Composed October 31, 2008

Deployment Example: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0

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

Part No: 820–5985
November 2008

Composed October 31, 2008

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc., ou ses filiales, aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

081031@21288

Contents

Preface 9

Part I About This Deployment 15

1 Components and Features 17
1.1 Deployment Architecture and Components 17
1.2 Key Features of Deployment 21
1.3 Sequential Component Interactions 21

2 Technical Overview 29
2.1 Host Machines 29
2.2 Software 30
2.3 Main Service URLs 30
2.4 Intercomponent Communication 32
2.5 Firewall Rules 34
2.6 Viewing Replicated Entries 35

3 Before You Begin 37
3.1 Technical Reference 37
3.2 Setting Up the Load Balancers 37
3.3 Obtaining Secure Socket Layer Certificates 38
3.4 Resolving Host Names 38
3.5 Known Issues and Limitations 39

Part II Building the Environment 41

4 Installing Sun Java System Directory Server and Creating Instances for Sun OpenSSO Enterprise User Data 43

4.1 Installing and Configuring Directory Server 1 and Directory Server 2 43

▼ To Download the Directory Server Bits and Required Patches to the Directory Server Host Machines 44

▼ To Patch the Directory Server Host Machines 46

▼ To Install Directory Server 1 47

▼ To Create an OpenSSO Enterprise User Data Instance on Directory Server 1 48

▼ To Create a Base Suffix for the User Data Instance on Directory Server 1 49

▼ To Install Directory Server 2 50

▼ To Create an OpenSSO Enterprise User Data Instance on Directory Server 2 51

▼ To Create a Base Suffix for the User Data Instance on Directory Server 2 52

4.2 Enabling Multi-Master Replication of the User Data Instances 53

▼ To Enable Multi-Master Replication for User Data Instance on Directory Server 1 54

▼ To Enable Multi-Master Replication for User Data Instance on Directory Server 2 55

▼ To Change the Default Replication Manager Password for Each User Data Instance 56

▼ To Create Replication Agreements for Each User Data Instance 57

▼ To Initialize the Replication Agreements 58

▼ To Verify Successful User Data Replication 59

4.3 Enabling Secure Communication for the Directory Server User Data Instances 61

▼ To Install a Root Certificate and a Server Certificate on Directory Server 1 61

▼ To Install a Root Certificate and a Server Certificate on Directory Server 2 63

4.4 Configuring Load Balancer 1 for the User Data Instances 64

▼ To Request a Certificate for the User Data Load Balancer 65

▼ To Import the Root Certificate to the User Data Load Balancer 66

▼ To Install the Server Certificate to the User Data Load Balancer 67

▼ To Configure the User Data Load Balancer 1 68

▼ To Create an SSL Proxy for SSL Termination at the User Data Load Balancer 1 72

4.5 Importing Test Users 74

▼ To Import Test User Data into the Replicated Directory Server Instances 74

5 Deploying and Configuring OpenSSO Enterprise 79

5.1 Installing the Application Server Web Containers 79

▼ To Create a Non-Root User on the OpenSSO Enterprise 1 Host Machine 80
▼ To Install Application Server on the OpenSSO Enterprise 1 Host Machine 80
▼ To Create a Non-Root User on the OpenSSO Enterprise 2 Host Machine 90
▼ To Install Application Server on the OpenSSO Enterprise 2 Host Machine 91
5.2 Configuring Load Balancer 2 for OpenSSO Enterprise 100
▼ To Request a Certificate for the OpenSSO Enterprise Load Balancer 101
▼ To Install a CA Root Certificate to the OpenSSO Enterprise Load Balancer 102
▼ To Install the Server Certificate to the OpenSSO Enterprise Load Balancer 103
▼ To Configure the OpenSSO Enterprise Load Balancer 103
▼ To Create an SSL Proxy for SSL Termination at the OpenSSO Enterprise Load Balancer 106
5.3 Deploying and Configuring OpenSSO Enterprise 1 and OpenSSO Enterprise 2 108
▼ To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine 108
▼ To Deploy the OpenSSO Enterprise WAR as OpenSSO Enterprise 1 110
▼ To Copy the OpenSSO Enterprise WAR to the OpenSSO Enterprise 2 Host Machine 112
▼ To Deploy the OpenSSO Enterprise WAR File as OpenSSO Enterprise 2 113
▼ To Configure OpenSSO Enterprise 1 114
▼ To Configure OpenSSO Enterprise 2 116
5.4 Configuring the OpenSSO Enterprise Platform Service 117
▼ To Create a Site on OpenSSO Enterprise 1 118
▼ To Verify that the OpenSSO Enterprise Site was Configured Properly 120

6 Configuring OpenSSO Enterprise Realms for User Authentication 121
6.1 Modifying the Top-Level Realm for Test Users 121
▼ To Modify the Top-Level Realm for User Authentication 122
▼ To Verify that a User Can Successfully Authenticate 123
6.2 Creating and Configuring a Sub Realm for Test Users 123
▼ To Create a Sub Realm 124
▼ To Change the User Profile Configuration for the Sub Realm 124
▼ To Modify the Sub Realm for User Authentication 125
▼ To Verify That the Sub Realm Can Access the External User Data Store 126
▼ To Verify That the Sub Realm Subjects Can Successfully Authenticate 127

7 Installing and Configuring the Distributed Authentication User Interface 129
7.1 Installing the Distributed Authentication User Interface Web Containers 129

▼ To Create a Non-Root User on the Distributed Authentication User Interface 1 Host Machine 130
▼ To Install the Web Server for Distributed Authentication User Interface 1 130
▼ To Create a Non-Root User on the Distributed Authentication User Interface 2 Host Machine 136
▼ To Install Sun Java System Web Server for Distributed Authentication User Interface 2 . 137
7.2 Enabling Secure Communications Between the Web Server Instances and the Load Balancer 142
▼ To Request and Install a Server Certificate and a Root Certificate for Web Server 1 142
▼ To Create an SSL Enabled HTTP Listener Port on Web Server 1 144
▼ To Request and Install a Server Certificate and a Root Certificate for Web Server 2 147
▼ To Create an SSL Enabled HTTP Listener Port on Web Server 2 149
▼ To Import the Root Certificate to the Web Server 1 JDK Certificate Store 151
▼ To Import the Root Certificate to the Web Server 2 JDK Certificate Store 153
7.3 Configuring the Distributed Authentication User Interface Load Balancer 154
▼ To Request a Certificate for the Distributed Authentication User Interface Load Balancer 155
▼ To Import a Root Certificate to the Distributed Authentication User Interface Load Balancer 156
▼ To Import a Certificate to the Distributed Authentication User Interface Load Balancer 157
▼ To Configure the Distributed Authentication User Interface Load Balancer 158
▼ To Configure a Proxy for SSL Termination at the Distributed Authentication User Interface Load Balancer 160
7.4 Creating an Agent Profile with Custom User for the Distributed Authentication User Interface 162
▼ To Create an Agent Profile with Custom User for the Distributed Authentication User Interface 162
▼ To Verify that `authuiadmin` Was Created in Directory Server 163
7.5 Generating and Deploying the Distributed Authentication User Interface WAR 164
▼ To Generate the Distributed Authentication User Interface WAR 164
▼ To Deploy the Generated WAR as Distributed Authentication User Interface 1 165
▼ To Configure Distributed Authentication User Interface 1 168
▼ To Deploy the Generated WAR as Distributed Authentication User Interface 2 169
▼ To Configure Distributed Authentication User Interface 2 171
▼ To Configure Load Balancer Cookies for the Distributed Authentication User Interface 173
▼ To Verify That Authentication Using the Distributed Authentication User Interface Load Balancer is Successful 175

8 Configuring the Protected Resource Host Machines 177

8.1 Configuring the Protected Resource Host Machines with a J2EE Policy Agent 177

8.1.1 Installing and Configuring the J2EE Container and J2EE Policy Agent on Protected Resource 1 178

8.1.2 Installing and Configuring the J2EE Container and J2EE Policy Agent on Protected Resource 2 194

8.1.3 Creating Groups Using the OpenSSO Enterprise Console 210

8.1.4 Setting Up a Test for the J2EE Policy Agent 1 212

8.1.5 Setting Up a Test for the J2EE Policy Agent 2 217

8.1.6 Configuring the J2EE Policy Agents to Access the Distributed Authentication User Interface 223

8.2 Configuring the Protected Resource Host Machines with a Web Policy Agent 226

8.2.1 Installing and Configuring the Web Container and Web Policy Agent on Protected Resource 1 226

8.2.2 Installing Web Server and a Web Policy Agent on Protected Resource 2 238

8.2.3 Configuring the Web Policy Agents to Access the Distributed Authentication User Interface 250

9 Setting Up Load Balancers for the Policy Agents 253

9.1 Configuring the Web Policy Agents Load Balancer 253

▼ To Configure the Web Policy Agents Load Balancer 254

▼ To Create a Monitoring File on Each Host Machine for Load Balancer 4 256

▼ To Add Load Balancer 4 as a Virtual Host by Modifying the Web Policy Agent Properties 258

▼ To Configure Policy for the Web Policy Agents 260

▼ To Verify the Web Policy Agents Load Balancer Configuration is Working Properly 261

9.2 Configuring the J2EE Policy Agents Load Balancer 262

▼ To Configure the J2EE Policy Agents Load Balancer 262

▼ To Add Load Balancer 5 as a Virtual Host by Modifying the J2EE Policy Agent Properties 264

▼ To Configure Policy for the J2EE Policy Agents 265

▼ To Verify the J2EE Policy Agent Load Balancer Configuration is Working Properly 267

10 Implementing Session Failover 269

10.1 Session Failover Architecture 269

10.2 Installing the Session Failover Components 270

▼ To Install Session Failover Components on Message Queue 1 270
▼ To Install Session Failover Components on Message Queue 2 274
10.3 Configuring and Verifying Session Failover 278
▼ To Configure OpenSSO Enterprise for Session Failover 278
▼ To Verify That the Administrator Session Fails Over 280
▼ To Verify that the User Session Fails Over 281

Part III Reference: Summaries of Server and Component Configurations 283

A Directory Server Host Machines, Test Users and Load Balancer 285

B OpenSSO Enterprise Host Machines and Load Balancer 289

C OpenSSO Enterprise Distributed Authentication User Interface Host Machines and Load Balancer 293

D Protected Resource Host Machine Web Containers, Policy Agents and Load Balancers 297

E Message Queue Servers 303

F Known Issues and Limitations 305

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 *Deployment Example: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0* contains the following sections:

- "About This Guide" on page 9
- "Before You Read This Book" on page 9
- "Related Documentation" on page 10
- "Searching Sun Product Documentation" on page 12
- "Typographical Conventions" on page 13

About This Guide

Deployment Example: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0 provides instructions for building an OpenSSO solution for authentication, authorization and access control. The procedures in this guide were used to build, deploy and test this deployment in a lab facility. 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:

- 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 10
- "Adjunct Product Documentation" on page 11

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* (this guide) 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* 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 `famadm` 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.

Adjunct Product Documentation

Useful information can be found in the documentation for the following products:

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: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0*, and the part number is 820–5985.

Default Paths and Directory Names

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

TABLE P–2 Default Paths and Directory Names

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

Typographical Conventions

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

TABLE P–3 Typographic Conventions

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

TABLE P–3 Typographic Conventions *(Continued)*

Typeface	Meaning	Example
`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.

PART I

About This Deployment

This first part of *Deployment Example: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0* provides introductory material and an overview of the deployment solution. 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: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0 includes procedures for installing, deploying and configuring a number of host machines and applications. This chapter contains introductory information on the deployment example and includes the following sections:

- "1.1 Deployment Architecture and Components" on page 17
- "1.2 Key Features of Deployment" on page 21
- "1.3 Sequential Component Interactions" on page 21

1.1 Deployment Architecture and Components

The following graphic illustrates the deployment architecture — where the components will be situated when the deployment is complete. A list of the components that comprise the architecture follows.

