep 119963–08
```

No results are returned which indicates that the patch is not yet installed on the system.

```
# patchadd -p | grep 120011-14
```

No results are returned which indicates that the patch is not yet installed on the system.

3 **Make a directory for downloading the patch you need and change into it.**

```
# mkdir /export/patches
# cd /export/patches
```

4 **Download the patches.**

You can search for patches directly at `http://sunsolve.sun.com`. Navigate to the PatchFinder page, enter the patch number, click Find Patch, and download the appropriate patch.

Note – Signed patches are downloaded as JAR files. Unsigned patches are downloaded as ZIP files.

5 **Unzip the patch file.**

```
# unzip 119963–08.zip
# unzip 120011–14.zip
```

6 **Run** `patchadd` **to install the patches.**

```
# patchadd /export/patches/119963–08
# patchadd /export/patches/120011–14
```

7 **After installation is complete, run** `patchadd` **to verify that the patch was added successfully.**

```
# patchadd -p | grep 119963–08
```

In this example, a series of patch numbers are displayed, and the patch 119963–08 is present.

```
# patchadd -p | grep 120011-14
```

In this example, a series of patch numbers are displayed, and the patch 120011–14 is present.

▼ To Install and Configure Sun Java System Web Server on Protected Resource 1

Before You Begin This procedure assumes you have just finished "To Patch the Protected Resource 1 Host Machine" on page 234 and are still logged in as the root user.

1 **Create a directory into which you can download the Web Server bits and change into it.**

```
# mkdir /export/WS7
# cd /export/WS7
```

2 **Download the Sun Java System Web Server 7.0 Update 3 software from** `http://www.sun.com/download/products.xml?id=45ad781d`**.**

Follow the instructions on the Sun Microsystems Product Downloads web site for downloading the software.

3 **Unpack the Web Server package.**

```
# gunzip sjsws-7_0u3-solaris-sparc.tar.gz
# tar xvf sjsws-7_0u3-solaris-sparc.tar
```

4 **Run** `setup`**.**

```
# cd /export/WS7
# ./setup --console
```

5 **When prompted, provide the following information.**

Welcome to the Sun Java System Web Server 7.0u3 installation wizard. ... You will be asked to specify preferences that determine how Sun Java System Web Server 7.0U3 is installed and configured. The installation program pauses as questions are presented so you can read the information and make your choice. When you are ready to continue, press Enter. (Return on some keyboards.)	Press Enter. Continue to press Enter when prompted.
Have you read the Software License Agreement and do you accept all terms [no] {"," goes back, "!" exits}?	Enter **yes**.
Sun Java System Web Server 7.0 Installation Directory [/sun/webserver7] {"," goes back, "!" exits} :	Enter **/opt/SUNWwbsvr**
Specified directory /opt/SUNWwbsvr does not exist. Create Directory? [Yes/No] [yes] {"," goes back, "!" exits}	Enter **yes**.
Select Type of Installation 1. Express 2. Custom 3. Exit What would you like to do? [1] {"," goes back, "!" exits}	Enter **2**.
Component Selection 1. Server Core 2. Server Core 64-biy Binaries 3. Administration Command Line Interface 4. Sample Applications 5. Language Pack Enter the comma-separated list [1,2,3,4,5] {"," goes back, "!" exits}	Enter **1,3,5**.

`Java Configuration`
 	
`Sun Java System Web Server 7.0 requires Java`
`Se Development Kit (JDK). Provide the path`
`to a JDK 1.5.0_15 or greater.`
 	
`1. Install Java SE Development Kit (JDK) 1.5.0_15`
`2. Reuse existing Java SE Development Kit`
`   (JDK) 1.5.0_15`
`3. Exit`
 	
`What would you like to do? [1]`
`{"," goes back, "!" exits}`
	Enter **1**.

`Administrative Options`
 	
`1. Create an Administration Server and a`
`   Web Server Instance`
`2. Create an Administration Node`
 	
`Enter your option. [1]`
`{"," goes back, "!" exits}`
	Enter **1**.

`Create SMF services for server instances`
`[yes/no] [no] {"," goes back, "!" exits}`
	Accept the default value.

`Host Name [pr1.sp-example.com]`
`{"," goes back, "!" exits}`
	Accept the default value.

`SSL Port [8989]`
`{"," goes back, "!" exits}`
	Accept the default value.

`Create a non-SSL Port? [yes/no] [no]`
`{"," goes back, "!" exits}`
	Enter **no**.

`Runtime User ID [root]`
`{"," goes back, "!" exits}`
	Accept the default value (for the administration server).

`Administrator User Name [admin]`
`{"," goes back, "!" exits}`
	Accept the default value.

`Administrator Password:`	Enter **web4dmin**.
`Retype Password:`	Enter **web4dmin**.

`Server Name [pr1.sp-example.com]`
`{"," goes back, "!" exits}`
	Accept the default value.

`Http Port [8080]`
`{"," goes back, "!" exits}`
	Enter **1080**.

`Runtime User ID [webserverd]`
`{"," goes back, "!" exits}`
	Enter **root** (for the instance).

`Document Root Directory [/opt/SUNWwbsvr/`
`https-pr1.sp-example.com/docs]`
`{"," goes back, "!" exits}`
	Accept the default value.

`Start Administration Server [yes/no]`
`[yes] {"," goes back, "!" exits}`
	Enter **no**.

`Ready To Install`
 	
`1. Install Now`
`2. Start Over`
`3. Exit Installation`
 	
`What would you like to do [1]`
`{"," goes back, "!" exits}?`
	Enter**1**.

When installation is complete, the following message is displayed:

```
Installation Successful.
```

6. **Start the Web Server administration server.**

```
# cd /opt/SUNWwbsvr/admin-server/bin
# ./startserv
```

7. **Run `netstat` to verify that the port is open and listening.**

```
# netstat -an | grep 8989

*.8989               *.*                0      0 49152      0 LISTEN
```

8. **(Optional) Login to the Web Server administration console at `https://pr1.sp-example.com:8989` as the administrator.**

Username	**admin**
Password	**web4dmin**

You should see the Web Server administration console.

9. **(Optional) Log out of the Web Server console and close the browser.**

10. **Start the Protected Resource 1 Web Server instance.**

```
# cd /opt/SUNWwbsvr/https-pr1.sp-example.com/bin
# ./startserv

Sun Java System Web Server 7.0U3 B06/16/2008 12:00
info: CORE5076: Using [Java HotSpot(TM) Server VM, Version 1.5.0_15] from
[Sun Microsystems Inc.]
info: HTTP3072: http-listener-1: http://pr1.sp-example.com:1080 ready to
```

```
accept requests
info: CORE3274: successful server startup
```

11 **Run `netstat` to verify that the port is open and listening.**

```
# netstat -an | grep 1080

*.1080               *.*                0      0 49152      0 LISTEN
```

12 **(Optional) Access the Protected Resource 1 instance at `http://pr1.sp-example.com:1080` using a web browser.**

You should see the default Web Server index page.

13 **Log out of the `pr1.sp-example.com` host machine.**

To Import a Certificate Authority Root Certificate to Protected Resource 1

The Certificate Authority (CA) root certificate enables the web policy agent to trust the certificate from the OpenSSO Enterprise Load Balancer 2, and to trust the certificate chain that is formed from the CA to the server certificate.

Before You Begin

- Copy the same CA root certificate used in "To Install a CA Root Certificate to OpenSSO Enterprise Load Balancer 2" on page 186 to the `pr1.sp-example.com` host machine. In this example, the file is `/export/software/ca.cer`.
- Backup `cacerts` before modifying it.

1 **As a root user, log into the `pr1.sp-example.com` host machine.**

2 **Import the CA root certificate into `cacerts`, the certificate store.**

```
# /opt/SUNWwbsvr/jdk/jre/bin/keytool -import -trustcacerts
-alias OpenSSLTestCA -file /export/software/ca.cer
-keystore /opt/SUNWwbsvr/jdk/jre/lib/security/cacerts -storepass changeit

Owner: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=Sun,
O=Sun,L=Santa Clara, ST=California C=US
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=OpenSSLTestCA, OU=Sun,
O=Sun,L=Santa Clara, ST=California C=US
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:14:51 PDT 2008 18 07:66:19 PDT 2006
until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
MD5: 78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA
```

```
Trust this certificate: [no] yes

Certificate was added to keystore.
```

3 **Verify that the CA root certificate was imported.**

```
# /opt/SUNWwbsvr/jdk/jre/bin/keytool -list
-keystore /opt/SUNWwbsvr/jdk/jre/lib/security/cacerts
-storepass changeit | grep -i open

openSSLTestCA, Sep 20, 2008, trustedCertEntry,
```

4 **Log out of the `pr1.sp-example.com` host machine.**

▼ To Install and Configure Web Policy Agent on Protected Resource 1

Before You Begin The `JAVA_HOME` environment variable should be set to `/opt/SUNWwbsvr/jdk/jre`.

1 **As a root user, log into the `pr1.sp-example.com` host machine.**

2 **Create a directory into which you can download the Web Server agent bits and change into it.**

```
# mkdir /export/WebPA1
# cd /export/WebPA1
```

3 **Create a text file that contains the Agent Profile password.**

The Web Policy Agent installer requires this for installation.

```
# cat > agent.pwd

webagent1
```

Hit Control D to terminate the command

```
^D
```

4 **Download the web policy agent for Web Server from http://www.sun.com/download/.**

```
# ls -al

total 7512
drwxr-xr-x   2 root     root         512 Jul 24 14:48 .
drwxr-xr-x  11 root     root         512 Jul 24 14:41 ..
-rw-r--r--   1 root     root          10 Jul 24 14:42 agent.pwd
-rw-r--r--   1 root     root           9 Jul 24 14:42 agentadm.pwd
```

```
-rw-r--r--   1 root     root     3826794 Jul 24 14:48 sjsws_v70_SunOS_sparc_agent_3.zip
```

5 Unzip the downloaded file.

```
# unzip sjsws_v70_SunOS_sparc_agent_3.zip
```

6 Run the agent installer.

```
# cd /export/WebPA1/web_agents/sjsws_agent/bin
# ./agentadmin --custom-install
```

7 When prompted, do the following.

`Please read the following License Agreement carefully:`	Press Enter and continue to press Enter until you have reached the end of the License Agreement.
`Do you completely agree with all the terms and conditions of this License Agreement (yes/no): [no]:`	Type **`yes`** and press Enter.
`Enter the Sun Java System Web Server Config Directory Path [/var/opt/SUNWwbsvr7/ https-pr1.sp-example.com/config]:`	Type **`/opt/SUNWwbsvr/https-pr1.sp-example.com/config`** and press Enter.
`Enter the OpenSSO Enterprise URL including the deployment URI (http://opensso.sample.com:58080/opensso)`	Type **`https://lb4.example.com:1081/opensso`** and press Enter.
`Enter the Agent URL: (http://agent1.sample.com:1234)`	Type **`http://pr1.sp-example.com:1080`** and press Enter.
`Enter the Encryption Key [WSpf7aqc3AFIGvf2mCqvNBOsf44cDrf3].`	Accept the default value.
`Enter the Agent profile name [UrlAccessAgent]:`	Type **`webagent-1`** and press Enter.
`Enter the path to a file that contains the password to be used for identifying the Agent.`	Type **`/export/WebPA1/agent.pwd`** and press Enter. **Note –** A warning message is displayed regarding the existence of the agent profile.

```
-----------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------

Sun Java System Web Server Config Directory :
 /opt/SUNWwbsvr/https-pr1.sp-example.com/config
OpenSSO Server URL :
 https://lb4.sp-example.com:1081/opensso
Agent URL : http://pr1.sp-example.com:1080
Encryption Key :
 WSpf7aqc3AFIGvf2mCqvNBOsf44cDrf3
Agent Profile name : webagent-1
Agent Profile Password file name :
 /export/WebPA1/agent.pwd
Agent Profile will be created right now by
 agent installer : true
Agent Administrator : amadmin
Agent Administrator’s password file name :
 /export/WebPA1/agentadm.pwd

Verify your settings above and decide from
the choices below.

  1. Continue with Installation
  2. Back to the last interaction
  3. Start Over
  4. Exit

Please make your selection [1]:
```

Type **1** and press Enter.

8 **Restart the Web Server 1 instance.**

```
# cd /opt/SUNWwbsvr/https-pr1.sp-example.com/bin
# ./stopserv; ./startserv

server has been shutdown
Sun Java System Web Server 7.0U3 B06/16/2008 12:00
info: CORE3016: daemon is running as super-user
info: CORE5076: Using [Java HotSpot(TM) Server VM, Version 1.5.0_15]
from [Sun Microsystems Inc.]
info: HTTP3072: http-listener-1: http://pr1.sp-example.com:1080 ready to
accept requests
info: CORE3274: successful server startup
```

9 **Verify that the Web Policy Agent was successfully created in OpenSSO Enterprise using the following sub procedure.**

a. **Access** `https://lb4.sp-example.com:1081/opensso/console` **from a web browser.**

b. **Log in to the OpenSSO Enterprise console as the administrator.**

User Name: **amadmin**

Password: **ossoadmin**

c. **Under the Access Control tab, click `/ (Top Level Realm)`.**

d. **Click the Agents tab.**

By default, the Web tab is displayed. You should see `webagent-1` under the Agent table.

e. **Click `webagent-1`.**

The `webagent-1` properties page is displayed.

f. **Log out of the console and close the browser.**

10 **Remove the password files.**

```
# cd /export/WebPA1
# rm agent.pwd
# rm agentadm.pwd
```

11 **Log out of the `pr1.sp-example.com` host machine.**

▼ To Enable the Web Policy Agent to Run in SSO Only Mode

1 **Access `https://lb4.sp-example.com:1081/opensso/console` from a web browser.**

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

User Name: **amadmin**

Password: **ossoadmin**

3 **Under the Access Control tab, click `/ (Top Level Realm)`.**

4 **Click the Agents tab.**

5 **Click the Web tab.**

`webagent-1` is displayed under the Agent table.

6 **Click `webagent-1`.**

The `webagent-1` properties page is displayed.

7 **Click the General link on the `webagent-1` properties page.**

8 **Select the check box to enable the SSO Mode Only property.**

9 **Click Save.**

10 **Log out of the OpenSSO Enterprise console and close the browser.**

11 **Log in to the `pr1.sp-example.com` host machine as root user.**

12 **Restart the Web Server.**

```
# cd /opt/SUNWwbsvr/https-pr1.sp-example.com/bin
# ./stopserv
# ./startserv
```

13 **Log out of the `pr1.sp-example.com` host machine.**

14 **Verify the configurations with the following sub procedure.**

a. **Close and reopen the browser.**

b. **Access `https://lb4.sp-example.com:1080/index.html` from a web browser.**

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

User Name: **`spuser`**

Password: **`spuser`**

The default Web Server page is displayed.

d. **Close the browser.**

▼ To Configure the Web Policy Agent for SAML v2 Communication

1 **Access `https://lb4.sp-example.com:1081/opensso/console` from a web browser.**

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

User Name: **`amadmin`**

Password: **`ossoadmin`**

3 **Under the Access Control tab, click `/ (Top Level Realm)`.**

4 **Click the Agents tab.**

5 **Click the Web tab.**

`webagent-1` is displayed under the Agent table.

6 **Click `webagent-1`.**

The `webagent-1` properties page is displayed.

7 **Click the OpenSSO Services tab.**

The Edit webagent-1 page is displayed.

8 **Click the Login URL link on the Edit webagent-1 page.**

9 **Remove the existing value of the OpenSSO Login URL property.**

This value is displayed in the Selected box.

10 **Enter `https://lb4.sp-example.com:1081/opensso/spssoinit?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1181/opensso` in the text box and click Add.**

This URL redirects the agent to the identity provider for authentication.

11 **Select the existing value of the OpenSSO Logout URL attribute and click Delete.**

12 **Enter `https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSingleLogoutInit.jsp?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1181/opensso` in the text box and click Add.**

13 **Enter `http://www.sun.com` as a value of the Logout Redirect URL attribute and click Add.**

14 **Enter `http://pr1.sp-example.com:1080/logout.html` as a value of the Agent Logout URL List attribute and click Add.**

15 **Click Save.**

16 **Log out of the OpenSSO Enterprise console and close the browser.**

17 **Log in to the `pr1.sp-example.com` host machine.**

18 **Create the `logout.html` file using the following sub procedure.**