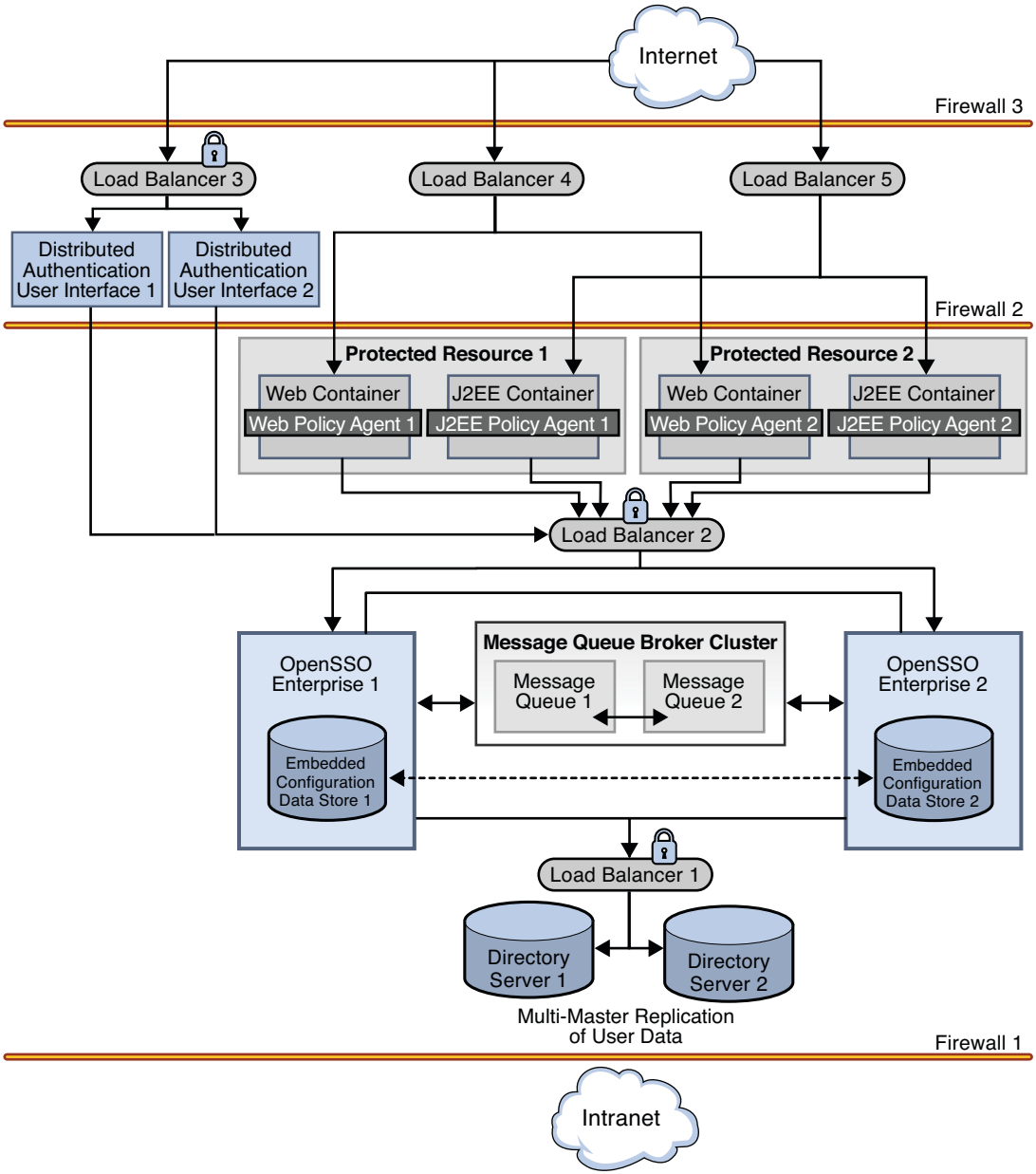


FIGURE 1–1 Deployment Architecture

Note – Although referred to in the illustration, firewalls are not used in this deployment. For general information on integrating firewalls into this deployment, see "2.5 Firewall Rules" on page 34.

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

Sun OpenSSO Enterprise

Two instances of OpenSSO Enterprise provide the core functionality. Each instance is configured with its own embedded configuration data store. Configuration data includes information about services, administrative users, realms, policies, and more. User data is accessed through a single load balancer deployed in front of two instances of Sun Java System Directory Server.

Distributed Authentication User Interface

The Distributed Authentication User Interface is a component of OpenSSO Enterprise that provides a thin presentation layer for user authentication. During user authentication, the Distributed Authentication User Interface interacts with OpenSSO Enterprise to retrieve credentials from the user data store, thus protecting the OpenSSO Enterprise servers from direct user access.

Note – The Distributed Authentication User Interface does not directly interact with the user data store.

Sun Java System Directory Server

Two instances of Directory Server provide storage for the OpenSSO Enterprise user data. User entries will be created for testing this deployment. Both instances of Directory Server are masters that engage in multi-master replication. Multi-master replication allows data to be synchronized in real time between two directories, providing high availability to the OpenSSO Enterprise layer.

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

Sun OpenSSO Enterprise Policy Agents 3.0

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 one of two configured load balancers.

Protected Resource Host Machines

The protected resources host machines contain the content for which access is restricted. Towards this end, web servers, application servers and policy agents will be installed. Two load balancers are configured in front of the host machines to balance traffic passing through the policy agents.

Sun Java System Message Queue

OpenSSO Enterprise uses two instances of Message Queue to form a cluster for distributing client connections and delivering messages. The Berkeley Database by Sleepycat Software, Inc. is the session store database. When an instance of OpenSSO Enterprise goes down and session failover is enabled, the user's session token can be retrieved from one of the Message Queues by the available instance of OpenSSO Enterprise. This ensures that the user remains continuously authenticated, allowing access to the protected resources without having to reauthenticate.

Load Balancers

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

Distributed Authentication User Interface Load Balancer. This external-facing load balancer exposes the remote, web-based Distributed Authentication User Interface for user authentication and self-registration.

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

J2EE Policy Agents Load Balancer. The load balancer in front of the J2EE policy agents installed on the Protected Resource machines provides round-robin load balancing and a single virtual server by balancing traffic passing through the agents.

Web Policy Agents Load Balancer. The load balancer in front of the web policy agents installed on the Protected Resource machines provides round-robin load balancing and a single virtual server by balancing traffic passing through the agents.

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 for the instances of OpenSSO Enterprise. It detects individual Directory Server failures and recoveries, taking failed servers off the load balancer list.

1.2 Key Features of Deployment

- All components (including installations of OpenSSO Enterprise and Directory Server, the Distributed Authentication User Interface, and policy agents) are redundant to achieve high availability.
- All components use ZIP-based installation.
- All components use load-balancing for session failover and high performance.
- Each instance of OpenSSO Enterprise is installed with an embedded configuration data store.
- Each instance of Directory Server contains `am-users` to serve as the user data store.
- OpenSSO Enterprise instances are configured to run as non-root users.
- The environment is configured for system failover capability, ensuring that when one instance of OpenSSO Enterprise goes down, requests are redirected to the second instance.

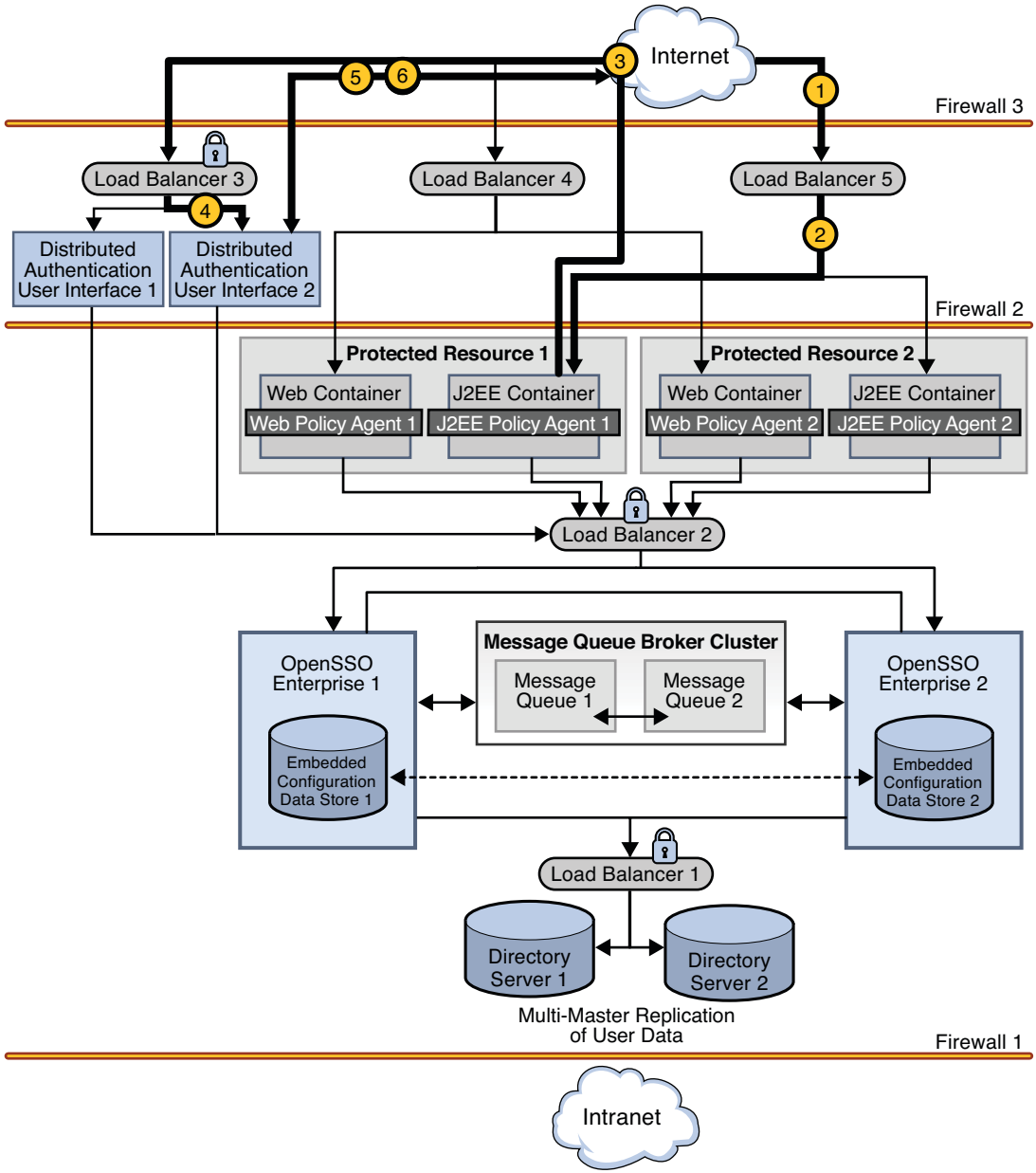
Caution – It is important to note that system failover, by itself, does not ensure OpenSSO Enterprise session failover which is configured separately.

- The environment is configured for session failover capability. Session failover ensures that when the instance of OpenSSO Enterprise *where the user's session was created* goes down, the user's session token can still be retrieved from a backend session database. Thus, the user is continuously authenticated, and does not have to log into the system again unless the session is invalidated as a result of logout or session expiration.
- Communications to the OpenSSO Enterprise load balancer, to the Distributed Authentication User Interface load balancer, and to the Directory Server load balancer are in Secure Sockets Layer (SSL).
- Policy agents are configured with a unique agent profile to authenticate to OpenSSO Enterprise.
- The Distributed Authentication User Interface uses a custom user profile to authenticate to OpenSSO Enterprise instead of the default `amadmin` or `UrlAccessAgent`.

1.3 Sequential Component Interactions

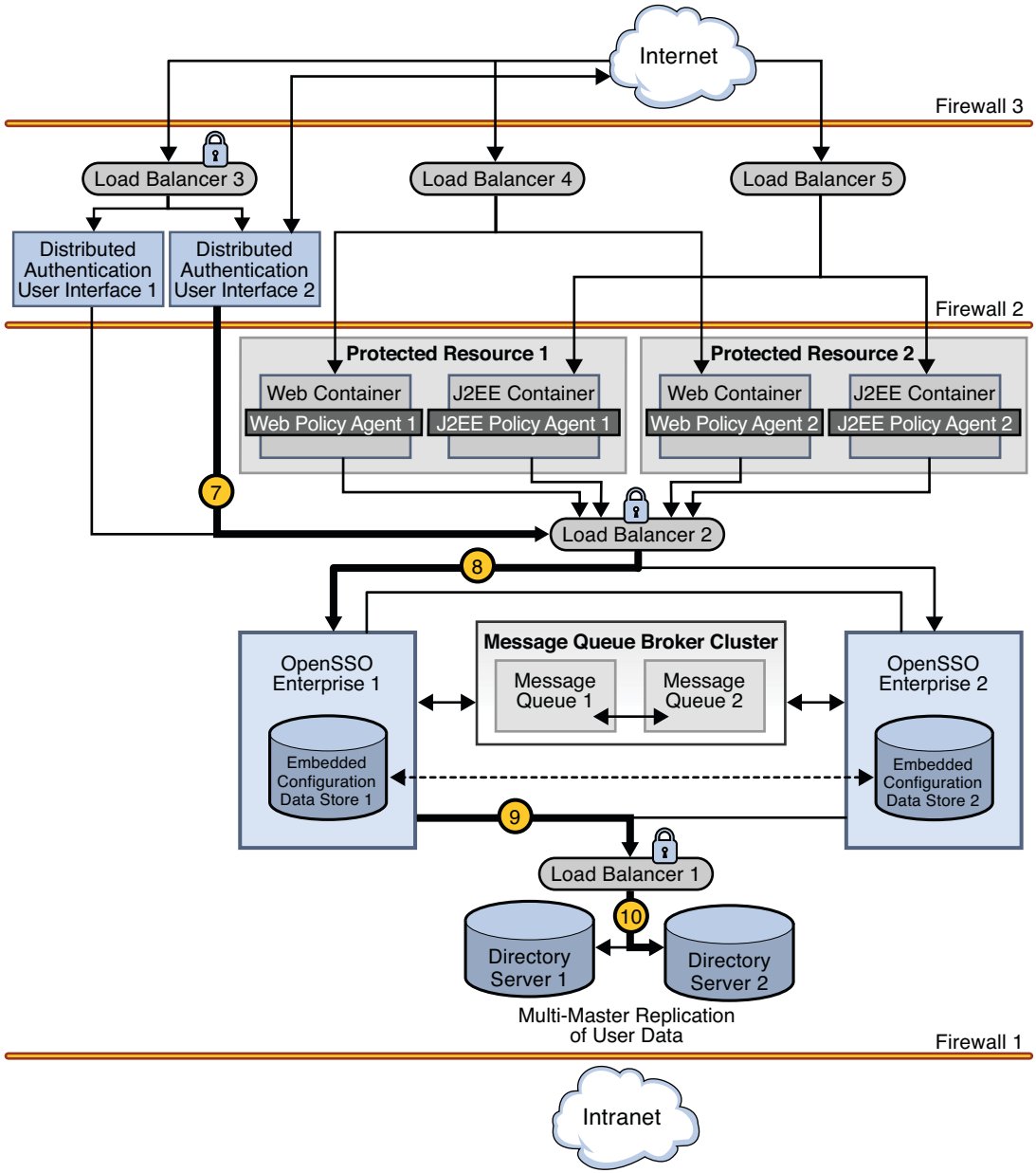
The following sequence describes the interactions between the various components in this deployment. The interactions are illustrated and the numbered steps correspond to the numbers in the diagrams.