```
# cd /opt/SUNWwbsvr/https-pr1.sp-example.com/docs
# vi logout.html
```

This creates an empty file.

19 **Restart the Web Server.**

```
# cd /opt/SUNWwbsvr/https-pr1.sp-example.com/bin
# ./stopserv
# ./startserv
```

20 **Log out of the `pr1.sp-example.com` host machine.**

21 **Verify the configurations with the following sub procedure.**

a. **Access `http://pr1.sp-example.com:1080/index.html` from a web browser.**

The OpenSSO Enterprise login page on the identity provider side is displayed. The browser is then redirected to the identity provider for authentication.

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

User Name:	**`idpuser`**
Password:	**`idpuser`**

The default Web Server page is displayed.

c. **Access `http://pr1.sp-example.com:1080/logout.html` from a web browser.**

This will log out the user from the service provider and the identity provider using the SAML v2 single logout protocol.

d. **Close the browser.**

PART IV

Configuring and Testing the SAML v2 Communications

This fourth part of *Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0* contains the procedures used to configure for SAML v2 communications and test the configurations between the environment we prepared in Part III and Part II. It contains the following chapters.

CHAPTER 11

Configuring OpenSSO Enterprise for SAML v2

This deployment consists of a service provider and an identity provider that communicate for purposes of federation using SAML v2. Towards this end, we configure each instance of OpenSSO Enterprise (respectively) acting as the identity provider and the service provider as *hosted*. Additionally, we configure each hosted instance with the necessary information to communicate with the *remote* provider — in essence, with each other. In this chapter, we configure the instances of OpenSSO Enterprise as SAML v2 providers.

- "11.1 Configuring OpenSSO Enterprise as the Hosted Identity Provider" on page 249
- "11.2 Configuring OpenSSO Enterprise as the Hosted Service Provider" on page 255
- "11.3 Configuring the Hosted Service Provider to Communicate with the Remote Identity Provider" on page 265

11.1 Configuring OpenSSO Enterprise as the Hosted Identity Provider

This section provides the procedures for configuring OpenSSO Enterprise on the identity provider side as a hosted identity provider using the Common Tasks wizard. Use the following list of procedures as a checklist for completing the task.

1. "To Configure the Hosted Identity Provider" on page 249
2. "To View the Hosted Identity Provider Metadata in XML Format" on page 250

▼ To Configure the Hosted Identity Provider

Configure the instance of OpenSSO Enterprise deployed in Part II and situated behind Load Balancer 2, as a hosted identity provider. This procedure creates the `idpcot` circle of trust.

1 **Access** `https://lb2.idp-example.com:1081/opensso/console` **from a web browser.**

2 Log in to the OpenSSO Enterprise console as the administrator.

Username	`amadmin`
Password	`ossoadmin`

The Common Tasks tab is displayed.

3 Click Create Hosted Identity Provider under Create SAML v2 Providers.

The Create a SAML v2 Identity Provider on this Server page is displayed.

4 Make the following changes on the Create a SAML v2 Identity Provider on this Server page.

- Select the No radio button for *Do you have metadata for this provider?*
- Under metadata properties, type **`https://lb2.idp-example.com:1081/opensso`** as the value for Name.
- Under metadata properties, select **`test`** as the value for Signing Key.
- Under Circle of Trust properties, type **`idpcot`** as the value for the New Circle of Trust.
- Accept the default values for any remaining properties.

5 Click Configure.

6 Select Finish to end the task.

This instance of OpenSSO Enterprise is now configured as a SAML v2 identity provider.

7 Click the Federation tab to verify the hosted identity provider configurations.

- Confirm that `idpcot` was created under the Circle of Trust table with one entity: `https://lb2.idp-example.com:1081/opensso|saml2`.
- Confirm that `https://lb2.idp-example.com:1081/opensso|saml2` was created under the Entity Providers table.

▼ To View the Hosted Identity Provider Metadata in XML Format

This optional procedure displays, in a browser window, the standard and extended metadata for the hosted identity provider in XML format. The XML can be viewed as displayed or copied into a text file and saved.

Before You Begin This procedure assumes that you have just completed "To Configure the Hosted Identity Provider" on page 249 and are still logged in to the OpenSSO Enterprise console.

1 **Access `https://lb2.idp-example.com:1081/opensso/ssoadm.jsp` from the web browser.**

`ssoadm.jsp` is a Java Server Page (JSP) version of the `ssoadm` command line interface. In this procedure it is used to display the hosted identity provider metadata.

2 **Click `export-entity`.**

The `export-entity` page is displayed.

3 **Enter the following values for each option and click Submit.**

entityid
The `EntityID` is the unique uniform resource identifier (URI) used to identify a particular provider. In this deployment, type **`https://lb2.idp-example.com:1081/opensso`**.

realm
The OpenSSO Enterprise realm in which the data resides. In this deployment as all data resides in the top-level realm, type **`/`**.

sign
Leave this unchecked.

meta-data-file
Set this flag to export the standard metadata for the provider.

extended-data-file
Set this flag to export the extended metadata for the provider.

spec
Type **`saml2`**.

4 **View the XML-formatted metadata in the browser window.**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EntityDescriptor entityID="https://lb2.idp-example.com:1081/opensso"
 xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
  <IDPSSODescriptor WantAuthnRequestsSigned="false"
   protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  <KeyDescriptor use="signing">
   <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:X509Data>
    <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
    </ds:X509Certificate>
    </ds:X509Data>
    </ds:KeyInfo>
  </KeyDescriptor>
  <ArtifactResolutionService index="0" isDefault="true" Binding=
   "urn:oasis:names:tc:SAML:2.0:bindings:SOAP" Location=
   "https://lb2.idp-example.com:1081/opensso/ArtifactResolver/metaAlias/idp"/>
  <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:
   HTTP-Redirect" Location="https://lb2.idp-example.com:1081/opensso/
   IDPSloRedirect/metaAlias/idp" ResponseLocation="
   https://lb2.idp-example.com:1081/opensso/IDPSloRedirect/metaAlias/idp"/>
  <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:
   HTTP-POST" Location="https://lb2.idp-example.com:1081/opensso/IDPSloPOST/
   metaAlias/idp" ResponseLocation="https://lb2.idp-example.com:1081/opensso/
   IDPSloPOST/metaAlias/idp"/>
  <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
   Location="https://lb2.idp-example.com:1081/opensso/IDPSloSoap/metaAlias/idp"/>
  <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:
   HTTP-Redirect" Location="https://lb2.idp-example.com:1081/opensso/
   IDPMniRedirect/metaAlias/idp" ResponseLocation=
   "https://lb2.idp-example.com:1081/opensso/IDPMniRedirect/metaAlias/idp"/>
  <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
   Location="https://lb2.idp-example.com:1081/opensso/IDPMniPOST/metaAlias/idp"
   ResponseLocation="https://lb2.idp-example.com:1081/opensso/
   IDPMniPOST/metaAlias/idp"/>
  <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
   Location="https://lb2.idp-example.com:1081/opensso/IDPMniSoap/metaAlias/idp"/>
  <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</NameIDFormat>
  <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
  <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</NameIDFormat>
  <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</NameIDFormat>
  <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
   Location="https://lb2.idp-example.com:1081/opensso/SSORedirect/metaAlias/idp"/>
  <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
   Location="https://lb2.idp-example.com:1081/opensso/SSOPOST/metaAlias/idp"/>
  <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
   Location="https://lb2.idp-example.com:1081/opensso/SSOSoap/metaAlias/idp"/>
  <NameIDMappingService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
   Location="https://lb2.idp-example.com:1081/opensso/NIMSoap/metaAlias/idp"/>
   <AssertionIDRequestService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
    Location="https://lb2.idp-example.com:1081/opensso/AIDReqSoap/
    IDPRole/metaAlias/idp"/>
   <AssertionIDRequestService Binding="urn:oasis:names:tc:SAML:2.0:bindings:URI"
    Location="https://lb2.idp-example.com:1081/opensso/AIDReqUri/
    IDPRole/metaAlias/idp"/>
  </IDPSSODescriptor>
</EntityDescriptor>

Entity descriptor was exported to file, web.
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EntityConfig entityID="https://lb2.idp-example.com:1081/opensso" hosted="true"
 xmlns="urn:sun:fm:SAML:2.0:entityconfig">
    <IDPSSOConfig metaAlias="/idp">
      <Attribute name="wantNameIDEncrypted">
          <Value/>
      </Attribute>
      <Attribute name="AuthUrl">
          <Value/>
      </Attribute>
      <Attribute name="nameIDFormatMap">
        <Value>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=</Value>
        <Value>urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos=</Value>
        <Value>urn:oasis:names:tc:SAML:1.1:nameid-format:
         WindowsDomainQualifiedName=</Value>
         <Value>urn:oasis:names:tc:SAML:1.1:nameid-format:
          X509SubjectName=</Value>
         <Value>urn:oasis:names:tc:SAML:1.1:nameid-format:
          emailAddress=mail</Value>
       </Attribute>
       <Attribute name="cotlist">
         <Value>idpcot</Value>
       </Attribute>
       <Attribute name="saeIDPUrl">
         <Value>https://lb2.idp-example.com:1081/opensso/idpsaehandler/
          metaAlias/idp</Value>
       </Attribute>
       <Attribute name="idpAuthncontextClassrefMapping">
         <Value>urn:oasis:names:tc:SAML:2.0:ac:classes:
          PasswordProtectedTransport|0||default</Value>
       </Attribute>
       <Attribute name="appLogoutUrl">
         <Value/>
       </Attribute>
       <Attribute name="idpAccountMapper">
         <Value>com.sun.identity.saml2.plugins.
          DefaultIDPAccountMapper</Value>
       </Attribute>
       <Attribute name="autofedEnabled">
         <Value>false</Value>
       </Attribute>
        <Attribute name="signingCertAlias">
            <Value>test</Value>
        </Attribute>
        <Attribute name="assertionCacheEnabled">
            <Value>false</Value>
        </Attribute>
```

```
<Attribute name="idpAuthncontextMapper">
    <Value>com.sun.identity.saml2.plugins.
     DefaultIDPAuthnContextMapper</Value>
</Attribute>
<Attribute name="assertionEffectiveTime">
    <Value>600</Value>
</Attribute>
<Attribute name="wantMNIResponseSigned">
    <Value/>
</Attribute>
<Attribute name="wantMNIRequestSigned">
    <Value/>
</Attribute>
<Attribute name="attributeMap">
    <Value>EmailAddress=mail</Value>
    <Value>Telephone=telephonenumber</Value>
</Attribute>
<Attribute name="discoveryBootstrappingEnabled">
    <Value>false</Value>
</Attribute>
<Attribute name="basicAuthUser">
    <Value/>
</Attribute>
<Attribute name="idpAttributeMapper">
    <Value>com.sun.identity.saml2.plugins.
     DefaultIDPAttributeMapper</Value>
</Attribute>
<Attribute name="idpECPSessionMapper">
    <Value>com.sun.identity.saml2.plugins.
     DefaultIDPECPSessionMapper</Value>
</Attribute>
<Attribute name="basicAuthPassword">
    <Value/>
</Attribute>
<Attribute name="basicAuthOn">
    <Value>false</Value>
</Attribute>
<Attribute name="wantLogoutResponseSigned">
    <Value/>
</Attribute>
<Attribute name="wantLogoutRequestSigned">
    <Value/>
</Attribute>
<Attribute name="encryptionCertAlias">
    <Value/>
</Attribute>
<Attribute name="wantArtifactResolveSigned">
```

```
                <Value/>
            </Attribute>
            <Attribute name="assertionNotBeforeTimeSkew">
                <Value>600</Value>
            </Attribute>
            <Attribute name="autofedAttribute">
                <Value/>
            </Attribute>
            <Attribute name="saeAppSecretList"/>
        </IDPSSOConfig>
</EntityConfig>

Entity configuration was exported to file, web.
```

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

11.2 Configuring OpenSSO Enterprise as the Hosted Service Provider

This section provides the procedures for configuring OpenSSO Enterprise on the service provider side as a hosted service provider using the Common Tasks wizard. Use the following list of procedures as a checklist for completing the task.

1. "To Configure the Hosted Service Provider" on page 255
2. "To View the Hosted Service Provider Metadata in XML Format" on page 256

▼ To Configure the Hosted Service Provider

Configure the instance of OpenSSO Enterprise deployed in Part III, situated behind Load Balancer 2 on the service provider side, as a hosted service provider. This procedure creates the `spcot` circle of trust.

1 **Access** `https://lb4.sp-example.com:1081/opensso/console` **from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

The Common Tasks tab is displayed.

3 **Click Create Hosted Service Provider under Create SAML v2 Providers.**

The Create a SAML v2 Service Provider on this Server page is displayed.

4 **Make the following changes on the Create a SAML v2 Service Provider on this Server page.**

- Select the No radio button for *Do you have metadata for this provider?*
- Under metadata properties, type **`https://lb4.sp-example.com:1081/opensso`** as the value for Name.
- Under metadata properties, select **`test`** as the value for Signing Key.
- Under Circle of Trust properties, select the Add to New radio button and type **`spcot`** as the value for the New Circle of Trust.
- Accept the default values for any remaining properties.

5 **Click Configure.**

A pop up screen is displayed that reads:

```
Service provider is configured.
You can modify the provider’s profile under the Federation tab.

Do you want to create a remote identity provider?
```

6 **Click No on the pop up screen.**

The OpenSSO Enterprise console is displayed and this instance is now configured as a SAML v2 service provider.

▼ To View the Hosted Service Provider Metadata in XML Format

This optional procedure displays, in a browser window, the standard and extended metadata for the hosted service provider in XML format. The XML can be viewed as displayed or copied into a text file and saved.

Before You Begin This procedure assumes that you have just completed “To Configure the Hosted Service Provider” on page 255 and are still logged in to the OpenSSO Enterprise console.

1 **Access `https://lb4.sp-example.com:1081/opensso/ssoadm.jsp` from the web browser.**

`ssoadm.jsp` is a Java Server Page (JSP) version of the `ssoadm` command line interface. In this procedure it is used to display the hosted service provider metadata.

2 **Click** `export-entity`**.**

The `export-entity` page is displayed.

3 **Enter the following values for each option and click Submit.**

entityid
: The `EntityID` is the unique uniform resource identifier (URI) used to identify a particular provider. In this deployment, type **`https://lb4.sp-example.com:1081/opensso`**.

realm
: The OpenSSO Enterprise realm in which the data resides. In this deployment as all data resides in the top-level realm, type **`/`**.

sign
: Leave this box unchecked.

meta-data-file
: Set this flag to export the standard metadata for the provider.

extended-data-file
: Set this flag to export the extended metadata for the provider.

spec
: Type **`saml2`**.