1. A user attempts to access a J2EE application hosted on both Protected Resource 1 and Protected Resource 2.



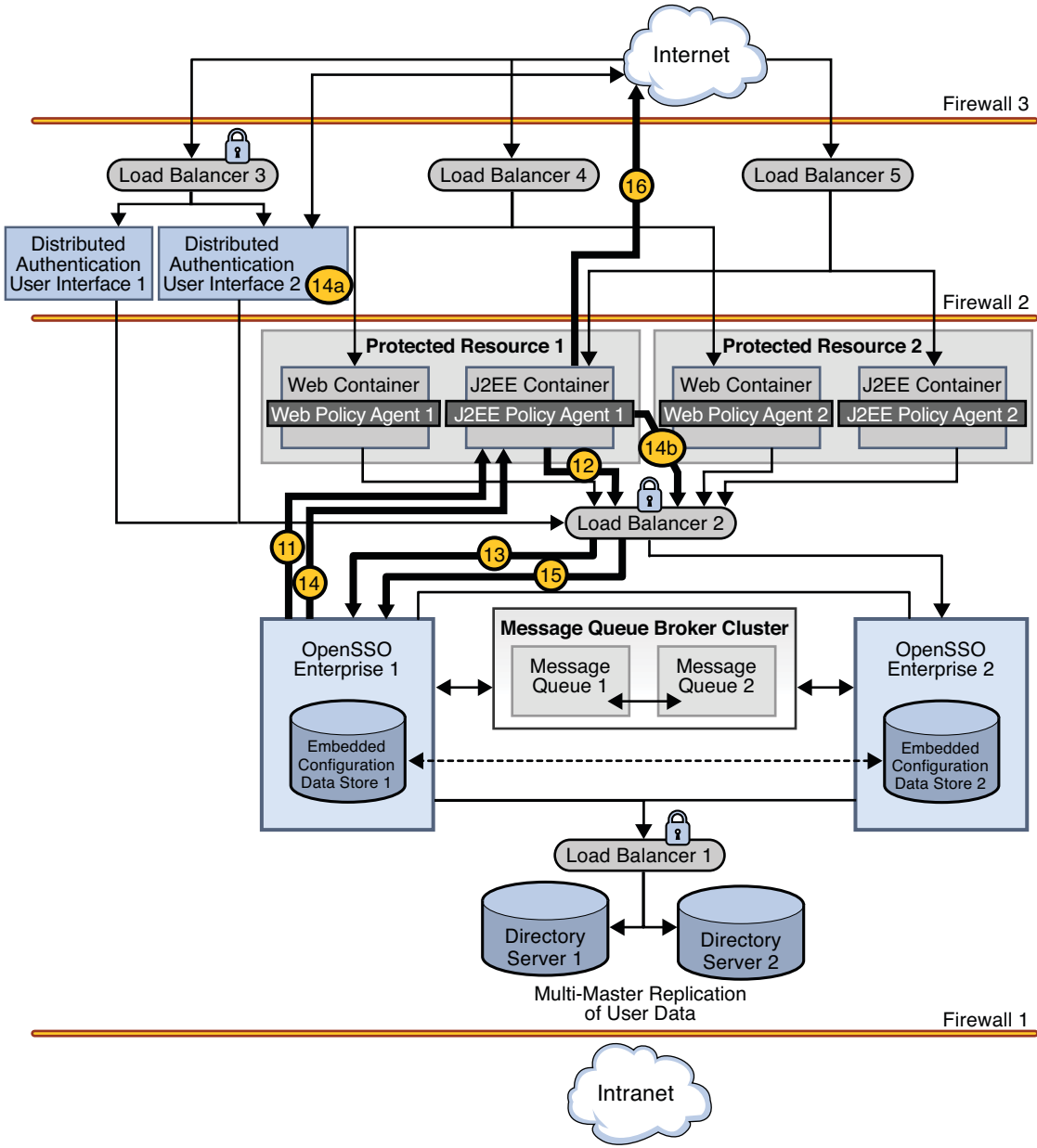
2. Load Balancer 5 directs the user to Protected Resource 1 where J2EE Policy Agent 1 intercepts the request.

3. J2EE Policy Agent 1 checks for an OpenSSO Enterprise cookie (`SSOToken`). In this scenario, no cookie is found and the request is returned to the browser which redirects the request to Load Balancer 3, the load balancer for the instances of the Distributed Authentication User Interface.
4. Load Balancer 3 routes the user request to Distributed Authentication User Interface 2.
5. Distributed Authentication User Interface 2 displays a login page to the user.
6. The user enters credentials on the login page which are returned to Distributed Authentication User Interface 2.
7. Distributed Authentication User Interface 2 passes the credentials to Load Balancer 2.



8. Load Balancer 2 routes the credentials to OpenSSO Enterprise 1.
9. OpenSSO Enterprise 1 sends a request for validation of the credentials to Load Balancer 1 in front of the Directory Server instances.

10. Load Balancer 1 routes the request to Directory Server 2.
11. Authentication occurs using the Distributed Authentication User Interface. Assuming successful authentication, OpenSSO Enterprise Distributed Authentication User Interface 1 sends the response back to J2EE Policy Agent 1 which receives the request and checks again for the OpenSSO Enterprise cookie.



12. When a cookie is found, J2EE Policy Agent 1 sends a session validation request to the OpenSSO Enterprise Load Balancer 2.
13. The OpenSSO Enterprise Load Balancer 2 forwards the request to OpenSSO Enterprise 1 where the session originated. Cookie-based persistency enables proper routing.

14. OpenSSO Enterprise 1 sends a response back to J2EE Policy Agent 1.
    a. If the session is not valid, J2EE Policy Agent 1 redirects the user back to Distributed Authentication User Interface 2.
    b. If the session is valid, J2EE Policy Agent 1 receives the response and sends a request for policy evaluation to Load Balancer 2.
15. As the session is valid, the request is directed to OpenSSO Enterprise 1 to conduct the policy evaluation.
16. Based on the outcome of the policy evaluation, J2EE Policy Agent 1 allows or denies access to the resource. In this scenario, the user is allowed access.

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" on page 29
- "2.2 Software" on page 30
- "2.3 Main Service URLs" on page 30
- "2.4 Intercomponent Communication" on page 32
- "2.5 Firewall Rules" on page 34
- "2.6 Viewing Replicated Entries" on page 35

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
da–1	SPARC	Solaris 10
da–2	SPARC	Solaris 10
ds–1	x86	Solaris 10
ds–2	x86	Solaris 10
mq–1	x86	Solaris 10
mq-2	x86	Solaris 10
osso–1	SPARC	Solaris 10
osso–2	SPARC	Solaris 10

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

Host Machine	Architecture	Operating System
pr–1	SPARC	Solaris 10
pr–2	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 Application Server	9.1 Update 1	http://www.sun.com/download/
Sun Java System Directory Server	6.1	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 Sun Java System Application Server and 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 table summarizes the main service URLs for the components used in this deployment example. For detailed configuration information, see Part III.

TABLE 2–3 Components and Main Service URLs

Components	Main Service URL
Directory Server Instances and Load Balancers	
Directory Server 1	`ldaps://ds-1.example.com:1736` (for monitor node)
	`ldaps://ds-1.example.com:1736` (for user data)
Directory Server 2	`ldaps://ds-2.example.com:1736` (for monitor node)
	`ldaps://ds-2.example.com:1736` (for user data)
Load Balancer 1	`ldaps://lb-1.example.com:489` (for user data)
OpenSSO Enterprise Instances and Load Balancer	
OpenSSO Enterprise 1	`https://osso-1.example.com:1081` (for monitor node)
	`https://osso-1.example.com:1081/opensso/console`
OpenSSO Enterprise 2	`https://osso-2.example.com:1081` (for monitor node)
	`https://osso-2.example.com:1081/opensso/console`
Load Balancer 2	`https://lb-2.example.com:1081`
Distributed Authentication User Interfaces and Load Balancer	
Distributed Authentication User Interface 1	`https://da-1.example.com:1443` (for monitor node)
	`https://da-1.example.com:1443/distAuth/` (for users)
Distributed Authentication User Interface 2	`https://da-2.example.com:1443` (for monitor node)
	`https://da-2.example.com:1443/distAuth/` (for users)
Load Balancer 3	`https://lb-3.example.com:1443` (secure port)
Protected Resources 1 and 2: Web Containers, Policy Agents and Load Balancers	
Web Container 1	`https://pr-1.example.com:8989` (for Sun Java System Web Server administration console)
Web Policy Agent 1	`http://pr-1.example.com:1080`
J2EE Container 1	`http://pr-1.example.com:7001/console` (for BEA Weblogic administration server)
J2EE Policy Agent 1	`http://pr-1.example.com:1081/agentapp`

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

Components	Main Service URL
Web Container 2	`https://pr-2.example.com:8989` (for Sun Java System Web Server administration console)
Web Policy Agent 2	`http://pr-2.example.com:1080`
J2EE Container 2	`http://pr-2.example.com:7001/console` (for BEA WebLogic administration server)
J2EE Policy Agent 2	`http://pr-2.example.com:1081/agentapp`
Policy Agent Load Balancers	
Load Balancer 4	`http://lb-4.example.com:90` (for web policy agents)
Load Balancer 5	`http://lb-5.example.com:91` (for J2EE policy agents)
Message Queue Broker Instances	
Message Queue 1	`http://mq-1.example.com:7777`
Message Queue 2	`http://mq-2.example.com:7777`

2.4 Intercomponent Communication

The following table provides an overview of the types of communication that take place between servers, load balancers, and other components in the deployment example.

TABLE 2–4 Summary of Intercomponent Communication

Entity A	Entity B	Bi-Directional	Port	Protocol	Traffic Type
Internet Users	Load Balancer 4		90	HTTP	Application Traffic
Internet Users	Load Balancer 5		91	HTTP	Application Traffic
Internet Users	Load Balancer 3		1443	HTTPS	Internet User Authentication
Load Balancer 3	Distributed Authentication User Interface 1		1443	HTTPS	Internet User Authentication

TABLE 2–4 Summary of Intercomponent Communication *(Continued)*

Entity A	Entity B	Bi-Directional	Port	Protocol	Traffic Type
Load Balancer 3	Distributed Authentication User Interface 2		1443	HTTPS	Internet User Authentication
Load Balancer 4	Protected Resource 1		1080	HTTP	Application Traffic
Load Balancer 4	Protected Resource 2		1080	HTTP	Application Traffic
Load Balancer 5	Protected Resource 1		1081	HTTP	Application Traffic
Load Balancer 5	Protected Resource 2		1081	HTTP	Application Traffic
Distributed Authentication User Interface 1	Load Balancer 2		1081	HTTPS	Internet User Authentication
Distributed Authentication User Interface 2	Load Balancer 2		1081	HTTPS	Internet User Authentication
Protected Resource 1	Load Balancer 2		1081	HTTPS	Agent - OpenSSO Enterprise communication
Protected Resource 2	Load Balancer 2		1081	HTTPS	Agent - OpenSSO Enterprise communication
Load Balancer 3	OpenSSO Enterprise 1		1081	HTTPS	Agent - OpenSSO Enterprise communication for authentication
Load Balancer 3	OpenSSO Enterprise 2		1081	HTTPS	Agent - OpenSSO Enterprise communication for authentication
OpenSSO Enterprise 1	OpenSSO Enterprise 2	Yes	1081	HTTPS	Back-channel communication
OpenSSO Enterprise 1	Message Queue 1		7777	HTTP	Session communication
OpenSSO Enterprise 1	Load Balancer 1		489	LDAPS	User profile communication for authentication
OpenSSO Enterprise 2	Message Queue 2		7777	HTTP	Session communication
OpenSSO Enterprise 2	Load Balancer- 2		489	LDAPS	User profile communication for authentication
Message Queue 1	Message Queue 2	Yes	7777	HTTP	Session communication
Message Queue 2	Message Queue 1	Yes	7777	HTTP	Session communication
Load Balancer 1	Directory Server 1		1736	LDAPS	User profile communication for authentication

TABLE 2–4 Summary of Intercomponent Communication *(Continued)*

Entity A	Entity B	Bi-Directional	Port	Protocol	Traffic Type
Load Balancer 1	Directory Server 2		1736	LDAPS	User profile communication for authentication
Directory Server 1	Directory Server 2	Yes	1489	LDAP	Data replication communication
Directory Server 2	Directory Server 1	Yes	1489	LDAP	Data replication communication

2.5 Firewall Rules

Actual firewalls are not set up in this deployment example. If firewalls were deployed they would protect critical components using three distinct security zones as illustrated in "1.1 Deployment Architecture and Components" on page 17. One zone is completely secure, protected by all three firewalls, and used for internal traffic only. The second, less secure zone is protected by only two firewalls but is also for internal traffic only. The third, minimally-secured *demilitarized zone* (DMZ) leaves only simple components and interfaces exposed to the Internet and is used for external traffic. Thus, direct access to individual instances of OpenSSO Enterprise and Directory Server is allowed only if permitted by firewall rules. Based on the illustration cited:

- The instances of OpenSSO Enterprise are isolated between an internal firewall and the DMZ, and exposed through an external-facing load balancer. The load balancer and instances together provide high data availability within the infrastructure.
- The policy agents themselves are deployed behind a load balancer configured in the DMZ.
- The Distributed Authentication User Interface would be deployed in the DMZ for communication with OpenSSO Enterprise behind a firewall, additionally protecting the OpenSSO Enterprise instances from exposure in the minimally-secured DMZ.