4 **View the XML-formatted metadata in the browser window.**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EntityDescriptor entityID="https://lb4.sp-example.com:1081/opensso"
  xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
  <SPSSODescriptor AuthnRequestsSigned="false" WantAssertionsSigned=
   "false" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
   <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
    Location="https://lb4.sp-example.com:1081/opensso/SPSloRedirect/metaAlias/sp"
    ResponseLocation="https://lb4.sp-example.com:1081/opensso/
    SPSloRedirect/metaAlias/sp"/>
   <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    Location="https://lb4.sp-example.com:1081/opensso/SPSloPOST/metaAlias/sp"
    ResponseLocation="https://lb4.sp-example.com:1081/opensso/SPSloPOST/metaAlias/sp"/>
   <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
    Location="https://lb4.sp-example.com:1081/opensso/SPSloSoap/metaAlias/sp"/>
   <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:
    HTTP-Redirect" Location="https://lb4.sp-example.com:1081/opensso/SPMniRedirect/
    metaAlias/sp" ResponseLocation="https://lb4.sp-example.com:1081/opensso/
    SPMniRedirect/metaAlias/sp"/>
   <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:
    HTTP-POST" Location="https://lb4.sp-example.com:1081/opensso/SPMniPOST/
    metaAlias/sp" ResponseLocation="https://lb4.sp-example.com:1081/opensso/
    SPMniPOST/metaAlias/sp"/>
   <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
    Location="https://lb4.sp-example.com:1081/opensso/SPMniSoap/metaAlias/sp"
    ResponseLocation="https://lb4.sp-example.com:1081/opensso/SPMniSoap/metaAlias/sp"/>
   <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</NameIDFormat>
   <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
   <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</NameIDFormat>
   <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</NameIDFormat>
   <AssertionConsumerService index="0" isDefault="true" Binding="urn:oasis:names:tc:
    SAML:2.0:bindings:HTTP-Artifact" Location="https://lb4.sp-example.com:1081/opensso/
    Consumer/metaAlias/sp"/>
   <AssertionConsumerService index="1" Binding="urn:oasis:names:tc:SAML:2.0:bindings:
    HTTP-POST" Location="https://lb4.sp-example.com:1081/opensso/
    Consumer/metaAlias/sp"/>
   <AssertionConsumerService index="2" Binding="urn:oasis:names:tc:SAML:2.0:
    bindings:PAOS" Location="https://lb4.sp-example.com:1081/opensso/Consumer/
    ECP/metaAlias/sp"/>
   </SPSSODescriptor>
   <IDPSSODescriptor WantAuthnRequestsSigned="false" protocolSupportEnumeration=
    "urn:oasis:names:tc:SAML:2.0:protocol">
        <KeyDescriptor use="signing">
            <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:X509Data>
                    <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
                    </ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </KeyDescriptor>
    <ArtifactResolutionService index="0" isDefault="true" Binding="urn:oasis:
     names:tc:SAML:2.0:bindings:SOAP" Location="https://lb4.sp-example.com:1081/
     opensso/ArtifactResolver/metaAlias/idp"/>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
     Location="https://lb4.sp-example.com:1081/opensso/IDPSloRedirect/metaAlias/idp"
     ResponseLocation="https://lb4.sp-example.com:1081/opensso/IDPSloRedirect/
     metaAlias/idp"/>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
     Location="https://lb4.sp-example.com:1081/opensso/IDPSloPOST/metaAlias/idp"
     ResponseLocation="https://lb4.sp-example.com:1081/opensso/IDPSloPOST/
     metaAlias/idp"/>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
     Location="https://lb4.sp-example.com:1081/opensso/IDPSloSoap/metaAlias/idp"/>
    <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
     Location="https://lb4.sp-example.com:1081/opensso/IDPMniRedirect/metaAlias/idp"
     ResponseLocation="https://lb4.sp-example.com:1081/opensso/IDPMniRedirect/
     metaAlias/idp"/>
    <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
     Location="https://lb4.sp-example.com:1081/opensso/IDPMniPOST/metaAlias/idp"
     ResponseLocation="https://lb4.sp-example.com:1081/opensso/IDPMniPOST/
     metaAlias/idp"/>
    <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
     Location="https://lb4.sp-example.com:1081/opensso/IDPMniSoap/metaAlias/idp"/>
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:
     persistent</NameIDFormat>
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:
     transient</NameIDFormat>
    <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:
     emailAddress</NameIDFormat>
    <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:
     unspecified</NameIDFormat>
    <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:
     HTTP-Redirect" Location="https://lb4.sp-example.com:1081/opensso/
     SSORedirect/metaAlias/idp"/>
    <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
     Location="https://lb4.sp-example.com:1081/opensso/SSOPOST/metaAlias/idp"/>
    <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
     Location="https://lb4.sp-example.com:1081/opensso/SSOSoap/metaAlias/idp"/>
     <NameIDMappingService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://lb4.sp-example.com:1081/opensso/NIMSoap/metaAlias/idp"/>
      <AssertionIDRequestService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
       Location="https://lb4.sp-example.com:1081/opensso/AIDReqSoap/IDPRole/
       metaAlias/idp"/>
      <AssertionIDRequestService Binding="urn:oasis:names:tc:SAML:2.0:bindings:URI"
       Location="https://lb4.sp-example.com:1081/opensso/AIDReqUri/IDPRole/
       metaAlias/idp"/>
   </IDPSSODescriptor>
</EntityDescriptor>

Entity descriptor was exported to file, web.

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EntityConfig entityID="https://lb4.sp-example.com:1081/opensso" hosted="true"
 xmlns="urn:sun:fm:SAML:2.0:entityconfig">
    <SPSSOConfig metaAlias="/sp">
        <Attribute name="wantNameIDEncrypted">
            <Value/>
        </Attribute>
        <Attribute name="idpProxyList"/>
        <Attribute name="spAccountMapper">
            <Value>com.sun.identity.saml2.plugins.DefaultSPAccountMapper</Value>
        </Attribute>
        <Attribute name="enableIDPProxy">
            <Value>false</Value>
        </Attribute>
        <Attribute name="ECPRequestIDPListGetComplete">
            <Value/>
        </Attribute>
        <Attribute name="cotlist">
            <Value>spcot</Value>
        </Attribute>
        <Attribute name="transientUser">
            <Value>anonymous</Value>
        </Attribute>
        <Attribute name="spAuthncontextComparisonType">
            <Value>exact</Value>
        </Attribute>
        <Attribute name="wantAssertionEncrypted">
            <Value/>
        </Attribute>
        <Attribute name="spAdapter">
            <Value/>
        </Attribute>
        <Attribute name="spAuthncontextClassrefMapping">
            <Value>urn:oasis:names:tc:SAML:2.0:ac:classes:
             PasswordProtectedTransport|0|default</Value>
        </Attribute>
        <Attribute name="appLogoutUrl">
            <Value/>
        </Attribute>
        <Attribute name="saml2AuthModuleName">
            <Value/>
        </Attribute>
        <Attribute name="autofedEnabled">
            <Value>true</Value>
        </Attribute>
        <Attribute name="localAuthURL">
            <Value/>
        </Attribute>
        <Attribute name="spAttributeMapper">
            <Value>com.sun.identity.saml2.plugins.DefaultSPAttributeMapper</Value>
        </Attribute>
        <Attribute name="signingCertAlias">
            <Value/>
        </Attribute>
        <Attribute name="wantMNIResponseSigned">
            <Value/>
        </Attribute>
        <Attribute name="wantMNIRequestSigned">
```

```
            <Value/>
        </Attribute>
        <Attribute name="attributeMap">
            <Value>EmailAddress=EmailAddress</Value>
            <Value>Telephone=Telephone</Value>
        </Attribute>
        <Attribute name="saeSPUrl">
            <Value>https://lb4.sp-example.com:1081/opensso/spsaehandler/
             metaAlias/sp</Value>
        </Attribute>
        <Attribute name="responseArtifactMessageEncoding">
            <Value>URI</Value>
        </Attribute>
        <Attribute name="idpProxyCount">
            <Value>0</Value>
        </Attribute>
        <Attribute name="basicAuthUser">
            <Value/>
        </Attribute>
        <Attribute name="useIntroductionForIDPProxy">
            <Value>false</Value>
        </Attribute>
        <Attribute name="wantArtifactResponseSigned">
            <Value/>
        </Attribute>
        <Attribute name="intermediateUrl">
            <Value/>
        </Attribute>
        <Attribute name="defaultRelayState">
            <Value/>
        </Attribute>
        <Attribute name="basicAuthPassword">
            <Value/>
        </Attribute>
        <Attribute name="wantPOSTResponseSigned">
            <Value/>
        </Attribute>
        <Attribute name="wantAttributeEncrypted">
            <Value/>
        </Attribute>
        <Attribute name="basicAuthOn">
            <Value>false</Value>
        </Attribute>
        <Attribute name="spAdapterEnv"/>
        <Attribute name="saeSPLogoutUrl">
            <Value>https://lb4.sp-example.com:1081/opensso/samples/
             saml2/sae/saeSPApp.jsp</Value>
        </Attribute>
        <Attribute name="ECPRequestIDPListFinderImpl">
            <Value>com.sun.identity.saml2.plugins.ECPIDPFinder</Value>
        </Attribute>
        <Attribute name="wantLogoutResponseSigned">
            <Value/>
        </Attribute>
        <Attribute name="wantLogoutRequestSigned">
            <Value/>
        </Attribute>
        <Attribute name="encryptionCertAlias">
            <Value/>
        </Attribute>
        <Attribute name="spAuthncontextMapper">
            <Value>com.sun.identity.saml2.plugins.DefaultSPAuthnContextMapper</Value>
        </Attribute>
        <Attribute name="assertionTimeSkew">
            <Value>300</Value>
        </Attribute>
        <Attribute name="ECPRequestIDPList"/>
        <Attribute name="autofedAttribute">
            <Value>mail</Value>
        </Attribute>
        <Attribute name="saeAppSecretList">
            <Value>url=https://lb4.sp-example.com:1081/opensso/samples/saml2/sae/
             saeSPApp.jsp|type=symmetric|secret=AQICIbz4afzilWzbmo6QD9lQ9
             U4kEBrMlvZy</Value>
        </Attribute>
    </SPSSOConfig>
    <IDPSSOConfig metaAlias="/idp">
        <Attribute name="description">
            <Value/>
        </Attribute>
        <Attribute name="signingCertAlias">
            <Value>test</Value>
        </Attribute>
        <Attribute name="encryptionCertAlias">
            <Value/>
        </Attribute>
        <Attribute name="basicAuthOn">
            <Value>false</Value>
        </Attribute>
        <Attribute name="basicAuthUser">
            <Value/>
        </Attribute>
        <Attribute name="basicAuthPassword">
            <Value/>
```

```
</Attribute>
<Attribute name="autofedEnabled">
    <Value>false</Value>
</Attribute>
<Attribute name="autofedAttribute">
    <Value/>
</Attribute>
<Attribute name="assertionEffectiveTime">
    <Value>600</Value>
</Attribute>
<Attribute name="idpAuthncontextMapper">
    <Value>com.sun.identity.saml2.plugins.DefaultIDPAuthnContextMapper</Value>
</Attribute>
<Attribute name="idpAuthncontextClassrefMapping">
    <Value>urn:oasis:names:tc:SAML:2.0:ac:classes:
      PasswordProtectedTransport|0||default</Value>
</Attribute>
<Attribute name="idpAccountMapper">
    <Value>com.sun.identity.saml2.plugins.DefaultIDPAccountMapper</Value>
</Attribute>
<Attribute name="idpAttributeMapper">
    <Value>com.sun.identity.saml2.plugins.DefaultIDPAttributeMapper</Value>
</Attribute>
<Attribute name="assertionIDRequestMapper">
    <Value>com.sun.identity.saml2.plugins.DefaultAssertionIDRequestMapper</Value>
</Attribute>
<Attribute name="nameIDFormatMap">
    <Value>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress=mail</Value>
    <Value>urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName=</Value>
    <Value>urn:oasis:names:tc:SAML:1.1:nameid-format:
     WindowsDomainQualifiedName=</Value>
    <Value>urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos=</Value>
    <Value>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=</Value>
</Attribute>
<Attribute name="idpECPSessionMapper">
    <Value>com.sun.identity.saml2.plugins.DefaultIDPECPSessionMapper</Value>
</Attribute>
<Attribute name="attributeMap"/>
<Attribute name="wantNameIDEncrypted">
    <Value/>
</Attribute>
<Attribute name="wantArtifactResolveSigned">
    <Value/>
</Attribute>
<Attribute name="wantLogoutRequestSigned">
    <Value/>
</Attribute>
```

```
        <Attribute name="wantLogoutResponseSigned">
            <Value/>
        </Attribute>
        <Attribute name="wantMNIRequestSigned">
            <Value/>
        </Attribute>
        <Attribute name="wantMNIResponseSigned">
            <Value/>
        </Attribute>
        <Attribute name="cotlist">
            <Value>spcot</Value>
        </Attribute>
        <Attribute name="discoveryBootstrappingEnabled">
            <Value>false</Value>
        </Attribute>
        <Attribute name="assertionCacheEnabled">
            <Value>false</Value>
        </Attribute>
        <Attribute name="assertionNotBeforeTimeSkew">
            <Value>600</Value>
        </Attribute>
        <Attribute name="saeAppSecretList"/>
        <Attribute name="saeIDPUrl">
            <Value>https://lb4.sp-example.com:1081/opensso/idpsaehandler/metaAlias/
             idp</Value>
        </Attribute>
        <Attribute name="AuthUrl">
            <Value/>
        </Attribute>
        <Attribute name="appLogoutUrl">
            <Value/>
        </Attribute>
    </IDPSSOConfig>
</EntityConfig>

Entity configuration was exported to file, web.
```

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

11.3 Configuring the Hosted Service Provider to Communicate with the Remote Identity Provider

After configuring the providers, enable the hosted service provider to communicate with the remote identity provider by loading the identity provider metadata into the instance of OpenSSO Enterprise acting as the service provider.

▼ To Import the Remote Identity Provider Metadata into the Hosted Service Provider

1 **Access** `https://lb4.sp-example.com:1081/opensso/console` **from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

The Common Tasks tab is displayed.

3 **Click Register Remote Identity Provider under Create SAML v2 Providers.**

The Create a SAML v2 Remote Identity Provider page is displayed.

4 **Make the following changes on the Create a SAML v2 Remote Identity Provider page.**

- Select the URL radio button for *Where does the metadata file reside?*
- Type **`https://lb2.idp-example.com:1081/opensso/saml2/jsp/exportmetadata.jsp`** as the value of *URL where metadata is located*.
- Under Circle of Trust, select the Add to Exiting radio button and select the **`spcot`** circle of trust from the drop down menu.

5 **Click Configure.**

6 **Select Finish to end the task.**

CHAPTER 12

Testing the SAML v2 Profiles

Following are the SAML v2 profiles used for testing the SAML v2 configurations.

- Federation
- Single Logout
- Single Sign On
- Federation Termination

SAML v2 profiles can be initiated from the service provider side or from the identity provider side of the deployment. There are two ways in which the SAML v2 configurations can be tested and the procedures for these options are in the following sections.

- "12.1 Using the OpenSSO Enterprise Common Tasks Wizard" on page 267
- "12.2 Using Specially Constructed URLs" on page 269

12.1 Using the OpenSSO Enterprise Common Tasks Wizard

This automated test uses the Test Federation Connectivity work flow option under the Common Tasks tab of the OpenSSO Enterprise console.

▼ To Test SAML v2 Using the Common Tasks Wizard

1 **Access** `https://lb2.idp-example.com:1081/opensso/console` **from a web browser.**

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

Username **`amadmin`**

Password **`ossoadmin`**

The Common Tasks tab is displayed.

3 **Under the Common Tasks tab, click Test Federation Connectivity.**

The Validate Federation Setup page is displayed.

4 **Select the radio button next to `idpcot`, the circle of trust that contains the providers you are testing.**

The providers in `idpcot` are displayed.

5 **Click Start Test.**

A pop up is displayed.

6 **Click OK on the pop up.**

Your administrator session is terminated and the test is run.

7 **When displayed, log in to the OpenSSO Enterprise console on the identity provider side with the following information.**

Username	**`idpuser`**
Password	**`idpuser`**

With successful authentication, the OpenSSO Enterprise console on the service provider side is displayed.

8 **Log in to the OpenSSO Enterprise console on the service provider side with the following information.**

Username	**`spuser`**
Password	**`spuser`**

With successful authentication, the two accounts are linked. Single logout follows the successful federation.

9 **When displayed to test single sign on, log in to the OpenSSO Enterprise console on the identity provider side with the following information.**

Username	**`idpuser`**
Password	**`idpuser`**

Following successful authentication on the identity provider side, the user is logged in to the service provider through a back channel, demonstrating single sign on. Finally, the user profile federation is terminated. Thus, the following has occurred:

- A user is successfully authenticated with two different providers and the user's separate profiles are federated.
- The user is logged out of both providers verifying single logout.

- The user is logged back in to both providers by providing credentials to only one of them verifying single sign on.
- The federation between the two user profiles is terminated.

10 Click Cancel to return to the OpenSSO Enterprise console login page.

12.2 Using Specially Constructed URLs

In this section, test SAML v2 communications for the following profiles and bindings using specially constructed URLs.

- Browser Artifact Profile (SOAP/HTTP)
- Browser POST Profile (SOAP/HTTP)
- Back Channel SOAP Over HTTP
- Front Channel HTTP

Tests can be initiated from the identity provider side or the service provider side. The following procedures provide the constructed URLs and procedures for accessing them.

- "12.2.1 Testing Identity Provider Initiated URLs" on page 269
- "12.2.2 Testing Service Provider Initiated URLs" on page 273

12.2.1 Testing Identity Provider Initiated URLs

The following tests are initiated on the identity provider side to test SAML v2 communications with the service provider.

- "12.2.1.1 Testing Persistent Federation" on page 269
- "12.2.1.2 Testing Single Logout" on page 271
- "12.2.1.3 Testing Single Sign On" on page 271
- "12.2.1.4 Testing Federation Termination" on page 272

12.2.1.1 Testing Persistent Federation

Name identifiers are used by the identity provider and the service provider to communicate with each other regarding a user. In this test, a *persistent* identifier is used to federate the identity provider's user profile with the same user's profile on the service provider side.

- "To Test Persistent Federation Using the Browser Artifact Profile" on page 270
- "To Test Persistent Federation Using the Browser POST Profile" on page 270

▼ To Test Persistent Federation Using the Browser Artifact Profile

1. **Enter the persistent federation URL in a web browser:**
   `https://lb2.idp-example.com:1081/opensso/saml2/jsp/idpSSOInit.jsp?metaAlias=/idp&spEntityID=https://lb4.sp-example.com:1081/opensso`.

   The request is directed to OpenSSO Enterprise on the service provider side.

2. **Log in to the OpenSSO Enterprise console as a test user.**

   User Name: **`spuser`**

   Password: **`spuser`**

   The login request is redirected to OpenSSO Enterprise on the identity provider side.

3. **Log in to the OpenSSO Enterprise console as a test user.**

   User Name: **`idpuser`**

   User Name: **`idpuser`**

   The browser message "Single Sign-On succeeded" is displayed confirming that federation has succeeded.

4. **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

▼ To Test Persistent Federation Using the Browser POST Profile

1. **Enter the persistent federation URL in a web browser:**
   `https://lb2.idp-example.com:1081/opensso/saml2/jsp/idpSSOInit.jsp?metaAlias=/idp&spEntityID=https://lb4.sp-example.com:1081/opensso&binding=HTTP-POST`.

   The request is directed to OpenSSO Enterprise on the service provider side.

2. **Log in to the OpenSSO Enterprise console as a test user.**

   User Name: **`spuser`**

   Password: **`spuser`**

   The login request is redirected to OpenSSO Enterprise on the identity provider side.

3. **Log in to the OpenSSO Enterprise console as a test user.**

   User Name: **`idpuser`**

   User Name: **`idpuser`**

The browser message “Single Sign-On succeeded” is displayed confirming that federation has succeeded.

4 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

12.2.1.2 Testing Single Logout

Single logout permits session termination of all participants in the session. The logout request can be initiated by any participant in the session.

- “To Test Single Logout Using Back Channel SOAP Over HTTP” on page 271
- “To Test Single Logout Using Front Channel HTTP” on page 271

▼ To Test Single Logout Using Back Channel SOAP Over HTTP

1 **Enter the single logout URL in a web browser:**