You may set up firewalls to allow traffic to flow as described in the following table.

TABLE 2–5 Summary of Firewall Rules

From	To	Port #	Protocol	Traffic Type
Internet users	Load Balancer 3	1443	HTTPS	User authentication
Internet users	Load Balancer 4	90	HTTP	Application access by internet user
Internet users	Load Balancer 5	91	HTTP	Application access by internet user

TABLE 2–5 Summary of Firewall Rules *(Continued)*

From	To	Port #	Protocol	Traffic Type
Distributed Authentication User Interface 1	Load Balancer 2	1081	HTTPS	User authentication
Distributed Authentication User Interface 2	Load Balancer 2	1081	HTTPS	User authentication
Load Balancer 4	Protected Resource 1	1080	HTTP	Application access by user
Load Balancer 5	Protected Resource 2	1081	HTTP	Application access by user

2.6 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" on page 37
- "3.2 Setting Up the Load Balancers" on page 37
- "3.3 Obtaining Secure Socket Layer Certificates" on page 38
- "3.4 Resolving Host Names" on page 38
- "3.5 Known Issues and Limitations" on page 39

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 III 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 sections require load-balancing hardware and software.

- "4.4 Configuring Load Balancer 1 for the User Data Instances" on page 64
- "5.2 Configuring Load Balancer 2 for OpenSSO Enterprise" on page 100
- "7.3 Configuring the Distributed Authentication User Interface Load Balancer" on page 154
- "9.1 Configuring the Web Policy Agents Load Balancer" on page 253
- "9.2 Configuring the J2EE Policy Agents Load Balancer" on page 262

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 test certificate issued by OpenSSL and 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 within each procedure. You should know how to request server certificates from your CA of choice before beginning a deployment. The following 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 61
- "To Install a Root Certificate and a Server Certificate on Directory Server 2" on page 63
- "To Install Application Server on the OpenSSO Enterprise 1 Host Machine" on page 80
- "To Install Application Server on the OpenSSO Enterprise 2 Host Machine" on page 91
- "To Request a Certificate for the OpenSSO Enterprise Load Balancer" on page 101
- "To Install a CA Root Certificate to the OpenSSO Enterprise Load Balancer" on page 102
- "To Install the Server Certificate to the OpenSSO Enterprise Load Balancer" on page 103
- "To Request and Install a Server Certificate and a Root Certificate for Web Server 1" on page 142
- "To Request and Install a Server Certificate and a Root Certificate for Web Server 2" on page 147
- "To Import the Root Certificate to the Web Server 1 JDK Certificate Store" on page 151
- "To Import the Root Certificate to the Web Server 2 JDK Certificate Store" on page 153
- "To Request a Certificate for the Distributed Authentication User Interface Load Balancer" on page 155
- "To Import a Root Certificate to the Distributed Authentication User Interface Load Balancer" on page 156
- "To Import a Certificate to the Distributed Authentication User Interface Load Balancer" on page 157

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        DirectoryServer-1     ds-1.example.com
1xx.xx.xx.x2        DirectoryServer-2     ds-2.example.com
1xx.xx.xx.x3        OpenSSO-1         osso-1.example.com
1xx.xx.xx.x4        OpenSSO-2         osso-2.example.com
```

3.5 Known Issues and Limitations

See Appendix F, "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 Environment

This second part of *Deployment Example: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0* provides the instructions for installing and configuring the deployment and its components. 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 Sun OpenSSO Enterprise User Data"
- Chapter 5, "Deploying and Configuring OpenSSO Enterprise"
- Chapter 6, "Configuring OpenSSO Enterprise Realms for User Authentication"
- Chapter 7, "Installing and Configuring the Distributed Authentication User Interface"
- Chapter 8, "Configuring the Protected Resource Host Machines"
- Chapter 9, "Setting Up Load Balancers for the Policy Agents"
- Chapter 10, "Implementing Session Failover"

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 Sun OpenSSO Enterprise 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:

- "4.1 Installing and Configuring Directory Server 1 and Directory Server 2" on page 43
- "4.2 Enabling Multi-Master Replication of the User Data Instances" on page 53
- "4.3 Enabling Secure Communication for the Directory Server User Data Instances" on page 61
- "4.4 Configuring Load Balancer 1 for the User Data Instances" on page 64
- "4.5 Importing Test Users" on page 74

Note – If you have an existing user data store, you can go directly to the instructions in Chapter 5, "Deploying and Configuring OpenSSO Enterprise," followed by Chapter 6, "Configuring OpenSSO Enterprise Realms for User Authentication."

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 and creating the directory instances named `am-users` in which the OpenSSO Enterprise 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 44
2. "To Patch the Directory Server Host Machines" on page 46
3. "To Install Directory Server 1" on page 47

4. "To Create an OpenSSO Enterprise User Data Instance on Directory Server 1" on page 48
5. "To Create a Base Suffix for the User Data Instance on Directory Server 1" on page 49
6. "To Install Directory Server 2" on page 50
7. "To Create an OpenSSO Enterprise User Data Instance on Directory Server 2" on page 51
8. "To Create a Base Suffix for the User Data Instance on Directory Server 2" on page 52

▼ 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) 6.1 bits and the required system patches to both the Directory Server 1 host machine (`ds-1.example.com`) and the Directory Server 2 host machine (`ds-2.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`**
Step 2: Select Version	**`6.1`**
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 118855–36, patch 119964–08, and patch 122033–05 are required.

---

3 **Log into the `ds–1` 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.

```
# patchadd -p | grep 118855–36
```

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

```
# patchadd -p | grep 119964–08
```

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

```
# patchadd -p | grep 122033–05
```

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/DS61
# cd /export/DS61
```

8 **Download the *Base Full Install* of Directory Server EE 6.1 — 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 `ds–1` host machine.**

10 **Repeat this same procedure on the `ds–2` host machine.**

▼ To Patch the Directory Server Host Machines

If necessary, use this procedure to patch both the `ds–1` host machine and the `ds–2` host machine.

1. **Log in to the `ds–1` 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 118855–36.zip
# unzip 119964-08.zip
# unzip 122033-05.zip
```

4. **Install the patches.**

```
# patchadd /export/patches/118855-36
# patchadd /export/patches/119964-08
# patchadd /export/patches/122033-05
```

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.**

```
# patchadd -p | grep 118855–36
```

A series of patch numbers are displayed, and the patch 118855–36 is present.

```
# patchadd -p | grep 119964-08
```

A series of patch numbers are displayed, and the patch 119964-08 is present.

```
# patchadd -p | grep 122033-05
```

A series of patch numbers are displayed, and the patch 122033-05 is present.

7. **Log out of the `ds–1` host machine.**

8. **Repeat this same procedure on the `ds–2` 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 44 and "To Patch the Directory Server Host Machines" on page 46 have been completed.

1. **Log in to the `ds–1` 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/DS61
   # ls

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

   # gunzip DSEE.6.1.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 an OpenSSO Enterprise User Data Instance on Directory Server 1

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. It will be populated with user data in "4.5 Importing Test Users" on page 74.

Before You Begin This procedure assumes you have just completed "To Install Directory Server 1" on page 47 and are still logged into the `ds–1` 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 `am-users`.**

```
# ./dsadm create -p 1489 -P 1736 /var/opt/mps/am-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/am-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 ds-1.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/6.1
...
```

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 an OpenSSO Enterprise User Data Instance on Directory Server 1" on page 48 and are still logged into the `ds-1` 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 dc=company,dc=com
```

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

```
Certificate "CN=ds-1, 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 `am-users` in a command line list of directory instances.

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

am-users serverroot
```

4 **Log out of the `ds–1` 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 44 and "To Patch the Directory Server Host Machines" on page 46 have been completed.

1 **Log in to the `ds–2` 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/DS61
# ls

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

# gunzip DSEE.6.1.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 an OpenSSO Enterprise 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. It will be populated with user data in "4.5 Importing Test Users" on page 74.

Before You Begin This procedure assumes you have just completed "To Install Directory Server 2" on page 50 and are still logged into the `ds–2` 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 `am-users`.**

```
# ./dsadm create -p 1489 -P 1736 /var/opt/mps/am-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/am-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 ds-2.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/6.1
...
```

▼ 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 an OpenSSO Enterprise User Data Instance on Directory Server 2" on page 51 and are still logged into the `ds-2` 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 dc=company,dc=com
```

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

```
Certificate "CN=ds-2, 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 `am-users` in a command line list of directory instances.

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

am-users serverroot
```

4 **Log out of the** `ds–2` **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 `am-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 54
2. "To Enable Multi-Master Replication for User Data Instance on Directory Server 2" on page 55
3. "To Change the Default Replication Manager Password for Each User Data Instance" on page 56
4. "To Create Replication Agreements for Each User Data Instance" on page 57
5. "To Initialize the Replication Agreements" on page 58
6. "To Verify Successful User Data Replication" on page 59

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

1. **Log in to the `ds–1` 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 "ds-1.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 ds-1.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
"ds-1.example.com:1489"
```

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

5. **Log out of the `ds–1` host machine.**

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

1. **Log in to the `ds–2` 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 "ds-2.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 ds-2.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
"ds-2.example.com:1489"
```

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

5. **Log out of the `ds–2` 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 `ds–1` 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 ds-1.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 `ds–1` host machine.**

7. **Log in to the `ds–2` 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 ds-2.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** `ds–2` **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, we are simply making the user data instances aware of each other.

1 **Log in to the** `ds–1` **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 ds-1.example.com
 -p 1489 dc=company,dc=com ds-2.example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf init-repl-dest dc=company,dc=com ds-2.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 ds-2.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** `ds–1` **host machine.**

5 **Log in to the** `ds–2` **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 ds-2.example.com -p 1489
dc=company,dc=com ds-1.example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Use "dsconf init-repl-dest dc=company,dc=com ds-1.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 ds-1.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** `ds–2` **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** `ds–1` **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 ds-1.example.com
 -p 1489 dc=company,dc=com ds-2.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 ds-1.example.com
 -p 1489 dc=company,dc=com ds-2.example.com:1489

Enter "cn=Directory Manager" password: dsmanager

Started initialization of "ds-2.example.com:1489"; Aug 25, 2008 3:10:01 PM
Sent 2 entries.
Completed initialization of "ds-2.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 ds-1.example.com
 -p 1489 dc=company,dc=com ds-2.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 58 and are still logged into the `ds–1` host machine as a root user.

1 **Run** `ldapmodify` **on the** `ds-1` **host machine to create a new directory entry.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapmodify -a -h ds-1.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 `ds–2` 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 **Now run `ldapdelete` on Directory Server 2 to delete the entry just created.**

```
# ./ldapdelete -h ds-2.example.com -p 1489
 -D "cn=Directory Manager" -w dsmanager
"ou=People,dc=company,dc=com"
```

5 **Now, as a root user on Directory Server 1, run `ldapsearch` 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 Enabling Secure Communication for the Directory Server User Data Instances

By default, when an instance of Directory Server is created (in this case, `am-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 `am-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 61
2. "To Install a Root Certificate and a Server Certificate on Directory Server 2" on page 63

▼ 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 38.

1. **Log in to the `ds–1` 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=ds-1.example.com,
OU=OpenSSO Enterprise, O=Sun Microsystems, L=Santa Clara
 ST=California, C=US" -F ascii -o ds-1.csr /var/opt/mps/am-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/am-users ds-1 ds-1.cer
```

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

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

   A list of certificates for the `am-users` instance is displayed including the `defaultCert` and `ds-1`.

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

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

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

```
# ./dsadm list-certs -C /var/opt/mps/am-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 ds-1.example.com
-p 1489 ssl-rsa-cert-name:ds-1

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/am-users
# ./dsadm start /var/opt/mps/am-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 ds-1.example.com -p 1736
-Z -P /var/opt/mps/am-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 `ds–1` 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 38.

1 **Log in to the `ds–2` 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=ds-2.example.com,
OU=OpenSSO Enterprise, O=Sun Microsystems, L=Santa Clara
 ST=California, C=US" -F ascii -o ds-2.csr /var/opt/mps/am-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/am-users ds-2 ds-2.cer
```

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

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

A list of certificates for the `am-users` instance is displayed including the `defaultCert` and `ds-2`.

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

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

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

```
# ./dsadm list-certs -C /var/opt/mps/am-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 ds-2.example.com
-p 1489 ssl-rsa-cert-name:ds-2

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/am-users
# ./dsadm start /var/opt/mps/am-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 ds-2.example.com -p 1736
-Z -P /var/opt/mps/am-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 `ds–2` host machine.**

4.4 Configuring Load Balancer 1 for the User Data Instances

Load Balancer 1 is configured in front of the Directory Server user data instances. 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 Request a Certificate for the User Data Load Balancer" on page 65
2. "To Import the Root Certificate to the User Data Load Balancer" on page 66
3. "To Install the Server Certificate to the User Data Load Balancer" on page 67
4. "To Configure the User Data Load Balancer 1" on page 68
5. "To Create an SSL Proxy for SSL Termination at the User Data Load Balancer 1" on page 72

▼ To Request a Certificate for the User Data Load Balancer

Generate a request for a server certificate to send to a CA. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 38.

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 using the following information.**

Username *username*

Password *password*

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:	**`lb-1.example.com`**
Organizational Unit Name:	**`Deployment`**
Domain Name:	**`lb-1.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-1.csr`.**

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

11. **Send `lb-1.csr` to the CA of your choice.**
    The CA issues and returns a certified server certificate named `lb-1.cer`.

▼ To Import the Root Certificate to the User Data Load Balancer

Import the CA root certificate on Load Balancer 1 to ensure that a link between Load Balancer 1 can be maintained with the CA. Use the same root certificate that you imported in "4.3 Enabling Secure Communication for the Directory Server User Data Instances" on page 61. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 38.

Before You Begin You should already have a root certificate from the CA of your choice.

1. **Access `https://is-f5.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 Certificate OpenSSL_CA_Cert page is displayed.
12. **Click Return to Certificate Administration on the Certificate OpenSSL_CA_Cert page.**

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

▼ To Install the Server Certificate to the User Data Load Balancer

Before You Begin This procedure assumes you have received the server certificate requested in "To Request a Certificate for the User Data Load Balancer" on page 65, just completed "To Import the Root Certificate to the User Data Load Balancer" on page 66, 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 `lb-1.example.com` is in the Key List.
3. **In the Certificate ID column, click Install for `lb-1.example.com`.**
4. **In the Certificate File field, click Browse.**
5. **In the Choose File dialog, navigate to `lb-1.cer`, the server certificate, and click Open.**
6. **Click Install Certificate.**
7. **On the Certificate lb-1.example.com page, click Return to Certificate Administration Information.**

   Verify that the Certificate ID indicates `lb-1.example.com` on the SSL Certificate Administration page.
8. **Log out of the load balancer console.**

▼ To Configure the User Data 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 66 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	**`DirectoryServer-UserData-Pool`**
Load Balancing Method	Round Robin
Resources	Add the IP address and port number of both Directory Server host machines: `ds-1:1736` and `ds-2:1736`.

   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 `lb-1.example.com`
Service	**`490`**

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

e. **Assign `DirectoryServer-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, `ds-1:1736`, 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, `ds–2:1736`, 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 persistence.**

The user data load balancer is configured 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, `DirectoryServer-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.**

a. **Log in as a root user to the host machine of each Directory Server instance.**

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

```
# cd /var/opt/mps/am-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 lb-1.example.com -p 490 -Z
-P /var/opt/mps/am-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
```

The `ldapsearch` operation should return entries. Make sure they display in the access log on only one Directory Server.

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

```
# cd /var/opt/mps/serverroot/ds6/bin
# ./dsadm stop /var/opt/mps/am-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 lb-1.example.com -p 490 -Z
-P /var/opt/mps/am-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
```

The `ldapsearch` operation should return entries. Verify that the 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/am-users
```

g. Stop Directory Server 2.

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

h. Perform the following 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 lb-1.example.com -p 490 -Z
-P /var/opt/mps/am-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
```

The `ldapsearch` operation should return entries. Make sure the entries display in the access log on only Directory Server 1.

i. Start Directory Server 2.

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

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

▼ To Create an SSL Proxy for SSL Termination at the User Data Load Balancer 1

SSL communication is terminated at Load Balancer 1. The request is then re-encrypted and securely forwarded to the SSL port of the Directory Server user data instance. Load Balancer 1 also encrypts the responses it receives back from the user data instance, 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 with the following information.**

   User name: *username*

Password: *password*

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 1.
Proxy Service:	**`489`**
	The secure port number
Destination Address:	The IP address of Load Balancer 1.
Destination Service:	**`490`**
	The non-secure port number
Destination Target:	Choose **`Local Virtual Server`**.
SSL Certificate:	Choose **`lb-1.example.com`**.
SSL Key:	Choose **`lb-1.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 and 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.**

4.5 Importing Test Users

Create user entries in the replicated Directory Server user data instances for the following users:

- `testuser1`
- `testuser2`

These users will be used to verify that the policy agent is configured and working properly. Additionally, the `Groups` container will be used for the same purpose.

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."

Use the following procedure, "To Import Test User Data into the Replicated Directory Server Instances" on page 74, to create an LDIF file for the test users and import the file into `ds–1`. The test users will then be replicated to `ds–2`.

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

1 **Log in to the `ds–1` 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=testuser1,ou=users,dc=company,dc=com
uid: testuser1
givenName: Test
```

```
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetadmin
objectClass: inetorgperson
objectClass: inetUser
sn: User1
cn: Test User1
userPassword: password
inetUserStatus: Active

dn: uid=testuser2,ou=users,dc=company,dc=com
uid: testuser2
givenName: Test
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
objectClass: inetUser
sn: User2
cn: Test User2
userPassword: password
inetUserStatus: Active
```

3 **Save the file as** `am-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 ds-1.example.com -p 1489
 -D "cn=Directory Manager" -w dsmanager
 -a -f /tmp/am-users.ldif

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

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

adding new entry uid=testuser1,ou=users,dc=company,dc=com

adding new entry uid=testuser2,ou=users,dc=company,dc=com
```

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

```
# ./ldapsearch -h ds-1.example.com
 -b "dc=company,dc=com" -p 1489 -D "cn=Directory Manager"
 -w dsmanager "uid=test*"

version: 1
```

```
dn: uid=testuser1,ou=users,dc=company,dc=com
uid: testuser1
givenName: Test
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetadmin
objectClass: inetorgperson
objectClass: inetUser
sn: User1
cn: Test User1
userPassword: {SSHA}H5LpB+QLZMoL9SiXzY/DokHKXRclELVy7w25AA==
inetUserStatus: Active

dn: uid=testuser2,ou=users,dc=company,dc=com
uid: testuser2
givenName: Test
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
objectClass: inetUser
sn: User2
cn: Test User2
userPassword: {SSHA}aLNFCQ1qw78KpJeloVZJAAa5QSAPf/9c2mxCQQ==
inetUserStatus: Active
```

6 **Log out of the `ds-1` host machine.**

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

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -h ds-2.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=testuser1,ou=users,dc=company,dc=com
uid: testuser1
givenName: Test
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetadmin
objectClass: inetorgperson
objectClass: inetUser
sn: User1
cn: Test User1
inetUserStatus: Active
userPassword: {SSHA}H5LpB+QLZMoL9SiXzY/DokHKXRclELVy7w25AA==

dn: uid=testuser2,ou=users,dc=company,dc=com
uid: testuser2
givenName: Test
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
objectClass: inetUser
sn: User2
cn: Test User2
inetUserStatus: Active
userPassword: {SSHA}aLNFCQ1qw78KpJeloVZJAAa5QSAPf/9c2mxCQQ==
```

8 **Log out of the `ds-2` host machine.**

CHAPTER 5

Deploying and Configuring OpenSSO Enterprise

This chapter contains instructions on how to deploy and configure two instances of Sun OpenSSO Enterprise 8.0. Post installation procedures are also included. It begins with the installation of an instance of Sun Java™ System Application Server (on each host machine) into which the OpenSSO Enterprise WAR will be deployed and contains the following sections:

- "5.1 Installing the Application Server Web Containers" on page 79
- "5.2 Configuring Load Balancer 2 for OpenSSO Enterprise" on page 100
- "5.3 Deploying and Configuring OpenSSO Enterprise 1 and OpenSSO Enterprise 2" on page 108
- "5.4 Configuring the OpenSSO Enterprise Platform Service" on page 117

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

1. "To Create a Non-Root User on the OpenSSO Enterprise 1 Host Machine" on page 80
2. "To Install Application Server on the OpenSSO Enterprise 1 Host Machine" on page 80
3. "To Create a Non-Root User on the OpenSSO Enterprise 2 Host Machine" on page 90
4. "To Install Application Server on the OpenSSO Enterprise 2 Host Machine" on page 91

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

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

1. **Log in to the `osso–1` 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 80 and are still logged into the `osso–1` 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://`*host-machine*`.`*domain*`: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 F, "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://osso-1.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 Load Balancer 2 and `ossodomain` using the following sub-procedure.**

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

`osso-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 osso-1
 -keystore keystore.jks -dname "CN=osso-1.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: osso-1
 Creation date: Aug 4, 2008
 Entry type: keyEntry
 Certificate chain length: 1
 Certificate[1]:
 Owner: CN=osso-1.example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
 Issuer: CN=osso-1.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 "osso-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 osso-1 -keypass domain2master
-keystore keystore.jks -storepass domain2master file osso-1.csr
```

`osso-1.csr` is the server certificate request.

d. **(Optional) Verify that `osso-1.csr` was created.**

```
# ls -la osso-1.csr

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

e. **Send `osso-1.csr` to the CA of your choice.**

The CA issues and returns a certified certificate named `osso-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 `osso-1` alias) with the server certificate received from the CA.

```
# keytool -import -file osso-1.cer -alias osso-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-1" is signed by CA.
```

i. Change the certificate alias from the default `s1as` to the new `osso-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="osso-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
```

Note – The second Application Server domain is only running as a non-root user and not sharing the domain administrator credentials used to start the server with the non-root user.

---

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

a. **Access `https://osso-1.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 `osso-1` host machine.**

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

1 **Log in to the `osso–2` 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 90 and are still logged into the `osso–2` 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://`*host-machine*`.`*domain*`: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 – The `FileNotFoundException` is a known issue. Please see Appendix F, "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://osso-2.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 Load Balancer 2 and `ossodomain` using the following sub-procedure.**

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

`osso-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 osso-2
 -keystore keystore.jks -dname "CN=osso-2.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: osso-2
 Creation date: Aug 4, 2008
 Entry type: keyEntry
 Certificate chain length: 1
 Certificate[1]:
 Owner: CN=osso-2.example.com, OU=OpenSSO, O=Sun Microsystems,
L=Santa Clara, ST=California, C=US
 Issuer: CN=osso-2.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 "osso-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 osso-2 -keypass domain2master
-keystore keystore.jks -storepass domain2master file osso-2.csr
```

`osso-2.csr` is the server certificate request.

d. **(Optional) Verify that `osso-2.csr` was created.**

```
# ls -la osso-2.csr

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

e. **Send `osso-2.csr` to the CA of your choice.**

The CA issues and returns a certified server certificate named `osso-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 `osso-2` alias) with the server certificate received from the CA.

```
# keytool -import -file osso-2.cer -alias osso-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 "osso-2" is signed by CA.
```

i. Change the certificate alias from the default `s1as` to the new `osso-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="osso-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
```

Note – The second Application Server domain is only running as a non-root user and not sharing the domain administrator credentials used to start the server with the non-root user.

---

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

a. Access `https://osso-2.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 `osso-2` host machine.**

5.2 Configuring Load Balancer 2 for OpenSSO Enterprise

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. Users external to the company will access the Distributed Authentication User Interface which, in turn, routes the request 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 the OpenSSO Enterprise Load Balancer" on page 101
2. "To Install a CA Root Certificate to the OpenSSO Enterprise Load Balancer" on page 102
3. "To Install the Server Certificate to the OpenSSO Enterprise Load Balancer" on page 103
4. "To Configure the OpenSSO Enterprise Load Balancer" on page 103
5. "To Create an SSL Proxy for SSL Termination at the OpenSSO Enterprise Load Balancer" on page 106

▼ To Request a Certificate for the OpenSSO Enterprise Load Balancer

Generate a request for a server certificate to send to a CA. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 38.

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 using the following information.**

   Username *username*

   Password *password*

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: **`lb-2.example.com`**

   Organizational Unit Name: **`Deployment`**

   Domain Name: **`lb-2.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 a CA Root Certificate to the OpenSSO Enterprise Load Balancer

Install the CA root certificate on Load Balancer 2 to ensure that a link between the Load Balancer 2 can be maintained with the CA. Use the same root certificate that you imported in "4.3 Enabling Secure Communication for the Directory Server User Data Instances" on page 61. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 38.

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

2 **Log in with the following information.**

User name: *username*

Password: *password*

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 the OpenSSO Enterprise Load Balancer

Before You Begin This procedure assumes you have received the server certificate requested in "To Request a Certificate for the OpenSSO Enterprise Load Balancer" on page 101 and just completed "To Install a CA Root Certificate to the OpenSSO Enterprise Load Balancer" on page 102.

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

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

The key `lb-2.example.com` is in the Key List.

3 **In the Certificate ID column, click Install for `lb-2.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 lb-2.example.com page, click Return to Certificate Administration Information.**

Verify that the Certificate ID indicates `lb-2.example.com` on the SSL Certificate Administration page.

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

▼ To Configure the OpenSSO Enterprise Load Balancer

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

2 **Log in using the following information:**

User name: *username*

Password: *password*

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	**`OSSO-Pool`**
Load Balancing Method	Round Robin
Resources	Add the IP addresses and port numbers for the OpenSSO Enterprise servers: `osso-1:1081` and `osso-2:1081`.

   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 `lb-2.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** `OSSO-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 `osso-1:1081` and `osso-2:1081`.**

d. **Mark the Add checkbox for `OSSO-1` and `OSSO-2`.**

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, `OSSO-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 `OSSO-1` 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, 692589248.36895.0000) for use in "To Create a Site on OpenSSO Enterprise 1" on page 118.

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 `OSSO-2` 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, 692589248.12345.0000) for use in "To Create a Site on OpenSSO Enterprise 1" on page 118.