```
https://lb2.idp-example.com:1081/opensso/saml2/jsp/idpSingleLogoutInit.jsp?
metaAlias=/idp&spEntityID=https://lb4.sp-example.com:1081/opensso&
binding=urn:oasis:names:tc:SAML:2.0:bindings:SOAP
```

The browser message “IDP initiated single logout succeeded” is displayed.

2 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

▼ To Test Single Logout Using Front Channel HTTP

1 **Enter the single logout URL in a web browser:**

```
https://lb2.idp-example.com:1081/opensso/saml2/jsp/idpSingleLogoutInit.jsp?
metaAlias=/idp&spEntityID=https://lb4.sp-example.com:1081/opensso
```

The message “IDP initiated single logout succeeded” is displayed.

2 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

12.2.1.3 Testing Single Sign On

In this test, the user accomplishes single sign on through the back channel.

- “To Test Single Sign-On Using the Browser Artifact Profile” on page 272
- “To Test Single Sign-On Using the Browser POST Profile” on page 272

▼ To Test Single Sign-On Using the Browser Artifact Profile

1 **Enter the single sign on URL in a web browser:**

`https://lb2.idp-example.com:1081/opensso/saml2/jsp/idpSSOInit.jsp?metaAlias=/idp&spEntityID=https://lb4.sp-example.com:1081/opensso`.

The request is directed to OpenSSO Enterprise on the service provider side.

2 **Log in to the OpenSSO Enterprise console as a test user.**

User Name: **`spuser`**

Password: **`spuser`**

The browser message "Single Sign-On succeeded" is displayed.

3 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`**.**

▼ To Test Single Sign-On Using the Browser POST Profile

1 **Enter the single sign on URL in a web browser:**

`https://lb2.idp-example.com:1081/opensso/saml2/jsp/idpSSOInit.jsp?metaAlias=/idp&spEntityID=https://lb4.sp-example.com:1081/opensso&binding=HTTP-POST`.

The login request is redirected to Access Manager.

2 **Log in to the OpenSSO Enterprise console as a test user.**

User Name: **`spuser`**

Password: **`spuser`**

The browser message "Single Sign-On succeeded" is displayed.

3 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`**.**

12.2.1.4 Testing Federation Termination

In this test, the federation previously authorized is terminated.

- "To Test Federation Termination Using Back Channel SOAP Over HTTP" on page 273
- "To Test Federation Termination Using Front Channel HTTP" on page 273

To Test Federation Termination Using Back Channel SOAP Over HTTP

1 **Enter the federation termination URL in a web browser:**

`https://lb2.idp-example.com:1081/opensso/saml2/jsp/idpMNIRequestInit.jsp?metaAlias=/idp&spEntityID=https://lb4.sp-example.com:1081/opensso&binding=urn:oasis:names:tc:SAML:2.0:bindings:SOAP&requestType=Terminate`.

The browser message "ManageNameID Request succeeded" is displayed confirming the federation has been terminated.

2 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

To Test Federation Termination Using Front Channel HTTP

1 **Enter the federation termination URL in a web browser:**

`https://lb2.idp-example.com:1081/opensso/saml2/jsp/idpMNIRequestInit.jsp?metaAlias=/idp&spEntityID=https://lb4.sp-example.com:1081/opensso&requestType=Terminate`.

The browser message "ManageNameID Request succeeded" is displayed confirming the federation has been terminated.

2 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

12.2.2 Testing Service Provider Initiated URLs

The following tests are initiated on the service provider side to test SAML v2 communications with the identity provider.

- "12.2.2.1 Testing Persistent Federation" on page 273
- "12.2.2.2 Testing Single Logout" on page 275
- "12.2.2.3 Testing Single Sign On" on page 275
- "12.2.2.4 Testing Federation Termination" on page 276

12.2.2.1 Testing Persistent Federation

Name identifiers are used by the identity provider and the service provider to communicate with each other regarding a user. In this test, a *persistent* identifier is used to federate the identity provider's user profile with the same user's profile on the service provider side.

- "To Test Persistent Federation Using the Browser Artifact Profile" on page 274
- "To Test Persistent Federation Using the Browser POST Profile" on page 274

▼ To Test Persistent Federation Using the Browser Artifact Profile

1 **Enter the persistent federation URL in a web browser:** `https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSSOInit.jsp?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1081/opensso`.

The request is directed to OpenSSO Enterprise on the identity provider side for authentication.

2 **Log in to the OpenSSO Enterprise console as test user.**

User Name:	**`idpuser`**
Password:	**`idpuser`**

The request is redirected to OpenSSO Enterprise on the service provider side.

3 **Log in to the OpenSSO Enterprise console as the test user.**

User Name:	**`spuser`**
User Name:	**`spuser`**

The browser message "Single Sign-On succeeded" is displayed confirming federation has succeeded.

4 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

▼ To Test Persistent Federation Using the Browser POST Profile

1 **Enter the persistent federation URL in a web browser:** `https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSSOInit.jsp?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1081/opensso&binding=HTTP-POST`.

The request is directed to OpenSSO Enterprise on the identity provider side for authentication.

2 **Log in to the OpenSSO Enterprise console as a test user.**

User Name:	**`idpuser`**
Password:	**`idpuser`**

The request is redirected to OpenSSO Enterprise on the service provider side.

3 **Log in to the OpenSSO Enterprise console as a test user.**

User Name:	**`spuser`**
User Name:	**`spuser`**

The browser message “Single Sign-On succeeded” is displayed confirming federation has succeeded.

4 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

12.2.2.2 Testing Single Logout

Single logout permits session termination of all participants in the session. The logout request can be initiated by any participant in the session.

- “To Test Single Logout Using Back Channel SOAP Over HTTP” on page 275
- “To Test Single Logout Using Front Channel HTTP” on page 275

▼ To Test Single Logout Using Back Channel SOAP Over HTTP

1 **Enter the single logout URL in a web browser:**

```
https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSSOInit.jsp?
metaAlias=/sp&binding=urn:oasis:names:tc:SAML:2.0:bindings:SOAP&
idpEntityID=https://lb2.idp-example.com:1081/opensso.
```

The message “SP initiated single logout succeeded” is displayed and both user profile sessions are ended.

2 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

▼ To Test Single Logout Using Front Channel HTTP

1 **Enter the single logout URL in a web browser:**

```
https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSSOInit.jsp?
metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1081/opensso.
```

The message “SP initiated single logout succeeded” is displayed and both user profile sessions are ended.

2 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

12.2.2.3 Testing Single Sign On

In this test, the user accomplishes single sign on through the back channel.

- “To Test Single Sign On Using the Browser Artifact Profile” on page 276
- “To Test Single Sign-On Using the Browser POST Profile” on page 276

▼ To Test Single Sign On Using the Browser Artifact Profile

1. **Enter the single sign on URL in a web browser:**

   `https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSSOInit.jsp?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1081/opensso`.

   The request is directed to OpenSSO Enterprise on the identity provider side for authentication.

2. **Log in to the OpenSSO Enterprise console as a test user.**

   User Name: **`idpuser`**

   Password: **`idpuser`**

   The browser message "Single Sign-On succeeded" is displayed.

3. **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

▼ To Test Single Sign-On Using the Browser POST Profile

1. **Enter the single sign on URL in a web browser:**

   `https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSSOInit.jsp?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1081/opensso&binding=HTTP-POST`.

   The request is directed to OpenSSO Enterprise on the identity provider side for authentication.

2. **Log in to the OpenSSO Enterprise console as a test user.**

   User Name: **`idpuser`**

   Password: **`idpuser`**

   The message "Single Sign-On succeeded" is displayed.

3. **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

12.2.2.4 Testing Federation Termination

In this test, the federation previously authorized is terminated.

- "To Terminate Federation Using Back Channel SOAP Over HTTP" on page 277
- "To Terminate Federation Using Front Channel HTTP" on page 277

▼ To Terminate Federation Using Back Channel SOAP Over HTTP

1 **Enter the federation termination URL in a web browser:**
`https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSSOInit.jsp?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1081/opensso&requestType=Terminate&binding=urn:oasis:names:tc:SAML:2.0:bindings:SOAP`.

The browser message "ManageNameID Request succeeded" is displayed confirming the federation has been terminated.

2 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

▼ To Terminate Federation Using Front Channel HTTP

1 **Enter the federation termination URL in a web browser:**
`https://lb4.sp-example.com:1081/opensso/saml2/jsp/spSSOInit.jsp?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1081/opensso&requestType=Terminate`.

The browser message "ManageNameID Request succeeded" is displayed confirming the federation has been terminated.

2 **(Optional) To view the SAML v2 assertion used, see the debug file in** `/export/ossoadm/config/opensso/debug/Federation`.

CHAPTER 13

Testing Secure Attribute Exchange

Secure Attribute Exchange (also referred to as Virtual Federation Proxy) provides a mechanism for one application to communicate identity information to a second application in a different domain. More specifically, it provides a secure gateway that enables legacy applications to communicate authentication attributes without having to deal with federation protocols and processing. Secure Attribute Exchange uses SAML v2 to transfer identity data between the communicating entities. This chapter contains the following sections for setting up and testing Secure Attribute Exchange.

Note – This chapter assumes you have completed Part II and Part III; in effect, creating two domains that can communicate using SAML v2. In this test, we use symmetric key encryption (one shared secret is used for both encryption and decryption) between all providers and applications.

13.1 Patching the Secure Attribute Exchange Host Machines

Patch the host machines that will be used to deploy the sample Secure Attribute Exchange JavaServer Pages application (bundled with OpenSSO Enterprise Client SDK). Towards this end, use different web containers from those on which OpenSSO Enterprise is installed. On our lab machines, the required Application Server patch is 117461–08. Results for your machine might be different. Read the latest documentation for your web container to determine if you need to install patches and, if so, what they might be. You can search for patches directly at http://sunsolve.sun.com. Navigate to the PatchFinder page, enter the patch number, click Find Patch, and download the appropriate patch for the Secure Attribute Exchange identity provider application host machine (sae.idp-example.com) and the Secure Attribute Exchange service provider application host machine (sae.sp-example.com).

▼ To Patch the OpenSSO Enterprise Host Machines

1 **Log in to the sae.idp-example.com host machine as a root user.**

2 **Run patchadd to see if the patch is already installed.**

```
# patchadd -p | grep 117461-08
```

A series of patch numbers are displayed, and patch 117461–08 is present so there is no need to install any patches at this time.

3 **Log out of the sae.idp-example.com host machine.**

4 **Log in to the sae.sp-example.com host machine as a root user.**

5 **Run patchadd to see if the patch is already installed.**

```
# patchadd -p | grep 117461-08
```

A series of patch numbers are displayed, and patch 117461–08 is present so there is no need to install any patches at this time.

6 **Log out of the sae.sp-example.com host machine.**

13.2 Installing Application Server on the Secure Attribute Exchange Identity Provider Host Machine

To test a Secure Attribute Exchange we configure and use JavaServer Pages (bundled with the OpenSSO Enterprise Client SDK) to emulate real world applications. saeIDPApp.jsp represents the identity provider application that will invoke a remote service provider application and pass attributes to it. It will be installed on the sae.idp-example.com host machine. The following procedures will install and configure one instance of Application Server as the web container for the identity provider application.

- “To Install Application Server on the Secure Attribute Exchange Identity Provider Host Machine” on page 281
- “To Secure Communications from the Identity Provider Host Machine” on page 284
- “To Modify the Identity Provider Web Container domain.xml Configuration File” on page 286

- “To Deploy the Client SDK on the Identity Provider Host Machine” on page 287

To Install Application Server on the Secure Attribute Exchange Identity Provider Host Machine

Before You Begin This procedure assumes you have completed “13.1 Patching the Secure Attribute Exchange Host Machines” on page 279.

1 **Log in to the `sae.idp-example.com` host machine as a root user.**

2 **Create a directory into which the Application Server bits can be downloaded and change into it.**

```
# mkdir /export/AS91
# cd /export/AS91
```

3 **Download the Sun Java System Application Server 9.1 Update 1 binary from the Sun Microsystems Product Download page to the `/export/AS91` directory.**

4 **Grant the downloaded binary execute permission using the `chmod` command.**

```
# chmod +x sjsas-9_1_01-solaris-sparc.bin
```

5 **Install the software.**

```
# ./sjsas-9_1_01-solaris-sparc.bin -console
```

6 **When prompted, provide the following information.**

`You are running the installation program for the Sun Java System Application Server. This program asks you to supply configuration preference settings that it uses to install the server.`
`This installation program consists of one or more selections that provide you with information and let you enter preferences that determine how Sun Java System Application Server is installed and configured.`
`When you are presented with the following question, the installation process pauses to allow you to read the information that has been presented When you are ready to continue, press Enter.`
	Press Enter to continue.

Have you read, and do you accept, all of the terms of the preceding Software License Agreement [no] {"<" goes back, "!" exits}?	Enter **yes**.
Installation Directory [/opt/SUNWappserver] {"<" goes back, "!" exits}	Enter **/opt/SUNWappserver91**
The specified directory "/opt/SUNWappserver91" does not exist. Do you want to create it now or choose another directory? 1. Create Directory 2. Choose New. Enter the number corresponding to your choice [1] {"<" goes back, "!" exits}	Enter **1** to create the directory.
The Sun Java System Application Server requires a Java 2 SDK. Please provide the path to a Java 2 SDK 5.0 or greater. [/usr/jdk/instances/jdk1.5.0] {"<" goes back, "!" exits}	Press Enter to accept the default value.
Supply the admin user's password and override any of the other initial configuration settings as necessary. Admin User [admin] {"<" goes back, "!" exits}	Press Enter to accept the default value.
Admin User's Password (8 chars minimum): Re-enter Password:	Enter **domain1pwd** and then re-enter **domain1pwd**.
Do you want to store admin user name and password in .asadminpass file in user's home directory [yes] {"<" goes back, "!" exits}?	Press Enter to accept the default value.

Admin Port [4848] {"<" goes back, "!" exits}
HTTP Port [8080] {"<" goes back, "!" exits}
HTTPS Port [8181] {"<" goes back, "!" exits}
	Press Enter to accept the three default values.

Do you want to enable Updatecenter client [yes] {"<" goes back, "!" exits}?	Press Enter to accept the default value.
Do you want to upgrade from previous Application Server version [no] {"<" goes back, "!" exits}?	Press Enter to accept the default value.

```
The following items for the product Sun Java
System Application Server will be installed:

Product: Sun Java System Application Server
Location: /opt/SUNWappserver91
Space Required: 161.61 MB
-------------------------------------------
Sun Java System message Queue 4.1
Application Server
Startup

Ready To Install

1. Install Now
2. Start Over
3. Exit Installation

What would you like to do [1]
{"<" goes back, "!" exits}?
```

Press Enter to accept the default value and begin the installation process.

```
- Installing Sun Java System Application
Server

|-1%-----25%-----50%-----75%-----100%|

 - Installation Successful.
```

When installation is complete, an `Installation Successful` message is displayed:

```
Next Steps:

1. Access the About Application Server 9.1 welcome
page at:
 file:///opt/SUNWappserver91/docs/about.html

2. Start the Application Server by executing:
  /opt/SUNWappserver91/bin/asadmin
  start-domain domain1

3. Start the Admin Console:
  http://sae.idp-example.com:4848

Please press Enter/Return key to exit the
installation program. {"!" exits}
```

Press Enter to exit the installation program.

7 Log out of the `sae.idp-example.com` host machine.

▼ To Secure Communications from the Identity Provider Host Machine

Create a request for a server certificate and import the certificate authority (CA) root certificate and server certificate to the keystore. This will secure communications initiated by the identity provider application.

Before You Begin Backup `domain.xml` before modifying it.

1 **Log in to the `sae.idp-example.com` host machine as a root user.**

2 **Generate a private/public key pair and reference it with the alias, `sae-idp`.**

`sae-idp` will be used in a later step to retrieve the public key which is contained in a self-signed certificate.

```
# cd /opt/SUNWappserver91/domains/domain1/config
# keytool -genkey -noprompt -keyalg rsa -keypass changeit
-alias sae-idp -keystore keystore.jks -dname "CN=sae.idp-example.com,
OU=OpenSSO, O=Sun Microsystems, L=Santa Clara, ST=California, C=US"
-storepass changeit
```

3 **Verify that the key pair was successfully created and stored in the certificate store using the following command.**

```
# keytool -list -v -keystore keystore.jks -storepass changeit
```

The output of this command lists a key entry with the alias `sae-idp`.

Note – The output of this command may list more than one certificate based on the entries in the keystore.

---

4 **Generate a server certificate request.**

```
# keytool -certreq -alias sae-idp -keypass changeit
-keystore keystore.jks -storepass changeit file sae-idp.csr
```

`sae-idp.csr` is the server certificate request.

5 **(Optional) Verify that `sae-idp.csr` was created.**

```
# ls -la sae-idp.csr

 -rw-r--r--   1 osso80adm staff        715 Apr  4 15:04 sae-idp.csr
```

6 **Send `sae-idp.csr` to the CA of your choice.**

The CA issues and returns a certified server certificate named `sae-idp.cer`.

7 Import ca.cer, the CA root certificate, into the certificate store.

The root certificate must be imported into two keystores (keystore.jks and cacerts.jks) with Application Server.

```
# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore keystore.jks -storepass changeit

Owner: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore

# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore cacerts.jks -storepass changeit

Owner: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore
```

8 Replace the self-signed public key certificate (associated with the s1as alias) with the server certificate received from the CA.

```
# keytool -import -file sae-idp.cer -alias sae-idp
-keystore keystore.jks -storepass changeit

Certificate reply was installed in keystore
```

9 **(Optional) Verify that the self-signed public key certificate has been overwritten by the server certificate received from the CA.**

```
# keytool -list -alias sae-idp -v -keystore keystore.jks
-storepass changeit

The certificate indicated by the alias "sae-idp" is signed by CA.
```

10 **Change the certificate alias from the default `s1as` to the new `sae-idp` in the `domain.xml` file for the `domain1` domain.**

The Application Server configuration file is `domain.xml`.

```
<http-listener acceptor-threads="1" address="0.0.0.0"
blocking-enabled="false" default-virtual-server="server" enabled="true"
family="inet" id="http-listener-2" port="1081" security-enabled="true"
server-name="" xpowered-by="true">
<ssl cert-nickname="sae-idp" client-auth-enabled="false" ssl2-enabled="false"
ssl3-enabled="true" tls-enabled="true" tls-rollback-enabled="true"/>
```

▼ To Modify the Identity Provider Web Container `domain.xml` Configuration File

Modify the following Java Virtual Machine (JVM) options in the Application Server configuration file, `domain.xml` to prepare for the installation of the Client SDK.

Before You Begin

- This procedure assumes you are still logged in as the root user to the `sae-idp` host machine.
- Backup `domain.xml` before modifying it.

1 **Change to the `config` directory.**

```
# cd /opt/SUNWappserver91/domains/domain1/config
```

2 **Open `domain.xml` in a text editor and make the following changes:**

- Replace **`<jvm-options>-client</jvm-options>`** with **`<jvm-options>-server</jvm-options>`**.
- Replace **`<jvm-options>-Xmx512m</jvm-options>`** with **`<jvm-options>-Xmx1024m</jvm-options>`**.

3 **Save the file and close it.**

4 **Restart the `domain1` domain.**

```
# cd /opt/SUNWappserver91/bin
# ./asadmin stop-domain

Server was successfully stopped.
```

```
./asadmin start-domain

Redirecting output to /opt/SUNWappserver91/domains/domain1/logs/server.log
```

5 **Verify that the certificate used for SSL communication is the root CA certificate.**

a. **Access** `https://sae.idp-example.com/index.html` **from a web browser.**

b. **View the details of the certificate in the security warning to ensure that it is Issued by "OpenSSLTestCA".**

After inspecting and accepting the certificate, you should see the default `index.html` page.

c. **Close the browser.**

▼ To Deploy the Client SDK on the Identity Provider Host Machine

When you deploy the Client SDK, you also deploy the `saeIDPApp.jsp`.

Before You Begin This procedure assumes you are still logged in as the root user to the `sae-idp` host machine.

1 **Get the Client SDK WAR using the following sub procedure.**

a. **Log in to the** `osso1.idp-example.com` **host machine.**

b. **Change to the** `/export/OSSO_BITS/opensso/samples/war` **directory.**

c. **Copy** `opensso-client-jdk15.war` **to the** `/export/OSSO_BITS/opensso/samples/war` **directory on the** `sae.idp-example.com` **host machine.**

d. **Log out of the** `osso1.idp-example.com` **host machine.**

2 **Access** `http://sae.idp-example.com:4848/login.jsf` **from a web browser.**

User Name: **`admin`**

Password: **`domain1pwd`**

3 **Click Web Applications in the left frame of Application Server.**

4 **Click Deploy.**

The Deploy Enterprise Applications/Modules page is displayed.

5 **Click the radio button next to Packaged file to be uploaded to the server and browse for the** `opensso-client-jdk15.war` **WAR in the** `/export/OSSO_BITS/opensso/samples/war` **directory.**

6 **Enter** `opensso-client` **as the Application Name.**

7 **Click OK to deploy the Client SDK.**

8 **(Optional) List the contents of the** `j2ee-modules` **directory to verify that the WAR was successfully deployed.**

a. **Change to the** `/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules` **directory.**

b. **List the contents of the directory.**

```
# ls -al

total 6
drwxr-xr-x 3 root staff 512 Aug 15 14:01 .
drwxr-xr-x 6 root staff 512 Aug 15 14:55 ..
drwxr-xr-x 21 root staff 1024 Aug 15 14:01 opensso-client
```

9 **Log out of the** `sae.idp-example.com` **host machine.**

Next Steps Add the IP address and host machine names to the `/etc/hosts` file on both the `sae.idp-example.com` and the `sae.sp-example.com` host machines as well as the host machine on which the browser is located.

13.3 Installing Application Server on the Secure Attribute Exchange Service Provider Host Machine

To test a Secure Attribute Exchange we configure and use JavaServer Pages (bundled with the OpenSSO Enterprise Client SDK) to emulate real world applications. `saeSPApp.jsp` represents the service provider application that will receive the attributes from the identity provider. It will be installed on the `sae.sp-example.com` host machine. The following procedures will install and configure one instance of Application Server as the web container for the service provider application.

- "To Install Application Server on the Secure Attribute Exchange Service Provider Host Machine" on page 289
- "To Secure Communications from the Service Provider Application" on page 292
- "To Modify the Service Provider Web Container `domain.xml` Configuration File" on page 294

- "To Deploy the Client SDK on the Service Provider Host Machine" on page 295

▼ To Install Application Server on the Secure Attribute Exchange Service Provider Host Machine

Before You Begin This procedure assumes you have completed "13.1 Patching the Secure Attribute Exchange Host Machines" on page 279.

1. **Log in to the `sae.sp-example.com` host machine as a root user.**

2. **Create a directory into which the Application Server bits can be downloaded and change into it.**

```
# mkdir /export/AS91
# cd /export/AS91
```

3. **Download the Sun Java System Application Server 9.1 Update 1 binary from the Sun Microsystems Product Download page to the `/export/AS91` directory.**

4. **Grant the downloaded binary execute permission using the `chmod` command.**

```
# chmod +x sjsas-9_1_01-solaris-sparc.bin
```

5. **Install the software.**

```
# ./sjsas-9_1_01-solaris-sparc.bin -console
```

6. **When prompted, provide the following information.**

`You are running the installation program for the Sun Java System Application Server. This program asks you to supply configuration preference settings that it uses to install the server.`
 	
`This installation program consists of one or more selections that provide you with information and let you enter preferences that determine how Sun Java System Application Server is installed and configured.`
 	
`When you are presented with the following question, the installation process pauses to allow you to read the information that has been presented When you are ready to continue, press Enter.`
	Press Enter to continue.

Have you read, and do you accept, all of the terms of the preceding Software License Agreement [no] {"<" goes back, "!" exits}?	Enter **yes**.
Installation Directory [/opt/SUNWappserver] {"<" goes back, "!" exits}	Enter **/opt/SUNWappserver91**
The specified directory "/opt/SUNWappserver91" does not exist. Do you want to create it now or choose another directory? 1. Create Directory 2. Choose New. Enter the number corresponding to your choice [1] {"<" goes back, "!" exits}	Enter **1** to create the directory.
The Sun Java System Application Server requires a Java 2 SDK. Please provide the path to a Java 2 SDK 5.0 or greater. [/usr/jdk/instances/jdk1.5.0] {"<" goes back, "!" exits}	Press Enter to accept the default value.
Supply the admin user's password and override any of the other initial configuration settings as necessary. Admin User [admin] {"<" goes back, "!" exits}	Press Enter to accept the default value.
Admin User's Password (8 chars minimum): Re-enter Password:	Enter **domain1pwd** and then re-enter **domain1pwd**.
Do you want to store admin user name and password in .asadminpass file in user's home directory [yes] {"<" goes back, "!" exits}?	Press Enter to accept the default value.

Admin Port [4848] {"<" goes back, "!" exits}
HTTP Port [8080] {"<" goes back, "!" exits}
HTTPS Port [8181] {"<" goes back, "!" exits}
	Press Enter to accept the three default values.

Do you want to enable Updatecenter client [yes] {"<" goes back, "!" exits}?	Press Enter to accept the default value.
Do you want to upgrade from previous Application Server version [no] {"<" goes back, "!" exits}?	Press Enter to accept the default value.

```
The following items for the product Sun Java
System Application Server will be installed:

Product: Sun Java System Application Server
Location: /opt/SUNWappserver91
Space Required: 161.61 MB
-------------------------------------------
Sun Java System message Queue 4.1
Application Server
Startup

Ready To Install

1. Install Now
2. Start Over
3. Exit Installation

What would you like to do [1]
{"<" goes back, "!" exits}?
```

Press Enter to accept the default value and begin the installation process.

```
- Installing Sun Java System Application
Server

|-1%-----25%-----50%-----75%-----100%|

 - Installation Successful.
```

When installation is complete, an `Installation Successful` message is displayed:

```
Next Steps:

1. Access the About Application Server 9.1 welcome
page at:
 file:///opt/SUNWappserver91/docs/about.html

2. Start the Application Server by executing:
  /opt/SUNWappserver91/bin/asadmin
  start-domain domain1

3. Start the Admin Console:
  http://sae.sp-example.com:4848

Please press Enter/Return key to exit the
installation program. {"!" exits}
```

Press Enter to exit the installation program.

7 **Log out of the** `sae.sp-example.com` **host machine.**

▼ To Secure Communications from the Service Provider Application

Create a request for a server certificate and import the certificate authority (CA) root certificate and server certificate to the keystore. This will secure communications initiated by the service provider application.

Before You Begin Backup `domain.xml` before modifying it.

1 **Log in to the `sae.sp-example.com` host machine as a root user.**

2 **Generate a private/public key pair and reference it with the alias, `sae-sp`.**

`sae-sp` will be used in a later step to retrieve the public key which is contained in a self-signed certificate.

```
# cd /opt/SUNWappserver91/domains/domain1/config
# keytool -genkey -noprompt -keyalg rsa -keypass changeit
-alias sae-sp -keystore keystore.jks -dname "CN=sae.sp-example.com,
OU=OpenSSO, O=Sun Microsystems, L=Santa Clara, ST=California, C=US"
-storepass changeit
```

3 **Verify that the key pair was successfully created and stored in the certificate store using the following command.**

```
# keytool -list -v -keystore keystore.jks -storepass changeit
```

The output of this command lists a key entry with the alias `sae-sp`.

Note – The output of this command may list more than one certificate based on the entries in the keystore.

---

4 **Generate a server certificate request.**

```
# keytool -certreq -alias sae-sp -keypass changeit
-keystore keystore.jks -storepass changeit file sae-sp.csr
```

`sae-sp.csr` is the server certificate request.

5 **(Optional) Verify that `sae-sp.csr` was created.**

```
# ls -la sae-sp.csr

 -rw-r--r--   1 osso80adm staff        715 Apr  4 15:04 sae-sp.csr
```

6 **Send `sae-sp.csr` to the CA of your choice.**

The CA issues and returns a certified server certificate named `sae-sp.cer`.

7 Import ca.cer, the CA root certificate, into the certificate store.

The root certificate must be imported into two keystores (keystore.jks and cacerts.jks) with Application Server.