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

▼ To Create an SSL Proxy for SSL Termination at the OpenSSO Enterprise Load Balancer

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 with the following information.**

User name: *username*

Password: *password*

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 **`lb-2.example.com`**.
SSL Key:	Choose **`lb-2.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://lb-2.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 108
2. "To Deploy the OpenSSO Enterprise WAR as OpenSSO Enterprise 1" on page 110
3. "To Copy the OpenSSO Enterprise WAR to the OpenSSO Enterprise 2 Host Machine" on page 112
4. "To Deploy the OpenSSO Enterprise WAR File as OpenSSO Enterprise 2" on page 113
5. "To Configure OpenSSO Enterprise 1" on page 114
6. "To Configure OpenSSO Enterprise 2" on page 116

▼ To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine

1 **As a root user, log in to the `osso–1` 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.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 108 and are still logged into the `osso–1` host machine

1 **On the `osso–1` 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 osso-1.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 `osso–1` 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 108.

1. **As a root user, log in to the `osso–2` host machine.**

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 `osso–1` 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 112 and are still logged into the `osso–2` host machine

1. **On the `osso–2` 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 osso-2.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 `osso–2` host machine.**

▼ To Configure OpenSSO Enterprise 1

1 **Access `https://osso-1.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	**`lb-1.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. **As a root user, log in to the `osso–1` 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
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 `osso–1` host machine.**

▼ To Configure OpenSSO Enterprise 2

1. **Access `https://osso-2.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://osso-1.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. **As a root user, log in to the `osso–2` 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
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 `osso–2` 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 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 118
2. "To Verify that the OpenSSO Enterprise Site was Configured Properly" on page 120

▼ To Create a Site on OpenSSO Enterprise 1

It is **not** necessary to repeat this procedure on OpenSSO Enterprise 2.

1. **Access `https://osso-1.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://lb-2.example.com:1081/opensso`**

   A new site called External is displayed in the Sites list.

6. **Click on the `https://osso-1.example.com:1081/opensso` server entry under the Servers list.**

   The Edit `https://osso-1.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 `OSSO-1` 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 the OpenSSO Enterprise Load Balancer" on page 103.

10. **Click Back to Server and Sites.**

11. **Click on the `https://osso-2.example.com:1081/opensso` server entry under the Servers list.**

    The Edit `https://osso-2.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 `OSSO-2` 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 the OpenSSO Enterprise Load Balancer" on page 103.

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 `osso–1` host machine.**

18 **Restart OpenSSO Enterprise 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 `osso–2` host machine.**

20 **Restart OpenSSO Enterprise 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://lb-2.example.com:1081/opensso/UI/Login`**.**

If an error message is displayed indicating that the browser cannot connect to either `osso-1.example.com` or `osso-2.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 secure 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.**

CHAPTER 6

Configuring OpenSSO Enterprise Realms for User Authentication

This chapter contains instructions on configuring OpenSSO Enterprise,to use an external user data store for authentication. (The external user data store and test users were set up in Chapter 4, "Installing Sun Java System Directory Server and Creating Instances for Sun OpenSSO Enterprise User Data"). 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 121
- "6.2 Creating and Configuring a Sub Realm for Test Users" on page 123

Caution – Do not do both procedures.

6.1 Modifying the Top-Level Realm for Test Users

At this point in the deployment, the OpenSSO Enterprise 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 now modify the external user data store configuration details using the OpenSSO Enterprise console to map the user data stores schema to the test user entries previously imported. 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 122
2. "To Verify that a User Can Successfully Authenticate" on page 123

▼ To Modify the Top-Level Realm for User Authentication

1. **Access `https://osso-1.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 `GenericLDAPv3` data store link is displayed.

6. **Click `GenericLDAPv3`.**

7. **On the `GenericLDAPv3` data store 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.**

   `testuser1` and `testuser2` are 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 a test user.

1 **Access** `https://osso-1.example.com:1081/opensso/UI/Login` **in a web browser.**

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

User Name: **`testuser1`**

Password: **`password`**

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.

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. Create a sub realm to authenticate external users against the Directory Server user data store instances. This 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.

- "To Create a Sub Realm" on page 124

- "To Change the User Profile Configuration for the Sub Realm" on page 124
- "To Modify the Sub Realm for User Authentication" on page 125
- "To Verify That the Sub Realm Can Access the External User Data Store" on page 126
- "To Verify That the Sub Realm Subjects Can Successfully Authenticate" on page 127

To Create a Sub Realm

When a sub realm is created it inherits configuration data (including which user data store) from `/ (Top Level Realm)` (the default root realm) and uses it to authenticate users. The user data store can be modified per sub realm. In this deployment, we use the inherited `GenericLDAPv3` data store.

1. **Access `https://osso-1.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 124 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 **Log out of the OpenSSO Enterprise console.**

▼ To Modify the Sub Realm for User Authentication

1 **Access** `https://osso-1.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** `users`**, the sub realm, under the Access Control tab.**

5 **Click the Data Stores tab.**

The `GenericLDAPv3` data store link is displayed.

6 **Click** `GenericLDAPv3`**.**

7 **On the** `GenericLDAPv3` **data store 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.**

`testuser1` and `testuser2` are displayed under Users (as well as others created during OpenSSO Enterprise configuration).

10 **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 made in "To Create a Sub Realm" on page 124 and "To Change the User Profile Configuration for the Sub Realm" on page 124.

1 **Access `https://osso-1.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.**

`testuser1` and `testuser2` are displayed under Users.

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

▼ To Verify That the Sub Realm Subjects Can Successfully Authenticate

1. **Access `https://osso-1.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 a user name and password from the `am-users` directory.**

   User Name **`testuser1`**

   Password **`password`**

   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 7

Installing and Configuring the Distributed Authentication User Interface

OpenSSO Enterprise provides a remote authentication interface to enable secure authentication. Deploying the Distributed Authentication User Interface to one or more web containers within a non-secure layer eliminates the exposure of service URLs to the end user. This chapter contains the procedures to install and configure the Distributed Authentication User Interface in the following sections.

- "7.1 Installing the Distributed Authentication User Interface Web Containers" on page 129
- "7.2 Enabling Secure Communications Between the Web Server Instances and the Load Balancer" on page 142
- "7.3 Configuring the Distributed Authentication User Interface Load Balancer" on page 154
- "7.4 Creating an Agent Profile with Custom User for the Distributed Authentication User Interface" on page 162
- "7.5 Generating and Deploying the Distributed Authentication User Interface WAR" on page 164

7.1 Installing the Distributed Authentication User Interface Web Containers

In this section, we will create a non-root user on the two machines that will host the Distributed Authentication User Interface and install Sun Java System Web Server using the non-root user. Use the following list of procedures as a checklist for completing the task.

- "To Create a Non-Root User on the Distributed Authentication User Interface 1 Host Machine" on page 130
- "To Install the Web Server for Distributed Authentication User Interface 1" on page 130
- "To Create a Non-Root User on the Distributed Authentication User Interface 2 Host Machine" on page 136
- "To Install Sun Java System Web Server for Distributed Authentication User Interface 2" on page 137

▼ To Create a Non-Root User on the Distributed Authentication User Interface 1 Host Machine

Create the non-root user using the `roleadd` command in the Solaris Operating Environment on the Distributed Authentication User Interface 1 (`da-1`) host machine.

1. **As a root user, log in to the `da-1` host machine.**

2. **Use `roleadd` to create a new user.**

```
# roleadd -s /sbin/sh -m -g staff -d /export/da80adm da80adm
```

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:/:
da80adm:x:223830:10::/export/da80adm:/sbin/sh
```

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

```
# cd /export/da80adm
# ls

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

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

```
# passwd da80adm
New Password: da80a6m
Re-ener new Pasword: da80a6m

passwd: password successfully changed for da80adm
```

Note – 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 the Web Server for Distributed Authentication User Interface 1

Before You Begin

- This procedure assumes that you have just completed "To Create a Non-Root User on the Distributed Authentication User Interface 1 Host Machine" on page 130 and are still logged in as the root user.

- Read the Web Server 7.0 Release Notes to determine the latest patches you might need to install.

1 **On the `da-1` host machine, install required patches if necessary.**

In this case, the Release Notes indicate that based on the hardware and operating system being used, patch 117461–08, patch 119963–08, and patch 120011–14 are required.

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

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

A list of patch numbers is displayed. This machine is already patched with 117461–08.

```
# 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.

b. **Make a directory for downloading the patches you need and change into it.**

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

c. **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.

---

d. **Unzip the patch files.**

```
# unzip 119963-08.zip
# unzip 120011-14.zip
```

e. **Run `patchadd` to install the patches.**

```
# patchadd /export/patches/119963-08
# patchadd /export/patches/120011-14
```

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

---

f. **After installation is complete, run `patchadd` to verify that each patch was added successfully.**

```
# patchadd -p | grep 119963-08
```

A series of patch numbers is displayed, and the patch 119963-08 is present.

```
# patchadd -p | grep 120011-14
```

A series of patch numbers is displayed, and the patch 120011–14 is present.

2 **Create a directory into which you can download the Web Server bits and change into it.**

```
# mkdir /export/WS7
# cd /export/WS7
```

3 **Download the Sun Java System Web Server 7.0 Update 2 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.

4 **Unpack the software package.**

```
# gunzip sjsws-7_0u2-solaris-sparc.tar.gz
# tar xvf sjsws-7_0u2-solaris-sparc.tar
```

5 **Run `setup`.**

```
# cd /export/WS7
# ./setup --console
```

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

`You will be asked to specify preferences that determine how Sun Java System Web Server 7.0U2 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]`
`{"<" 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-bit 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_12 or greater.`
 	
`1. Install Java SE Development Kit (JDK)`
`  1.5.0_12`
`2. Reuse existing Java SE Development Kit`
`  (JDK) 1.5.0_12 or greater`
`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]`
`{"<" goes back, "!" exits}`
	Enter **no**.

`Host Name [da-1.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]` `{"<" goes back, "!" exits}`	Enter **no**.
`Runtime User ID [root]` `{"<" goes back, "!" exits}`	Enter **da80adm**.
`Administrator User Name [admin]` `{"<" goes back, "!" exits}`	Accept the default value.
`Administrator Password:`	Enter **web4dmin**.
`Retype Password:`	Enter **web4dmin**.
`Server Name [da-1.example.com]` `{"<" goes back, "!" exits}`	Accept the default value.
`HTTP Port [8080]` `{"<" goes back, "!" exits}`	Enter **1080**.

`Document Root Directory [/opt/SUNWwbsvr/`
`https-da-1.example.com/docs]`
`{"<" goes back, "!" exits}`
	Accept the default value.

`Start Administration Server` `[yes/no] {"<" goes back, "!" exits}`	Enter **no**.

`Ready To Install`
 	
`1. Install Now`
`2. Start Over`
`3. Exit Installation`
 	
`What would you like to do?`
	Enter **1**.

When installation is complete, the following message is displayed:

```
Installation Successful.
```

7 **(Optional) To verify that Web Server was installed with the non-root user, examine the file permissions.**

```
# cd /opt/SUNWwbsvr/admin-server
# ls -al

total 16
drwxr-xr-x   8 root     root         512 Jul 19 10:36 .
drwxr-xr-x  11 da80adm  staff        512 Jul 19 10:36 ..
drwxr-xr-x   2 root     root         512 Jul 19 10:36 bin
drwx------   2 da80adm  staff        512 Jul 19 10:36 config
drwx------   3 da80adm  staff        512 Jul 19 11:09 config-store
drwx------   3 da80adm  staff        512 Jul 19 10:40 generated
drwxr-xr-x   2 da80adm  staff        512 Jul 19 10:40 logs
drwx------   2 da80adm  staff        512 Jul 19 10:36 sessions
```

The appropriate files and directories are owned by `da80adm`.

8 **Start the Web Server administration server.**

```
# su da80adm
# cd /opt/SUNWwbsvr/admin-server/bin
# ./startserv
```

9 **(Optional) Verify that the non-root user was able to start Web Server.**

a. **Access** `https://da-1.example.com:8989` **from a web browser.**

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

User Name: **`admin`**

Password: **`web4dmin`**

The Web Server administration console opens.

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

10 **Log out of the** `da–1` **host machine.**

▼ To Create a Non-Root User on the Distributed Authentication User Interface 2 Host Machine

Create the non-root user using the `roleadd` command in the Solaris Operating Environment on the Distributed Authentication User Interface 2 (`da-2`) host machine.

1. **As a root user, log in to the `da-2` host machine.**

2. **Use `roleadd` to create a new user.**

```
# roleadd -s /sbin/sh -m -g staff -d /export/da80adm da80adm
```

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:/:
da80adm:x:227627:10::/export/da80adm:/sbin/sh
```

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

```
# cd /export/da80adm
# ls

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

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

```
# passwd da80adm
New Password: da80a6m
Re-ener new Pasword: da80a6m

passwd: password successfully changed for da80adm
```

Note – 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 Sun Java System Web Server for Distributed Authentication User Interface 2

Before You Begin

- This procedure assumes that you have just completed "To Create a Non-Root User on the Distributed Authentication User Interface 2 Host Machine" on page 136 and are still logged in as the root user.
- Read the Web Server 7.0 Release Notes to determine the latest patches you might need to install.

1. **On the `da-2` host machine, install required patches if necessary.**

   In this case, the Release Notes indicate that based on the hardware and operating system being used, patch 117461–08, patch 119963–08, and patch 120011–14 are required.

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

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

   A list of patch numbers is displayed. This machine is already patched with 117461–08.

   ```
   # 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.

   b. **Make a directory for downloading the patches you need and change into it.**

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

   c. **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.