```
# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore keystore.jks -storepass changeit

Owner: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore

# keytool -import -trustcacerts -alias OpenSSLTestCA
-file ca.cer -keystore cacerts.jks -storepass changeit

Owner: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Issuer: EMAILADDRESS=nobody@nowhere.com, CN=openssltestca, OU=am,
  O=sun, L=santa clara, ST=california, C=us
Serial number: f59cd13935f5f498
Valid from: Thu Sep 20 11:41:51 PDT 2007 until: Thu Jun 17 11:41:51 PDT 2010
Certificate fingerprints:
  MD5:  78:7D:F0:04:8A:5B:5D:63:F5:EC:5B:21:14:9C:8A:B9
  SHA1: A4:27:8A:B0:45:7A:EE:16:31:DC:E5:32:46:61:9E:B8:A3:20:8C:BA

Trust this certificate? [no]: Yes

Certificate was added to keystore
```

8 Replace the self-signed public key certificate (associated with the s1as alias) with the server certificate received from the CA.

```
# keytool -import -file sae-sp.cer -alias sae-sp
-keystore keystore.jks -storepass changeit

Certificate reply was installed in keystore
```

9 **(Optional) Verify that the self-signed public key certificate has been overwritten by the server certificate received from the CA.**

```
# keytool -list -alias sae-sp -v -keystore keystore.jks
-storepass changeit

The certificate indicated by the alias "sae-sp" is signed by CA.
```

10 **Change the certificate alias from the default `s1as` to the new `sae-sp` in the `domain.xml` file for the `domain1` domain.**

The Application Server configuration file is `domain.xml`.

```
<http-listener acceptor-threads="1" address="0.0.0.0"
blocking-enabled="false" default-virtual-server="server" enabled="true"
family="inet" id="http-listener-2" port="1081" security-enabled="true"
server-name="" xpowered-by="true">
<ssl cert-nickname="sae-sp" client-auth-enabled="false" ssl2-enabled="false"
ssl3-enabled="true" tls-enabled="true" tls-rollback-enabled="true"/>
```

▼ To Modify the Service Provider Web Container `domain.xml` Configuration File

Modify the following Java Virtual Machine (JVM) options in the Application Server configuration file, `domain.xml` to prepare for the installation of the Client SDK.

Before You Begin

- This procedure assumes you are still logged in as the root user to the `sae-sp` host machine.
- Backup `domain.xml` before modifying it.

1 **Change to the `config` directory.**

```
# cd /opt/SUNWappserver91/domains/domain1/config
```

2 **Open `domain.xml` in a text editor and make the following changes:**

- Replace **`<jvm-options>-client</jvm-options>`** with **`<jvm-options>-server</jvm-options>`**.
- Replace **`<jvm-options>-Xmx512m</jvm-options>`** with **`<jvm-options>-Xmx1024m</jvm-options>`**.

3 **Save the file and close it.**

4 **Restart the `domain1` domain.**

```
# cd /opt/SUNWappserver91/bin
# ./asadmin stop-domain

Server was successfully stopped.
```

```
./asadmin start-domain

Redirecting output to /opt/SUNWappserver91/domains/domain1/logs/server.log
```

5 **Verify that the certificate used for SSL communication is the root CA certificate.**

a. **Access** `https://sae.sp-example.com/index.html` **from a web browser.**

b. **View the details of the certificate in the security warning to ensure that it is Issued by "OpenSSLTestCA".**

After inspecting and accepting the certificate, you should see the default `index.html` page.

c. **Close the browser.**

▼ To Deploy the Client SDK on the Service Provider Host Machine

When you deploy the Client SDK, you also deploy the `saeSPApp.jsp`.

Before You Begin This procedure assumes you are still logged in as the root user to the `sae-sp` host machine.

1 **Get the Client SDK WAR using the following sub procedure.**

a. **Log in to the** `osso1.sp-example.com` **host machine.**

b. **Change to the** `/export/OSSO_BITS/opensso/samples/war` **directory.**

c. **Copy** `opensso-client-jdk15.war` **to the** `/export/OSSO_BITS/opensso/samples/war` **directory on the** `sae.sp-example.com` **host machine.**

d. **Log out of the** `osso1.sp-example.com` **host machine.**

2 **Access** `http://sae.sp-example.com:4848/login.jsf` **from a web browser.**

User Name: **`admin`**

Password: **`domain1pwd`**

3 **Click Web Applications in the left frame of Application Server.**

4 **Click Deploy.**

The Deploy Enterprise Applications/Modules page is displayed.

5 **Click the radio button next to Packaged file to be uploaded to the server and browse for the** `opensso-client-jdk15.war` **WAR in the** `/export/OSSO_BITS/opensso/samples/war` **directory.**

6 **Enter** `opensso-client` **as the Application Name.**

7 **Click OK to deploy the Client SDK.**

8 **(Optional) List the contents of the** `j2ee-modules` **directory to verify that the WAR was successfully deployed.**

a. **Change to the** `/opt/SUNWappserver91/domains/domain1/applications/j2ee-modules` **directory.**

b. **List the contents of the directory.**

```
# ls -al

total 6
drwxr-xr-x 3 root staff 512 Aug 15 14:01 .
drwxr-xr-x 6 root staff 512 Aug 15 14:55 ..
drwxr-xr-x 21 root staff 1024 Aug 15 14:01 opensso-client
```

9 **Log out of the** `sae.sp-example.com` **host machine.**

Next Steps Add the IP address and host machine names to the `/etc/hosts` file on both the `sae.idp-example.com` and the `sae.sp-example.com` host machines as well as the host machine on which the browser is located.

13.4 Establishing Trust Between Communicating Entities

The following procedures will establish trust relationships between the communicating entities (in this case, the included JSP).

- "To Establish Trust Between OpenSSO Enterprise and the Application on the Identity Provider Side" on page 297
- "To Establish Trust Between OpenSSO Enterprise and the Application on the Service Provider Side" on page 299

▼ To Establish Trust Between OpenSSO Enterprise and the Application on the Identity Provider Side

Set up a trust relationship between `saeIDPApp.jsp`, the identity provider application, and OpenSSO Enterprise on the identity provider side.

Before You Begin Choose a shared secret for use between the identity provider application and the instance of OpenSSO Enterprise on the identity provider side; in this procedure, `secret12`.

1 **Make the following modifications to `saeIDPApp.jsp` and save the file.**

`saeIDPApp.jsp` is found in the *OpenSSO-Deploy-Base*`/samples/saml2/sae` directory.

- Change the value of `saeServiceURL` to **`https://lb2.idp-example.com:1081/opensso/idpsaehandler/metaAlias/idp`**.
- Change the value of `secret` to **`secret12`**.

Note – In a real deployment the application would store this shared secret in an encrypted file.

- Change the value of `spapp` to **`https://sae.sp-example.com:8181/opensso/saml2/sae/saeSPApp.jsp`**.

2 **Log in to the OpenSSO Enterprise console at `https://lb2.idp-example.com:1081/opensso` as the administrator.**

User Name: **`amadmin`**

Password: **`ossoadmin`**

3 **Access `https://lb2.idp-example.com:1081/opensso/encode.jsp` in a different browser window.**

This JSP encodes the shared secret.

4 **Enter `secret12` in the test field and click Encode.**

A string representing the identity provider's encoded password is displayed.

5 **Save the string for later use and close the browser window.**

In this case, `AQICrLO+CuXkZFna8uAS0/GiUUtwyQltVdw2`.

6 **From the OpenSSO Enterprise console, click the Federation tab.**

7 **Under Entity Providers, click** `https://lb2.idp-example.com:1081/opensso`**, the hosted identity provider.**

8 **Click the Advanced tab.**

9 **Under SAE Configuration, type the following in the New Value text box of the Per Application Security Configuration property and click Add.**

```
url=https://sae.idp-example.com:8181/opensso/saml2/sae/
saeIDPApp.jsp|type=symmetric|secret=AQICrLO+CuXkZFna8uAS0/GiUUtwyQltVdw2
```

10 **Click Save to save the profile.**

11 **Click the Assertion Processing tab.**

12 **Click the Attribute Mapper link.**

13 **Under the Attribute Map property, type the following New Values and click Add.**

- **`mail=mail`**
- **`branch=branch`**

These attributes will be sent as part of the SAML v2 assertion.

14 **Click Save to save the profile.**

15 **Click Back to return to the Federation tab.**

16 **Under Entity Providers, click** `https://lb4.sp-example.com:1081/opensso`**, the remote service provider.**

17 **Click the Advanced tab.**

18 **Under SAE Configuration, enter** `https://lb4.sp-example.com:1081/opensso/spsaehandler/metaAlias/sp` **in the SP URL field.**

19 **Under SAE Configuration again, enter** `https://sae.sp-example.com:8181/opensso/saml2/sae/saeSPApp.jsp` **in the SP Logout URL field.**

20 **Click Save to save the profile.**

21 **Click Back to return to the Federation tab.**

22 **Click the Access Control tab.**

23 **Under the Access Control tab, click `/ (Top Level Realm)`.**

24 **Click the Authentication tab.**

25 **Under General, click Advanced Properties.**

The Core profile page is displayed.

26 **Under User Profile, select the Ignored radio button and click Save.**

Note – This modification is specific to this deployment example only.

---

27 **Click Save to save the profile.**

28 **Click Back to Authentication.**

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

▼ To Establish Trust Between OpenSSO Enterprise and the Application on the Service Provider Side

Set up a trust relationship between OpenSSO Enterprise on the service provider side and `saeSPApp.jsp`, the service provider application.

Before You Begin Choose a shared secret for use between the service provider application and the instance of OpenSSO Enterprise on the service provider side; in this procedure, `secret12`.

1 **Log in to the OpenSSO Enterprise console at `https://lb4.sp-example.com:1081/opensso` as the administrator.**

User Name: **`amadmin`**

Password: **`ossoadmin`**

2 **Access `https://lb4.sp-example.com:1081/opensso/encode.jsp` in a different browser window.**

This JSP encodes the shared secret.

3 **Enter `secret12` and click Encode.**

A string representing the identity provider's encoded password is displayed.

4 **Save the string for later use and close the browser window.**

In this case, `AQICIbz4afzilWzbmo6QD9lQ9U4kEBrMlvZy`.

5 **From the OpenSSO Enterprise console, click the Federation tab.**

6 **Under Entity Providers, click** `https://lb4.sp-example.com:1081/opensso`**, the hosted service provider.**

7 **Click the Assertion Processing tab.**

8 **Under Attribute Mapper, add the following new values to the Attribute Map property.**

- **`mail=mail`**
- **`branch=branch`**

9 **Under Auto-Federation, check the Enabled box.**

10 **Also under Auto-Federation, enter `mail` in the Attribute field.**

The value of the Attribute property is the attribute previously mapped between the identity provider and the service provider allowing Auto-Federation to work.

11 **Click Save.**

12 **Click the Advanced tab.**

13 **Under SAE Configuration, type `https://lb4.sp-example.com:1081/opensso/spsaehandler/metaAlias/sp` as the value for the SP URL.**

14 **Type `https://sae.sp-example.com:8181/opensso/saml2/sae/saeSPApp.jsp` as the value for the SP Logout URL.**

15 **Type the following in the New Value field of the Per Application Security Configuration property and click Add.**

```
url=https://sae.sp-example.com:8181/opensso/saml2/sae/
saeSPApp.jsp|type=symmetric|secret=AQICIbz4afzilWzbmo6QD9lQ9U4kEBrMlvZy
```

16 **Click Save to save the profile.**

17 **Click Back to return to the Federation tab.**

18 **Click the Access Control tab.**

19 **Under the Access Control tab, click `/ (Top Level Realm)`.**

20 **Click the Authentication tab.**

21 **Under General, click Advanced Properties.**

The Core profile page is displayed.

22 **Under User Profile, select the Ignored radio button and click Save.**

Note – This modification is specific to this deployment example only.

23 **Click Save to save the profile.**

24 **Click Back to Authentication.**

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

13.5 Testing the Secure Attribute Exchange

In this test, `saeIDPApp.jsp` securely sends user authentication credentials to OpenSSO Enterprise on the identity provider side. The identity provider then uses basic SAML v2 to communicate these attributes to OpenSSO Enterprise on the service provider side. Finally, the service provider securely passes these same attributes to `saeSPApp.jsp`, the consumer.

Note – This test for Secure Attribute Exchange does not use the test users created in building the SP and IDP Environment. The values of Userid on local IDP, Authenticated auth level, mail attribute, and branch attribute are hard-coded in `saeIDPApp.jsp` as the default values for the test. Because we have not created the hard-coded test user on the service provider side, we previously set the User Profile to ignore on the service provider side.

▼ To Test the Secure Attribute Exchange Configurations

1 **Access `https://sae.idp-example.com:8181/opensso/saml2/sae/saeIDPApp.jsp` from a web browser.**

The Secure Attributes Exchange IDP APP SAMPLE page is displayed.

2 **Type the following values in the appropriate text field.**

Userid on local IDP
: **`testuser`**

Authenticated auth level
: **`0`**

mail attribute
: testuser@foo.com

branch attribute
: mainbranch

SP App URL
: **`https://sae.sp-example.com:8181/opensso/saml2/sae/saeSPApp.jsp`**

SAE URL on IDP end
: **`https://lb2.idp-example.com:1081/opensso/idpsaehandler/metaAlias/idp`**

This application's identity (should match Secret below)
: **`https://sae.idp-example.com:8181/opensso/saml2/sae/saeIDPApp.jsp`**

Crypto Type (symmetric | asymmetric)
: Select symmetric from the drop down menu.

Shared Secret / Private Key alias
: **`secret12`**

Key store path (asymmetric only)
: No value

Key store password (asymmetric only)
: No value

Private Key password (asymmetric only)
: No value

3 **Click Generate URL**

The Secure Attributes Exchange IDP APP SAMPLE is generated and the following links are displayed.

```
Click here to invoke the remote SP App via
http GET to local IDP : https://sae.sp-example.com:8181/
opensso/samples/saml2/sae/saeSPApp.jsp : ssourl

Click here to invoke the remote SP App via
http POST to IDP : https://sae.sp-example.com:8181/
opensso/samples/saml2/sae/saeSPApp.jsp : POST

This URL will invoke global Logout : slourl
```

`ssourl`, `POST`, and `slourl` are clickable.

4 **Click `ssourl`.**

The SAE SP APP SAMPLE page is displayed proving that Secure Attribute Exchange single sign-on has succeeded.

```
SAE SP APP SAMPLE


Secure Attrs :
sun.authlevel    0
sun.spentityid    https://lb4.sp-example.com:1081/opensso
branch    mainbranch
sun.idpentityid    https://lb2.idp-example.com:1081/opensso
mail    testuser@foo.com
```

5 **Enter `https://lb2.idp-example.com:1081/opensso/samples/saml2/sae/saeIDPApp.jsp` in the browser to regenerate the Secure Attributes Exchange IDP APP SAMPLE page.**

The Secure Attributes Exchange IDP APP SAMPLE is regenerated and the following links are displayed.

```
Click here to invoke the remote SP App via
http GET to local IDP : https://sae.sp-example.com:8181/
opensso/samples/saml2/sae/saeSPApp.jsp : ssourl

Click here to invoke the remote SP App via
http POST to IDP : https://sae.sp-example.com:8181/
opensso/samples/saml2/sae/saeSPApp.jsp : POST

This URL will invoke global Logout : slourl
```

`ssourl`, `POST`, and `slourl` are clickable.

6 **Click slourl.**

The Secure Attributes Exchange IDP APP SAMPLE is displayed.

7 **Type the following values in the appropriate text field.**

Userid on local IDP
testuser

Authenticated auth level
0

mail attribute
testuser@foo.com

branch attribute
mainbranch

SP App URL
`https://sae.sp-example.com:8181/opensso/saml2/sae/saeSPApp.jsp`

SAE URL on IDP end
: **`https://lb2.idp-example.com:1081/opensso/idpsaehandler/metaAlias/idp`**

This application's identity (should match Secret below)
: **`https://sae.idp-example.com:8181/opensso/saml2/sae/saeIDPApp.jsp`**

Crypto Type (symmetric | asymmetric)
: symmetric

Shared Secret / Private Key alias
: **`secret12`**

Key store path (asymmetric only)
: No value

Key store password (asymmetric only)
: No value

Private Key password (asymmetric only)
: No value

8 **Click Generate URL.**

The Secure Attributes Exchange IDP APP SAMPLE page is displayed.

```
Secure Attributes Exchange IDP APP SAMPLE

Setting up the following params:
branch=mainbranch
mail=testuser@foo.com
sun.userid=testuser
sun.authlevel=0
sun.spappurl=https://sae.sp-example.com:8181/opensso/
  saml2/sae/saeSPApp.jsp
sun.idpappurl=https://sae.idp-example.com:8181/opensso/
  saml2/sae/saeIDPApp.jsp


Click here to invoke the remote SP App via http GET to local IDP :
  https://sae.sp-example.com:8181/opensso/saml2/sae/saeSPApp.jsp : ssourl

Click here to invoke the remote SP App via http POST to IDP :
  https://sae.sp-example.com:8181/opensso/saml2/sae/saeSPApp.jsp

This URL will invoke global Logout : slourl
```

9 **Click slourl.**

The SAE SP APP SAMPLE page is displayed proving successful logout.

```
SAE SP APP SAMPLE
```

```
Secure Attrs :
sun.cmd    logout
sun.returnurl    https://lb4.sp-example.com:1081/opensso/SPSloRedirect/
metaAlias/sp?SAMLRequest=nZNva9swEMa%2FitHbkliS438iMQTCWErXpvUWxt5
d7HMqsCVPJ0P27WcnLaSDdlDQq5Oe%2Bz33cFoSdG2v7uzRDv4Jfw9IPghOXWtIna9
WbHBGWSBNykCHpHylyvW3OyXnXPXOelvZlgXbzYqRrKPDouKQQpOmnIsMRSMhgSgRIuU
gU55jLEQlWbBHR9qaFRvbjGqiAbeGPBg%2FljjPZjyfyfy7jFSUjOcXCzajNW3An1XP3
vekwrA9zJI5aWdxXtlOCZ6J0PZoiGxY7srWPmGtHVY%2B7NDDutVAIfUsuLf%2BwTy4d
ePR%2FQtcXIDFcgpAna25q0g%2BTgSI0E0eWXHlUc7xBF3fXrlsoFuGV4QX3P3Ycbv5B
C6YlI8DtLrR00z%2FpbOg3L2veS9VFnyxrgP%2Fsa2poutZc36qvANDGo1nhfwqbv78u
O334tGI26MRxzAWu%2F3NDp5%2FvsRxSeASR69KpGlPtqbG0yf2siC5iMe9SzMeJynK
KhVCZsAhr6s6y2OIDg1WUSq4uODfEovX4psPUvwF&RelayState=s212b785d4bda31
faa635552f1233bbbb3a2c5badb&sun.appreturn=true

Logout URL
```

10 **Click Logout URL on the page displayed in the previous step.**

At the bottom of the displayed page, you will see This proves SLO success.

Troubleshooting If there are issues running this test, see the OpenSSO Enterprise debug files located in the `/export/ossoadm/config/opensso/debug/Federation` directory on both the identity provider and the service provider sides.

CHAPTER 14

Testing Attribute Mapping

In this deployment there is no user data on the service provider side so, because of this, we map all identity provider users to an anonymous user which represents all users in the identity provider user data store when it presents itself to the service provider. This use case illustrates how you can pass user profile attributes from the identity provider to the service provider, and from the service provider site to its agent-protected applications. Communication from the identity provider to the service provider takes place using SAML v2 protocols. Communication from the service provider to its agent-protected applications uses agent-to-LDAP attribute mapping. This chapter contains the following sections.

- "14.1 Creating a Test User" on page 307
- "14.2 Configuring OpenSSO Enterprise for Attribute Mapping" on page 309
- "14.3 Testing Attribute Mapping" on page 313

14.1 Creating a Test User

Create a test user and modify the user profile for attribute mapping. Use the following as a checklist to complete this procedure.

1. "To Create a Test User for Attribute Mapping" on page 307
2. "To Edit the Test User Profile" on page 308

▼ To Create a Test User for Attribute Mapping

1 **Access** `https://lb2.idp-example.com:1081/opensso/console` **from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

The Common Tasks tab is displayed.

3 **Click the Access Control tab.**

4 **Click the** `/ (Top Level Realm)` **realm.**

5 **Under the Subjects tab, click User.**

6 **Under User, click** `New`**.**

The New User page is displayed.

7 **Enter the following values and click OK.**

ID	**`jsmith`**
First Name	**`John`**
Last Name	**`Smith`**
Full Name	**`John Smith`**
Password	**`jsmith`**
Password (confirm)	**`jsmith`**.
User Status	Click Active.

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

▼ To Edit the Test User Profile

Before You Begin This procedure assumes you have completed "To Create a Test User for Attribute Mapping" on page 307.

1 **Access** `https://lb2.idp-example.com:1081/opensso/console` **from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

The Common Tasks tab is displayed.

3 **Click the Access Control tab.**

4 **Click the** `/ (Top Level Realm)` **realm.**

5 **Under the Subjects tab, click User.**

6 **Under User, click** `John Smith`**.**

The Edit User — John Smith page is displayed.

7 **Enter the following values and click Save.**

Email Address	**`jsmith@jsmith.com`**
Telephone Number	**`408-555-5454`**

The profile is updated.

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

14.2 Configuring OpenSSO Enterprise for Attribute Mapping

This section contains the instructions to configure OpenSSO Enterprise for attribute mapping. Use the following as a checklist to complete the configurations.

1. "To Add SAML v2 Mappings to the Identity Provider Metadata" on page 309
2. "To Enable Anonymous Authentication" on page 310
3. "To Modify the Agent Profile to Use SAMLv2 Transient " on page 311
4. "To Map Identity Provider User Attributes to Service Provider Anonymous User Attributes" on page 312

▼ To Add SAML v2 Mappings to the Identity Provider Metadata

Map the appropriate LDAP attributes in the user data store to the attributes passed using SAML v2 using the OpenSSO Enterprise console on the identity provider side. When attributes on one OpenSSO Enterprise instance on the identity provider side are mapped, the mapping is made available to the second OpenSSO Enterprise instance on the identity provider side through the previous configuration of the two instances as a site in "5.4 Configuring the OpenSSO Enterprise Platform Service" on page 112

1 **Access** `https://lb2.idp-example.com:1081/opensso/console` **from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

The Common Tasks tab is displayed.

3 **Click the Federation tab.**

4 **Under Entity Providers, click** `https://lb2.idp-example.com:1081/opensso`**.**

The IDP profile page is displayed.

5 **Click the Assertion Processing tab.**

6 **Under Attribute Mapping, enter the following values and click Add.**

```
EmailAddress=EmailAddress
Telephone=Telephone
```

7 **Click Save.**

The profile is updated.

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

▼ To Enable Anonymous Authentication

Enable the Anonymous authentication module and confirm the creation of the anonymous user account on the service provider side.

Before You Begin This procedure assumes you have completed "To Create a Test User for Attribute Mapping" on page 307.

1 **Access** `https://lb4.sp-example.com:1081/opensso/console` **from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

The Common Tasks tab is displayed.

3 **Click the Access Control tab.**

4 **Click the** `/ (Top Level Realm)` **realm.**

5 **Click the Authentication tab.**

6 **Click the Modules Instances link.**

7 **Under Modules Instances, click New.**

The New Module Instance page is displayed.

8 **Enter the following values and click Save.**

Name	**`Anonymous`**
Type	Select Anonymous

The profile is updated.

9 **Under Modules Instances, click Anonymous.**

The Anonymous module instance profile is displayed.

10 **Confirm the default values for the following attributes.**

If the values in your instance are different, change them and save the profile.

Default Anonymous User Name	anonymous
Authentication Level	0

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

▼ To Modify the Agent Profile to Use SAMLv2 Transient

A transient name identifier is a temporary user identifier. In this use case, there is no user account on the service provider side so single sign-on is accomplished using a transient name identifier. All users passed from the identity provider to the service provider will be mapped to the anonymous user created in “To Enable Anonymous Authentication” on page 310. In this procedure, we modify the agent profile to use the transient name identifier format.

1 **Access `https://lb4.sp-example.com:1081/opensso/console` from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

The Common Tasks tab is displayed.

3 **Click the Access Control tab.**

4 **Click the `/ (Top Level Realm)` realm.**

5 **Click the Agents tab.**

6 **Click the Web tab.**

The Web profile page is displayed.

7 **Click `webagent-1` in the Agent table.**

The `webagent-1` profile page is displayed.

8 **Click the OpenSSO Services tab.**

9 **Select `https://lb4.sp-example.com:1081/opensso/spssoinit?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1081/opensso` in the OpenSSO Login URL property box and click Delete.**

10 **Enter `https://lb4.sp-example.com:1081/opensso/spssoinit?metaAlias=/sp&idpEntityID=https://lb2.idp-example.com:1081/opensso&NameIDFormat=transient` in the OpenSSO Login URL text box and click Add.**

11 **Click Save.**

The profile is updated.

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

▼ To Map Identity Provider User Attributes to Service Provider Anonymous User Attributes

Map the attributes being sent from the identity provider to the attributes configured for the anonymous user on the service provider side.

1 **Access `https://lb4.sp-example.com:1081/opensso/console` from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

The Common Tasks tab is displayed.

3 **Click the Access Control tab.**

4 **Click the `/ (Top Level Realm)` realm.**

5 **Click the Agents tab.**

6 **Click the Web tab.**

The Web profile page is displayed.

7 **Click `webagent-1` in the Agent table.**

The `webagent-1` profile page is displayed.

8 **Click the Application tab.**

9 **Click the Session Attribute Processing link.**

10 **Select `HTTP_HEADER` as the value for the Session Attribute Fetch Mode property.**

11 **Enter the following new values in the Session Attribute Map property text box and click Add.**

Map Key	Telephone
Corresponding Map Value	Telephone

12 **Enter the following new values in the Session Attribute Map property text box and click Add.**

Map Key	EmailAddress
Corresponding Map Value	EmailAddress

13 **Click Save.**

The profile is updated.

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

14.3 Testing Attribute Mapping

This test uses `snoop.jsp` to display the HTTP headers being passed in a browser window. Within the headers you see the attributes being passed to the service provider protected by the agent.

▼ To Verify That Attribute Mapping is Working Properly

1 **Log into the `pr1.sp-example.com` host machine as the root user.**

2 **Copy `snoop.jsp` to the `/opt/SUNWwbsvr/https-pr1.sp-example.com/docs` directory.**

`snoop.jsp` is in Appendix F, “The `snoop.jsp` File.”

3 **Access** `http://pr1.sp-example.com:1080/snoop.jsp` **from a web browser.**

The Web Policy Agent redirects the request to the OpenSSO Enterprise console on the identity provider side.

4 **Log in to the OpenSSO Enterprise console as the test user.**

Username **`jsmith@jsmith.com`**

Password **`jsmith`**

JSP Snoop page is the header from the HTTP request in the browser. Note the following:

- John Smith's telephone number and email address are included.
- The Remote user is anonymous and serves as confirmation of the transient user previously configured.

```
JSP Snoop page
Request information
Requested URL: http://pr1.sp-example.com:1080/snoop.jsp
Request method: GET
Request URI: /snoop.jsp
Request protocol: HTTP/1.1
Servlet path: /snoop.jsp
Path info: null
Path translated: null
Query string: null
Content length: -1
Content type: null
Server name: pr1.sp-example.com
Server port: 1080
Remote user: anonymous
Remote address: 192.18.120.83
Remote host: 192.18.120.83
Authorization scheme: DSAME
Request headers
Header: Value:
cookie  JSESSIONID=A7092AD436027D5B18DFCC8C65D7B580;
  iPlanetDirectoryPro=AQIC5wM2LY4SfcxahJE41EKzHCTvKn
  lulj6F8sTjtxvBpA8=@AAJTSQACMDMAAlMxAAIwMQ==#; amlbcookie=01
host     pr1.sp-example.com:1080
user-agent     Mozilla/5.0 (X11; U; SunOS sun4u; en-US;
rv:1.8.1.15) Gecko/20080703 Firefox/2.0.0.15
accept     text/xml,application/xml,application/xhtml+xml,
text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
accept-language     en-us,en;q=0.5
accept-encoding     gzip,deflate
accept-charset     ISO-8859-1,utf-8;q=0.7,*;q=0.7
keep-alive     300
```