   ---

   d. **Unzip the patch files.**

   ```
   # unzip 119963-08.zip
   # unzip 120011-14.zip
   ```

e. **Run `patchadd` to install the patches.**

```
# patchadd /export/patches/119963-08
# patchadd /export/patches/120011-14
```

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

---

f. **After installation is complete, run `patchadd` to verify that each patch was added successfully.**

```
# patchadd -p | grep 119963-08
```

A series of patch numbers is displayed, and the patch 119963-08 is present.

```
# patchadd -p | grep 120011-14
```

A series of patch numbers is displayed, and the patch 120011–14 is present.

2 **Create a directory into which you can download the Web Server bits and change into it.**

```
# mkdir /export/WS7
# cd /export/WS7
```

3 **Download the Sun Java System Web Server 7.0 Update 2 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.

4 **Unpack the software package.**

```
# gunzip sjsws-7_0u2-solaris-sparc.tar.gz
# tar xvf sjsws-7_0u2-solaris-sparc.tar
```

5 **Run `setup`.**

```
# cd /export/WS7
# ./setup --console
```

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

`You will be asked to specify`
`preferences that determine how Sun Java`
`System Web Server 7.0U2 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]`
`{"<" 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-bit 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_12 or greater.`
 	
`1. Install Java SE Development Kit (JDK)`
`  1.5.0_12`
`2. Reuse existing Java SE Development Kit`
`  (JDK) 1.5.0_12 or greater`
`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]`
`{"<" goes back, "!" exits}`
	Enter **no**.

`Host Name [da-2.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]` `{"<" goes back, "!" exits}`	Enter **no**.
`Runtime User ID [root]` `{"<" goes back, "!" exits}`	Enter **da80adm**.
`Administrator User Name [admin]` `{"<" goes back, "!" exits}`	Accept the default value.
`Administrator Password:`	Enter **web4dmin**.
`Retype Password:`	Enter **web4dmin**.
`Server Name [da-2.example.com]` `{"<" goes back, "!" exits}`	Accept the default value.
`HTTP Port [8080]` `{"<" goes back, "!" exits}`	Enter **1080**.

`Document Root Directory [/opt/SUNWwbsvr/ https-da-2.example.com/docs] {"<" goes back, "!" exits}`	Accept the default value.
`Start Administration Server [yes/no] {"<" goes back, "!" exits}`	Enter **no**.

`Ready To Install`
 	
`1. Install Now`
`2. Start Over`
`3. Exit Installation`
 	
`What would you like to do?`
	Enter **1**.

When installation is complete, the following message is displayed:

```
Installation Successful.
```

7 **(Optional) To verify that Web Server was installed with the non-root user, examine the file permissions.**

```
# cd /opt/SUNWwbsvr/admin-server
# ls -al

total 16
drwxr-xr-x   8 root     root         512 Jul 19 10:36 .
drwxr-xr-x  11 da80adm  staff        512 Jul 19 10:36 ..
drwxr-xr-x   2 root     root         512 Jul 19 10:36 bin
drwx------   2 da80adm  staff        512 Jul 19 10:36 config
drwx------   3 da80adm  staff        512 Jul 19 11:09 config-store
drwx------   3 da80adm  staff        512 Jul 19 10:40 generated
drwxr-xr-x   2 da80adm  staff        512 Jul 19 10:40 logs
drwx------   2 da80adm  staff        512 Jul 19 10:36 sessions
```

The appropriate files and directories are owned by `da80adm`.

8 **Start the Web Server administration server.**

```
# su da80adm
# cd /opt/SUNWwbsvr/admin-server/bin
# ./startserv
```

9 **(Optional) Verify that the non-root user was able to start Web Server.**

a. **Access** `https://da-2.example.com:8989` **from a web browser.**

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

User Name: **admin**

Password: **web4dmin**

The Web Server administration console opens.

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

10 **Log out of the** da–2 **host machine.**

7.2 Enabling Secure Communications Between the Web Server Instances and the Load Balancer

When a Web Server instance is created, it contains a default `http-listener` port. In the following sections, certificates are requested and installed, and a new `http-listener` port is created and enabled for secure communication with the OpenSSO Enterprise Load Balancer 3.

- "To Request and Install a Server Certificate and a Root Certificate for Web Server 1" on page 142
- "To Create an SSL Enabled HTTP Listener Port on Web Server 1" on page 144
- "To Request and Install a Server Certificate and a Root Certificate for Web Server 2" on page 147
- "To Create an SSL Enabled HTTP Listener Port on Web Server 2" on page 149
- "To Import the Root Certificate to the Web Server 1 JDK Certificate Store" on page 151
- "To Import the Root Certificate to the Web Server 2 JDK Certificate Store" on page 153

▼ To Request and Install a Server Certificate and a Root Certificate for Web Server 1

The `wadm` command line interface, bundled with the Web Server, is used to import the root and server certificates into the Web Server certificate store.

Before You Begin Copy the same root certificate imported in "4.3 Enabling Secure Communication for the Directory Server User Data Instances" on page 61 to the `da-1` host machine. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 38.

1 **As a root user, log in to the** da–1 **host machine.**

2 **Start the Web Server Administration Server.**

```
# su da80adm
# cd /opt/SUNWwbsvr/admin-server/bin
# ./startserv
```

3 **Create a temporary file that contains the administration password.**

This file will be used for certificate request generation and certificate installation

```
# cd /export/da80adm
# cat > admin.pwd

wadm_password=web4dmin

Hit Control D to terminate the command.

^D
```

4 **Generate a certificate signing request.**

```
# cd /opt/SUNWwbsvr/bin
# ./wadm create-cert-request --user=admin
--password-file=/export/da80adm/admin.pwd --host=da-1.example.com
--port=8989 --key-type=rsa --org="Sun Microsystems"
--org-unit="Sun Distributed Authentication"
--locality="Santa Clara" --state=California --country=US
--config=da-1.example.com --token=internal
--server-name=da-1.example.com
```

5 **Copy the output into a file named `da-1.csr` and send the request to the CA of your choice.**


```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIB2DCCAUECAQAwgZcxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlh
MRQwEgYDVQQHEwtTYW50YSBDbGFyYTEZMBcGA1UEChMQU3VuIE1pY3Jvc3lzdGVt
czEnMCUGA1UECxMeU3VuIERpc3RyaWJ1dGVkIEF1dGhlbnRpY2F0aW9uMRkwFwYD
VQQDExBkYS0xLmV4YW1wbGUuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDGdeNgE00/6o3nrG38yatMhnrJeUVR86Pj5rBk282DQQfVenuWt0hL8Y6q9KvT
JQRoeclWMl94ZErdtNY0qKqXZBxhC0CCtiAvNHJAg8zErGTOADs6ptmXkzVRGBXE
b7zLOGlROnK9xAw0wms/aFsbA/Mb0zMI5PDztRAf5A8fIQIDAQABoAAwDQYJKoZI
hvcNAQEFBQADgYEAqap+9N/T+pzzAZL+EiG3rciKcG+Ij94Yk+3q0hMj3d3xer8Q
1shLAy4za9qHvOnT8M7hpKY6lpw4Y4N+w3eIgfDc3aCnz1Aot5Na4alWJZ81SUAZ
Fl6fD7CX7KMtF6Agfpi5OV+NdOiBL6tQ7F7G70c3pYV5MnQvYf5dnuiZEkQ=
-----END NEW CERTIFICATE REQUEST-----
```


The CA issues and returns a certified server certificate named `da-1.cer`.

6 **Install `da-1.cer`, the server certificate.**

```
# ./wadm install-cert --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-1.example.com --port=8989
--token=internal --cert-type=server
--nickname=da-1 da-1.cer

CLI201 Command 'install-cert' ran successfully
```

7 **(Optional) Verify that the server certificate was properly installed.**

```
# ./wadm list-certs --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-1.example.com --token=internal
--cert-type=server

da-1
```

The output indicates that the server certificate was properly installed.

8 **Install `ca.cer`, the root certificate.**

```
# ./wadm install-cert --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-1.example.com --port=8989
--token=internal --cert-type=ca
--nickname=OpenSSLTestCA ca.cer

CLI201 Command 'install-cert' ran successfully
```

9 **(Optional) Verify that the root certificate was properly installed.**

```
# ./wadm list-certs --user=admin
--password-file=/export/da80adm/admin.pwd
--token=internal --cert-type=ca
--config=da-1.example.com | grep -i open

openSSLTestCA - sun
```

The output indicates that the root certificate was properly installed.

▼ To Create an SSL Enabled HTTP Listener Port on Web Server 1

The `wadm` command line interface, bundled with the Web Server, is used in this procedure.

Before You Begin This procedure assumes that you have just completed "To Request and Install a Server Certificate and a Root Certificate for Web Server 1" on page 142 and are still logged in as the non-root user.

1 **Create an SSL enabled HTTP listener port on Web Server 1.**

```
# ./wadm create-http-listener --user=admin
--password-file=/export/da80adm/admin.pwd
--host=da-1.example.com --port=8989
--listener-port=1443 --config=da-1.example.com
```

```
--server-name=da-1.example.com
--default-virtual-server-name=da-1.example.com
http-listener-2

CLI201 Command 'create-http-listener' ran successfully
```

2 **(Optional) Verify that the listener was created.**

```
# ./wadm get-ssl-prop --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-1.example.com
--http-listener=http-listener-2

tls=true
client-auth-timeout=60
client-auth=false
enabled=false
ssl2=false
max-client-auth-data=1048576
tls-rollback-detection=true
ssl3=true
```

The output indicates that the listener was properly created.

3 **Enable SSL for the newly created HTTP listener port.**

```
# ./wadm set-ssl-prop --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-1.example.com
--http-listener=http-listener-2
--enabled=true

CLI201 Command 'set-ssl-prop' ran successfully
```

4 **Associate the HTTP listener port with the nickname of the certificate.**

```
# ./wadm set-ssl-prop --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-1.example.com
--http-listener=http-listener-2
--server-cert-nickname=da-1

CLI201 Command 'set-ssl-prop' ran successfully
```

5 **(Optional) Verify that SSL is enabled on the listener port and is configured with an associated server certificate.**

```
# ./wadm get-ssl-prop --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-1.example.com
```

```
--http-listener=http-listener-2

tls=true
server-cert-nickname=da-1
client-auth-timeout=60
client-auth=false
enabled=true
ssl2=false
max-client-auth-data=1048576
tls-rollback-detection=true
ssl3=true
```

The output indicates that SSL is enabled and `da-1` is the associated certificate nickname.

6 **Deploy the modified configuration.**

```
# ./wadm deploy-config --user=admin
--password-file=/export/da80adm/admin.pwd
--host=da-1.example.com port=8989
da-1.example.com

CLI201 Command 'deploy-config' ran successfully
```

7 **Restart the Web Server instance.**

```
# cd /opt/SUNWwbsvr/https-da-1.example.com/bin
# ./stopserv ; ./startserv

server has been shutdown

Sun Java System Web Server 7.0U2 B12/09/2007 09:02
info: CORE5076: Using [Java HotSpot(TM) Server VM, Version 1.5.0_12]
from [Sun Microsystems Inc.]
info: HTTP3072: http-listener-1: http://da-1.example.com:1080 ready to
accept requests
info: HTTP3072: http-listener-2: https://da-1.example.com:1443 ready to
accept requests
info: CORE3274: successful server startup
```

The output indicates that `http-listener-2` is SSL is enabled and ready to accept requests.

8 **Remove the temporary administration password file.**

```
# cd /export/da80adm
# rm admin.pwd
```

9 **(Optional) Access `https://da-1.example.com:1443` from a web browser to verify that the secure port can be invoked.**

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.

▼ To Request and Install a Server Certificate and a Root Certificate for Web Server 2

The `wadm` command line interface, bundled with the Web Server, is used to import the root and server certificates into the Web Server certificate store.

Before You Begin Copy the same root certificate imported in "4.3 Enabling Secure Communication for the Directory Server User Data Instances" on page 61 to the `da-1` host machine. For more information, see "3.3 Obtaining Secure Socket Layer Certificates" on page 38.