```
connection     keep-alive
emailaddress     jsmith@jsmith.com
telephone     408-555-5454
Init parameters
Parameter:     Value:
fork     false
mappedfile     false
logVerbosityLevel     warning
com.sun.appserv.jsp.classpath     /opt/SUNWwbsvr/lib/webserv-rt.jar:
  /opt/SUNWwbsvr/lib/pwc.jar:/opt/SUNWwbsvr/lib/ant.jar:
  /opt/SUNWwbsvr/jdk/lib/tools.jar:/opt/SUNWwbsvr/lib/ktsearch.jar:
  /opt/SUNWwbsvr/lib/webserv-jstl.jar:/opt/SUNWwbsvr/lib/jsf-impl.jar:
  /opt/SUNWwbsvr/lib/jsf-api.jar:/opt/SUNWwbsvr/lib/webserv-jwsdp.jar:
  /opt/SUNWwbsvr/lib/container-auth.jar:/opt/SUNWwbsvr/lib/mail.jar:
  /opt/SUNWwbsvr/lib/activation.jar:
httpMethods     GET,HEAD,POST
```

PART V

Appendices

This final part of *Deployment Example: SAML v2 Using Sun OpenSSO Enterprise 8.0* contains technical configurations and other information regarding this deployment.

Note – The BIG-IP load balancer login page and configuration console for all load balancers in this deployment example is accessed from the URL, `is-f5.example.com`.

Login	username
Password	password

APPENDIX A

Identity Provider Directory Server Host Machines, Load Balancer and Test User

This appendix collects the information regarding the Directory Server instances. It contains the following tables:

- Sun Java System Directory Server 1 Host Machine
- Sun Java System Directory Server 2 Host Machine
- Load Balancer for Directory Server Host Machines
- Test User Entry

TABLE A–1 Sun Java System Directory Server 1 Host Machine

Components	Description	
Host Name	ds1.idp-example.com	
Installation Directory	/var/opt/mps/serverroot/	
Administrator User	cn=Directory Manager	
Administrator Password	dsmanager	
User Data Instance	Instance Name	idp-users
	Instance Directory	/var/opt/mps/idp-users
	Port Number	1489 (LDAP) 1736 (LDAPS)
	Base Suffix	dc=company,dc=com
	Users Suffix	ou=users,dc=company,dc=com
	Administrative User	cn=Directory Manager
	Administrative User Password	dsmanager

TABLE A–1 Sun Java System Directory Server 1 Host Machine *(Continued)*

Components	Description	
	Replication Manager	cn=replication manager,cn=replication,cn=config
	Replication Manager Password	replmanager

TABLE A–2 Sun Java System Directory Server 2 Host Machine

Component	Description	
Host Name	ds2.idp-example.com	
Installation Directory	/var/opt/mps/serverroot/	
Administrator User	cn=Directory Manager	
Administrator Password	dsmanager	
User Data Instance	Instance Name	idp-users
	Instance Directory	/var/opt/mps/idp-users
	Port Number	1489 (LDAP) 1736 (LDAPS)
	Base Suffix	dc=company,dc=com
	Users Suffix	ou=users,dc=company,dc=com
	Administrative User	cn=Directory Manager
	Administrative User Password	dsmanager
	Replication Manager	cn=replication manager,cn=replication,cn=config
	Replication Manager Password	replmanager

TABLE A–3 Load Balancer for Directory Server Host Machines

Component	Description
URL	lb1.idp-example.com
Method	Round Robin
Protected Servers	ds1.idp-example.com:1736
	ds2.idp-example.com:1736
Virtual Servers	lb1.idp-example.com:489
Monitors	ds1.idp-example.com:1736
	ds2.idp-example.com:1736

TABLE A–4 Test User Entry

UserID	Description	
idpuser	Password	idpuser
	DN	uid=idpuser1,ou=users,dc=company,dc=com

APPENDIX B

Service Provider Directory Server Host Machines, Load Balancer and Test User

This appendix collects the information regarding the Directory Server instances. It contains the following tables:

- Sun Java System Directory Server 1 Host Machine
- Sun Java System Directory Server 2 Host Machine
- Load Balancer for Directory Server Host Machines
- Test User Entry

TABLE B–1 Sun Java System Directory Server 1 Host Machine

Components	Description	
Host Name	ds1.sp-example.com	
Installation Directory	/var/opt/mps/serverroot/	
Administrator User	cn=Directory Manager	
Administrator Password	dsmanager	
User Data Instance	Instance Name	sp-users
	Instance Directory	/var/opt/mps/sp-users
	Port Number	1489 (LDAP) 1736 (LDAPS)
	Base Suffix	o=spusers.com
	Users Suffix	ou=users,o=spusers.com
	Administrative User	cn=Directory Manager
	Administrative User Password	dsmanager

TABLE B–1 Sun Java System Directory Server 1 Host Machine *(Continued)*

Components	Description	
	Replication Manager	cn=replication manager,cn=replication,cn=config
	Replication Manager Password	replmanager

TABLE B–2 Sun Java System Directory Server 2 Host Machine

Component	Description	
Host Name	ds2.sp-example.com	
Installation Directory	/var/opt/mps/serverroot/	
Administrator User	cn=Directory Manager	
Administrator Password	dsmanager	
User Data Instance	Instance Name	sp-users
	Instance Directory	/var/opt/mps/sp-users
	Port Number	1489 (LDAP) 1736 (LDAPS)
	Base Suffix	o=spusers.com
	Users Suffix	ou=users,o=spusers.com
	Administrative User	cn=Directory Manager
	Administrative User Password	dsmanager
	Replication Manager	cn=replication manager,cn=replication,cn=config
	Replication Manager Password	replmanager

TABLE B–3 Load Balancer for Directory Server Host Machines

Component	Description
URL	lb3.sp-example.com
Method	Round Robin
Protected Servers	ds1.sp-example.com:1736
	ds2.sp-example.com:1736
Virtual Servers	lb3.sp-example.com:489
Monitors	ds1.sp-example.com:1736
	ds2.sp-example.com:1736

TABLE B–4 Test User Entry

UserID	Description	
spuser	Password	spuser
	DN	uid=spuser1,ou=users,o=spusers.com

APPENDIX C

Identity Provider OpenSSO Enterprise Host Machines and Load Balancers

This appendix collects the information regarding the identity provider OpenSSO Enterprise host machines.

- OpenSSO Enterprise 1 Host Machine
- OpenSSO Enterprise 2 Host Machine
- Load Balancer for OpenSSO Enterprise Host Machines

TABLE C–1 OpenSSO Enterprise 1 Host Machine

Component	Description	
Host Name	osso1.idp-example.com	
Non-Root User	osso80adm	
Non-Root User Password	nonroot1pwd	
Sun Java System Application Server Administrative Server	Installation Directory	/opt/SUNWappserver91
	Administrative User	admin
	Administrative User Password	domain1pwd
	Ports	4848 (administration) 8080 (HTTP) 8181 (HTTPS)
	Default Domain Name	domain1
	Administrative Console URL	http://osso1.idp-example.com:4848

TABLE C–1 OpenSSO Enterprise 1 Host Machine *(Continued)*

Component	Description	
Sun Java System Application Server Non-Root User Domain	Name	ossodomain
	Directory	/export/osso80adm/domains/
	Administrative User	domain2adm
	Administrative User Password	domain2pwd
	Master Password	domain2master
	Ports	8989 (administration) 1080 (HTTP) 1081 (HTTPS)
	Administrative Console URL	http://osso2.idp-example.com:8989
OpenSSO Enterprise	Administrative User	amadmin
	Administrative User Password	ossoadmin
	Configuration Data Store	Embedded
	User Data Store	lb2.idp-example.com:489
	Agent User	agentuser
	Agent User Password	agentuser
	Administrative Console URL	https://osso2.idp-example.com:1081/opensso/console

TABLE C–2 OpenSSO Enterprise 2 Host Machine

Component	Description	
Host Name	osso2.idp-example.com	
Non-Root User	osso80adm	
Non-Root User Password	nonroot2pwd	
Sun Java System Application Server Administrative Server	Installation Directory	/opt/SUNWappserver91
	Administrative User	admin
	Administrative User Password	domain1pwd
	Ports	4848 (administration) 8080 (HTTP) 8181 (HTTPS)
	Default Domain Name	domain1
	Administrative Console URL	http://osso2.idp-example.com:4848
Sun Java System Application Server Non-Root User Domain	Name	ossodomain
	Directory	/export/osso80adm/domains/
	Administrative User	domain2adm
	Administrative User Password	domain2pwd
	Master Password	domain2master
	Ports	8989 (administration) 1080 (HTTP) 1081 (HTTPS)
	Administrative Console URL	http://osso2.idp-example.com:8989
OpenSSO Enterprise	Administrative User	amadmin
	Administrative User Password	ossoadmin

TABLE C–2 OpenSSO Enterprise 2 Host Machine *(Continued)*

Component	Description	
	Configuration Data Store	Embedded
	User Data Store	lb2.idp-example.com:489
	Agent User	agentuser
	Agent User Password	agentuser
	Administrative Console URL	https://osso2.idp-example.com:1081/opensso/console

TABLE C–3 Load Balancer for OpenSSO Enterprise Host Machines

Component	Description
URL	lb2.idp-.example.com
Method	Round Robin
Protected Servers	osso1.idp-example.com:1081 osso2.idp-example.com:1081
Virtual Servers	lb2.idp-example.com:489
Monitors	osso1.idp-example.com:1081 osso2.idp-example.com:1081
Cookie Name	amlbcookie

APPENDIX D

Service Provider OpenSSO Enterprise Host Machines and Load Balancers

This appendix collects the information regarding the service provider OpenSSO Enterprise host machines.

- OpenSSO Enterprise 1 Host Machine
- OpenSSO Enterprise 2 Host Machine
- Load Balancer for OpenSSO Enterprise Host Machines

TABLE D–1 OpenSSO Enterprise 1 Host Machine

Component	Description	
Host Name	osso1.sp-example.com	
Non-Root User	osso80adm	
Non-Root User Password	nonroot1pwd	
Sun Java System Application Server Administrative Server	Installation Directory	/opt/SUNWappserver91
	Administrative User	admin
	Administrative User Password	domain1pwd
	Ports	4848 (administration) 8080 (HTTP) 8181 (HTTPS)
	Default Domain Name	domain1
	Administrative Console URL	http://osso1.sp-example.com:4848

TABLE D–1 OpenSSO Enterprise 1 Host Machine *(Continued)*

Component	Description	
Sun Java System Application Server Non-Root User Domain	Name	ossodomain
	Directory	/export/osso80adm/domains/
	Administrative User	domain2adm
	Administrative User Password	domain2pwd
	Master Password	domain2master
	Ports	8989 (administration) 1080 (HTTP) 1081 (HTTPS)
	Administrative Console URL	http://osso2.sp-example.com:8989
OpenSSO Enterprise	Administrative User	amadmin
	Administrative User Password	ossoadmin
	Configuration Data Store	Embedded
	User Data Store	lb2.isp-example.com:489
	Agent User	agentuser
	Agent User Password	agentuser
	Administrative Console URL	https://osso2.sp-example.com:1081/opensso/console

TABLE D–2 OpenSSO Enterprise 2 Host Machine

Component	Description	
Host Name	osso2.sp-example.com	
Non-Root User	osso80adm	
Non-Root User Password	nonroot2pwd	
Sun Java System Application Server Administrative Server	Installation Directory	/opt/SUNWappserver91
	Administrative User	admin
	Administrative User Password	domain1pwd
	Ports	4848 (administration) 8080 (HTTP) 8181 (HTTPS)
	Default Domain Name	domain1
	Administrative Console URL	http://osso2.sp-example.com:4848
Sun Java System Application Server Non-Root User Domain	Name	ossodomain
	Directory	/export/osso80adm/domains/
	Administrative User	domain2adm
	Administrative User Password	domain2pwd
	Master Password	domain2master
	Ports	8989 (administration) 1080 (HTTP) 1081 (HTTPS)
	Administrative Console URL	http://osso2.sp-example.com:8989
OpenSSO Enterprise	Administrative User	amadmin
	Administrative User Password	ossoadmin

TABLE D–2 OpenSSO Enterprise 2 Host Machine *(Continued)*

Component	Description	
	Configuration Data Store	Embedded
	User Data Store	lb2.sp-example.com:489
	Agent User	agentuser
	Agent User Password	agentuser
	Administrative Console URL	https://osso2.sp-example.com:1081/opensso/console

TABLE D–3 Load Balancer for OpenSSO Enterprise Host Machines

Component	Description
URL	lb4.sp-.example.com
Method	Round Robin
Protected Servers	osso1.sp-example.com:1081
	osso2.sp-example.com:1081
Virtual Servers	lb2.sp-example.com:489
Monitors	osso1.sp-example.com:1081
	osso2.sp-example.com:1081
Cookie Name	amlbcookie

APPENDIX E

Service Provider Protected Resource Host Machine Web Containers and Policy Agents

This appendix collects the information regarding the web containers and policy agents installed on the Protected Resource host machine.

TABLE E–1 Protected Resource 1 Host Machine

Component	Description	
Host Name	pr1.sp-example.com	
BEA WebLogic Server Administration Server	Home Directory	/usr/local/bea
	Installation Directory	/usr/local/bea/weblogic10
	Domain Directory	/usr/local/bea/user_projects/domains/pr1
	Administration Server Directory	/usr/local/bea/user_projects/domains/pr1/servers/AdminServer
	Administrator	weblogic
	Administrator Password	bea10admin
	Port	7001
	Administration Console URL	http://pr1.sp-example.com:7001/console
BEA WebLogic Server Managed Server	Managed Server Directory	/usr/local/bea/user_projects/domains/pr1/servers/ApplicationServer-1
	Port	1081
	OpenSSO Enterprise URL	https://lb4.sp-example.com:1081/opensso
J2EE Policy Agent for BEA WebLogic Server	J2EE Agent Profile Name	j2eeagent–1

TABLE E–1 Protected Resource 1 Host Machine *(Continued)*

Component	Description	
	J2EE Agent Profile Password	j2eeagent1
	J2EE Agent URL	http://pr1.sp-example.com:1081/agentapp
Sun Java System Web Server Administration Server	Installation Directory	/opt/SUNWwbsvr/
	Default Administration Directory	/opt/SUNWwbsvr/admin-server
	Default Administrator	admin
	Default Administrator Password	web4dmin
	Runtime User ID	root
	Ports	8989 (SSL) 1080 (HTTP)
Sun Java System Web Server Instance	Instance Name	pr1.sp-example.com
	Instance Directory	/opt/SUNWwbsvr/https-pr-1.example.com
	Port	1080
	Service URL	http://pr1.sp-example.com:1080
Web Policy Agent for Sun Java System Web Server	Web Agent Profile Name	webagent-1
	Web Agent Profile Password	webagent1

APPENDIX F

The snoop.jsp File

This appendix contains the snoop.jsp file used in .

```
<HTML>
<HEAD>
<TITLE>JSP snoop page</TITLE>
<%@ page import="javax.servlet.http.
HttpUtils,java.util.Enumeration" %>
</HEAD>
<BODY>
<H1>JSP Snoop page</H1>
FIGURE 16?1 Output from snoop.jsp
Example 16?1
16.1 Mapping User Attributes fromthe Identity Provider to
a Single User on the Service Provider
284 Deployment Example 2: Federation Using SAML v2 ? April 2007
<H2>Request information</H2>
<TABLE>
<TR>
<TH align=right>Requested URL:</TH>
<TD><%= HttpUtils.getRequestURL(request) %></TD>
</TR>
<TR>
<TH align=right>Request method:</TH>
<TD><%= request.getMethod() %></TD>
</TR>
<TR>
<TH align=right>Request URI:</TH>
<TD><%= request.getRequestURI() %></TD>
</TR>
<TR>
<TH align=right>Request protocol:</TH>
<TD><%= request.getProtocol() %></TD>
</TR>
```

```
<TR>
<TH align=right>Servlet path:</TH>
<TD><%= request.getServletPath() %></TD>
</TR>
<TR>
<TH align=right>Path info:</TH>
<TD><%= request.getPathInfo() %></TD>
</TR>
<TR>
<TH align=right>Path translated:</TH>
<TD><%= request.getPathTranslated() %></TD>
</TR>
<TR>
<TH align=right>Query string:</TH>
<TD><%= request.getQueryString() %></TD>
</TR>
<TR>
<TH align=right>Content length:</TH>
<TD><%= request.getContentLength() %></TD>
</TR>
<TR>
<TH align=right>Content type:</TH>
<TD><%= request.getContentType() %></TD>
<TR>
<TR>
<TH align=right>Server name:</TH>
<TD><%= request.getServerName() %></TD>
16.1 Mapping User Attributes fromthe Identity Provider
to a Single User on the Service Provider
Chapter 16 ? Use Case 2: User AttributeMapping 285
<TR>
<TR>
<TH align=right>Server port:</TH>
<TD><%= request.getServerPort() %></TD>
<TR>
<TR>
<TH align=right>Remote user:</TH>
<TD><%= request.getRemoteUser() %></TD>
<TR>
<TR>
<TH align=right>Remote address:</TH>
<TD><%= request.getRemoteAddr() %></TD>
<TR>
<TR>
<TH align=right>Remote host:</TH>
<TD><%= request.getRemoteHost() %></TD>
<TR>
<TR>
```

```
<TH align=right>Authorization scheme:</TH>
<TD><%= request.getAuthType() %></TD>
<TR>
</TABLE>
<%
Enumeration e = request.getHeaderNames();
if(e != null && e.hasMoreElements()) {
%>
<H2>Request headers</H2>
<TABLE>
<TR>
<TH align=left>Header:</TH>
<TH align=left>Value:</TH>
</TR>
<%
while(e.hasMoreElements()) {
String k = (String) e.nextElement();
%>
<TR>
<TD><%= k %></TD>
<TD><%= request.getHeader(k) %></TD>
</TR>
<%
}
%>
</TABLE>
<%
16.1 Mapping User Attributes fromthe Identity Provider
to a Single User on the Service Provider
286 Deployment Example 2: Federation Using SAML v2 ? April 2007
}
%>
<%
e = request.getParameterNames();
if(e != null && e.hasMoreElements()) {
%>
<H2>Request parameters</H2>
<TABLE>
<TR valign=top>
<TH align=left>Parameter:</TH>
<TH align=left>Value:</TH>
<TH align=left>Multiple values:</TH>
</TR>
<%
while(e.hasMoreElements()) {
String k = (String) e.nextElement();
String val = request.getParameter(k);
String vals[] = request.getParameterValues(k);
%>
<TR valign=top>
<TD><%= k %></TD>
<TD><%= val %></TD>
<TD><%
for(int i = 0; i < vals.length; i++) {
if(i > 0)
out.print("<BR>");
out.print(vals[i]);
}
%></TD>
</TR>
<%
}
%>
</TABLE>
<%
}
%>
<%
e = getServletConfig().getInitParameterNames();
if(e != null && e.hasMoreElements()) {
%>
<H2>Init parameters</H2>
<TABLE>
<TR valign=top>
16.1 Mapping User Attributes fromthe Identity Provider
to a Single User on the Service Provider
Chapter 16 ? Use Case 2: User AttributeMapping 287
<TH align=left>Parameter:</TH>
<TH align=left>Value:</TH>
</TR>
<%
while(e.hasMoreElements()) {
String k = (String) e.nextElement();
String val = getServletConfig().getInitParameter(k);
%>
<TR valign=top>
<TD><%= k %></TD>
<TD><%= val %></TD>
</TR>
<%
}
%>
</TABLE>
<%
}
%>
```

```
</BODY>
</HTML>
```

APPENDIX G

Known Issues and Limitations

The issues in this appendix will be updated as more information becomes available.

TABLE G–1 Known Issues and Limitations

Reference Number	Description
4510	**Creating a non-root domain Shows a FileNotFoundException** For more information, see Issue 4510 on `https://glassfish.dev.java.net/`.