1. **As a root user, log in to the `da–2` host machine.**

2. **Start the Web Server Administration Server.**

```
# su da80adm
# cd /opt/SUNWwbsvr/admin-server/bin
# ./startserv
```

3. **Create a temporary file that contains the administration password.**

   This file will be used for certificate request generation and certificate installation

```
# cd /export/da80adm
# cat > admin.pwd

wadm_password=web4dmin

Hit Control D to terminate the command.

^D
```

4. **Generate a certificate signing request.**

```
# cd /opt/SUNWwbsvr/bin
# ./wadm create-cert-request --user=admin
--password-file=/export/da80adm/admin.pwd --host=da-2.example.com
--port=8989 --key-type=rsa --org="Sun Microsystems"
--org-unit="Sun Distributed Authentication"
--locality="Santa Clara" --state=California --country=US
```

```
--config=da-2.example.com --token=internal
--server-name=da-2.example.com
```

5 **Copy the output into a file named** `da-2.csr` **and send the request to the CA of your choice.**

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIB2DCCAUECAQAwgZcxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlh
MRQwEgYDVQQHEwtTYW50YSBDbGFyYTEZMBcGA1UEChMQU3VuIE1pY3Jvc3lzdGVt
czEnMCUGA1UECxMeU3VuIERpc3RyaWJ1dGVkIEF1dGhlbnRpY2F0aW9uMRkwFwYD
VQQDExBkYS0xLmV4YW1wbGUuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDGdeNgE00/6o3nrG38yatMhnrJeUVR86Pj5rBk282DQQfVenuWt0hL8Y6q9KvT
JQRoeclWMl94ZErdtNY0qKqXZBxhC0CCtiAvNHJAg8zErGTOADs6ptmXkzVRGBXE
b7zLOGlROnK9xAw0wms/aFsbA/Mb0zMI5PDztRAf5A8fIQIDAQABoAAwDQYJKoZI
hvcNAQEFBQADgYEAqap+9N/T+pzzAZL+EiG3rciKcG+Ij94Yk+3q0hMj3d3xer8Q
1shLAy4za9qHvOnT8M7hpKY6lpw4Y4N+w3eIgfDc3aCnz1Aot5Na4alWJZ81SUAZ
Fl6fD7CX7KMtF6Agfpi5OV+NdOiBL6tQ7F7G70c3pYV5MnQvYf5dnuiZEkQ=
-----END NEW CERTIFICATE REQUEST-----
```

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

6 **Install** `da-2.cer`**, the server certificate.**

```
# ./wadm install-cert --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-2.example.com --port=8989
--token=internal --cert-type=server
--nickname=da-2 da-2.cer

CLI201 Command 'install-cert' ran successfully
```

7 **(Optional) Verify that the server certificate was properly installed.**

```
# ./wadm list-certs --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-2.example.com --token=internal
--cert-type=server

da-2
```

The output indicates that the server certificate was properly installed.

8 **Install** `ca.cer`**, the root certificate.**

```
# ./wadm install-cert --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-2.example.com --port=8989
--token=internal --cert-type=ca
--nickname=OpenSSLTestCA ca.cer

CLI201 Command 'install-cert' ran successfully
```

9 **(Optional) Verify that the certificate was properly installed.**

```
# ./wadm list-certs --user=admin
--password-file=/export/da80adm/admin.pwd
 --token=internal --cert-type=ca
--config=da-2.example.com | grep -i open

openSSLTestCA - sun
```

The output indicates that the root certificate was properly installed.

▼ To Create an SSL Enabled HTTP Listener Port on Web Server 2

The `wadm` command line interface, bundled with the Web Server, is used in this procedure.

Before You Begin This procedure assumes that you have just completed "To Request and Install a Server Certificate and a Root Certificate for Web Server 2" on page 147 and are still logged in as the non-root user.

1 **Create an SSL enabled HTTP listener port on Web Server 2.**

```
# ./wadm create-http-listener --user=admin
--password-file=/export/da80adm/admin.pwd
--host=da-2.example.com --port=8989
--listener-port=1443 --config=da-2.example.com
--server-name=da-2.example.com
--default-virtual-server-name=da-2.example.com
http-listener-2

CLI201 Command 'create-http-listener' ran successfully
```

2 **(Optional) Verify that the listener was created.**

```
# ./wadm get-ssl-prop --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-2.example.com
--http-listener=http-listener-2

tls=true
client-auth-timeout=60
client-auth=false
enabled=false
ssl2=false
max-client-auth-data=1048576
```

```
tls-rollback-detection=true
ssl3=true
```

The output indicates that the listener was properly created.

3 **Enable SSL for the newly created HTTP listener port.**

```
# ./wadm set-ssl-prop --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-2.example.com
--http-listener=http-listener-2
--enabled=true

CLI201 Command 'set-ssl-prop' ran successfully
```

4 **Associate the HTTP listener port with the nickname of the certificate.**

```
# ./wadm set-ssl-prop --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-2.example.com
--http-listener=http-listener-2
--server-cert-nickname=da-2

CLI201 Command 'set-ssl-prop' ran successfully
```

5 **(Optional) Verify that SSL is enabled on the listener port and is associated with the server certificate.**

```
# ./wadm get-ssl-prop --user=admin
--password-file=/export/da80adm/admin.pwd
--config=da-2.example.com
--http-listener=http-listener-2

tls=true
server-cert-nickname=da-2
client-auth-timeout=60
client-auth=false
enabled=true
ssl2=false
max-client-auth-data=1048576
tls-rollback-detection=true
ssl3=true
```

The output indicates that SSL is enabled and `da-2` is the associated certificate nickname.

6 **Deploy the modified configuration.**

```
# ./wadm deploy-config --user=admin
--password-file=/export/da80adm/admin.pwd
--host=da-2.example.com port=8989
```

```
da-2.example.com

CLI201 Command 'deploy-config' ran successfully
```

7 **Restart the Web Server instance.**

```
# cd /opt/SUNWwbsvr/https-da-2.example.com/bin
# ./stopserv ; ./startserv

server has been shutdown

Sun Java System Web Server 7.0U2 B12/09/2008 09:02
info: CORE5076: Using [Java HotSpot(TM) Server VM, Version 1.5.0_12]
from [Sun Microsystems Inc.]
info: HTTP3072: http-listener-1: http://da-2.example.com:1080 ready to
accept requests
info: HTTP3072: http-listener-2: https://da-2.example.com:1443 ready to
accept requests
info: CORE3274: successful server startup
```

The output indicates that `http-listener-2` is SSL is enabled and ready to accept requests.

8 **Remove the temporary administration password file.**

```
# cd /export/da80adm
# rm admin.pwd
```

9 **(Optional) Access `https://da-2.example.com:1443` from a web browser to verify that the secure port can be invoked.**

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.

▼ To Import the Root Certificate to the Web Server 1 JDK Certificate Store

Before You Begin Copy `ca.cer`, the same CA root certificate used in "4.3 Enabling Secure Communication for the Directory Server User Data Instances" on page 61, to the JDK certificate store in the `/export/WS7` directory on the `da–1` host machine.

1 **As a root user, log into the `da–1` host machine.**

2 **Import `ca.cer` into `cacerts`, the certificate store.**

```
# /opt/SUNWwbsvr/jdk/jre/bin/keytool -import
-trustcacerts -alias OpenSSLTestCA -file /export/WS7/ca.cer
-keystore /opt/SUNWwbsvr/jdk/jre/lib/security/cacerts
-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 2008 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 **(Optional) Verify that the root certificate was successfully imported.**

```
# /opt/SUNWwbsvr/jdk/jre/bin/keytool -list
-keystore /opt/SUNWwbsvr/jdk/jre/lib/security/cacerts
-storepass changeit | grep -i open

openssltestca, Jul 1, 2008, trustedCertEntry
```

4 **Restart the Web Server instance.**

```
# su da80adm
# cd /opt/SUNWwbsvr/https-da-1.example.com/bin
# ./stopserv ; ./startserv

server has been shutdown

Sun Java System Web Server 7.0U2 B12/09/2008 09:02
info: CORE5076: Using [Java HotSpot(TM) Server VM, Version 1.5.0_12]
from [Sun Microsystems Inc.]
info: HTTP3072: http-listener-1: http://da-1.example.com:1080 ready to
accept requests
info: HTTP3072: http-listener-2: https://da-1.example.com:1443 ready to
accept requests
info: CORE3274: successful server startup
```

5 **Log out of the `da-1` host machine.**

▼ To Import the Root Certificate to the Web Server 2 JDK Certificate Store

Before You Begin Copy `ca.cer`, the same CA root certificate used in "4.3 Enabling Secure Communication for the Directory Server User Data Instances" on page 61, to the JDK certificate store in the `/export/WS7` directory on the `da–2` host machine.

1. **As a root user, log into the `da–2` host machine.**

2. **Import `ca.cer` into `cacerts`, the certificate store.**

```
# /opt/SUNWwbsvr/jdk/jre/bin/keytool -import
-trustcacerts -alias OpenSSLTestCA -file /export/WS7/ca.cer
-keystore /opt/SUNWwbsvr/jdk/jre/lib/security/cacerts
-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 2008 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. **(Optional) Verify that the root certificate was successfully imported.**

```
# /opt/SUNWwbsvr/jdk/jre/bin/keytool -list
-keystore /opt/SUNWwbsvr/jdk/jre/lib/security/cacerts
-storepass changeit | grep -i open

openssltestca, Jul 1, 2008, trustedCertEntry
```

4. **Restart the Web Server instance.**

```
# su da80adm
# cd /opt/SUNWwbsvr/https-da-2.example.com/bin
# ./stopserv ; ./startserv

server has been shutdown
```

```
Sun Java System Web Server 7.0U2 B12/09/2008 09:02
info: CORE5076: Using [Java HotSpot(TM) Server VM, Version 1.5.0_12]
from [Sun Microsystems Inc.]
info: HTTP3072: http-listener-1: http://da-2.example.com:1080 ready to
accept requests
info: HTTP3072: http-listener-2: https://da-2.example.com:1443 ready to
accept requests
info: CORE3274: successful server startup
```

5 **Log out of the** `da-2` **host machine.**

7.3 Configuring the Distributed Authentication User Interface Load Balancer

The Distributed Authentication User Interface Load Balancer 3 sends the user and agent requests to the OpenSSO Enterprise server where the session originated. Secure Sockets Layer (SSL) is terminated and regenerated before a request is forwarded to the Distributed Authentication User Interface servers to allow the load balancer to inspect the traffic for proper routing. Load Balancer 3 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 3 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 Distributed Authentication User Interface 1 and Distributed Authentication User Interface 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 the Distributed Authentication User Interface Load Balancer" on page 155
2. "To Import a Root Certificate to the Distributed Authentication User Interface Load Balancer" on page 156
3. "To Import a Certificate to the Distributed Authentication User Interface Load Balancer" on page 157
4. "To Configure the Distributed Authentication User Interface Load Balancer" on page 158
5. "To Configure a Proxy for SSL Termination at the Distributed Authentication User Interface Load Balancer" on page 160

▼ To Request a Certificate for the Distributed Authentication User Interface Load Balancer

Generate a certificate signing request to send to a CA.

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

2. **Log in to the BIG-IP console using the following information.**

   User Name: *username*

   Password: *password*

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

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

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

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

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

   Key Identifier: **`lb-3.example.com`**

   Organizational Unit Name: **`Deployment`**

Domain Name:	**`lb-3.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 text file named** `lb-3.csr`.

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

11. **Send** `lb-3.csr` **to the CA of your choice.**

▼ To Import a Root Certificate to the Distributed Authentication User Interface Load Balancer

The CA root certificate proves that the particular CA did, in fact, issue a particular certificate. For this purpose, import the root certificate of the CA that issued the Load Balancer 3 server certificate into the Load Balancer 3 certificate store.

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 38.

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

2. **Log in using the following information:**

User name:	*username*
Password:	*password*

3. **In the left pane of the 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 the file that contains the CA root certificate 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 OpenSSL_CA_Cert is now included in the Certificate ID list.

▼ To Import a Certificate to the Distributed Authentication User Interface Load Balancer

Before You Begin This procedure assumes you have received a certificate from a CA, just completed "To Import a Root Certificate to the Distributed Authentication User Interface Load Balancer" on page 156, 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 `lb-3.example.com` is in the Key List. This was generated in "To Request a Certificate for the Distributed Authentication User Interface Load Balancer" on page 155.

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

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

5. **In the Choose File dialog, navigate to the file that contains the certificate text sent to you by the CA and click Open.**

6. **Click Install Certificate.**

7. **On the Certificate lb-3.example.com page, click Return to Certificate Administration Information.**

   Verify that the Certificate ID indicates `lb-3.example.com` on the SSL Certificate Administration page.

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

▼ To Configure the Distributed Authentication User Interface Load Balancer

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

2. **Log in using the following information.**

   User name: *username*

   Password: *password*

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	**`AuthenticationUI-Pool`**
Load Balancing Method	Round Robin
Resources	Add the IP address and port number of both Distributed Authentication User Interface host machines: `da-1:1443` and `da-2:1443`.

   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.

   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. **On the Virtual Servers tab, click Add.**

   c. **In the Add Virtual Server wizard, enter the virtual server IP address and port number.**

Address	Enter the IP address for `lb-3.example.com`
Service	**`9443`**

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

e. **In the Pool Selection dialog box, assign the `AuthenticationUI-Pool` Pool.**

f. **Click Done.**

6 **Add Monitors.**

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

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

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

c. **Add a TCP monitor to each Web Server node.**

In the Node list, locate the IP address and port number for `da-1:1443` and `da-2:1443`, and select the Add checkbox.

d. **Click Apply.**

7 **Configure the load balancer for persistence.**

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

b. **Click the `AuthenticationUI-Pool` link.**

c. **Click the Persistence tab.**

d. **Under Persistence Type, select Passive HTTP Cookie.**

e. **Under Cookie Name, enter `DistAuthLBCookie`.**

f. **Click Apply.**

8 **In the left frame, click BIGpipe.**

9 **In the BIGpipe command window, type `makecookie` *IP-address*`:`*port*.**

IP-address is the IP address of the `da-1` host machine and *port* is the same machine's port number; in this case, 1443.

10 **Press Enter to execute the command.**

Something similar to `Set-Cookie: BIGipServer[poolname]=4131721920.41733.0000; path=/` is displayed. Save the numbered value (in this case, 4131721920.41733.0000) for use in "To Configure Load Balancer Cookies for the Distributed Authentication User Interface" on page 173.

11 **In the left frame, click BIGpipe again.**

12 **In the BIGpipe command window, type `makecookie` *IP-address*`:`*port*.**

IP-address is the IP address of the `da-2` host machine and *port* is the same machine's port number; in this case, 1443.

13 **Press Enter to execute the command.**

Something similar to `Set-Cookie: BIGipServer[poolname]=4148499136.41733.0000; path=/` is displayed. Save the numbered value (in this case, 4148499136.41733.0000) for use in "To Configure Load Balancer Cookies for the Distributed Authentication User Interface" on page 173.

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

▼ To Configure a Proxy for SSL Termination at the Distributed Authentication User Interface Load Balancer

Secure communication is terminated and regenerated at the load balancer before forwarding a request to the Distributed Authentication User Interface.

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

2 **Log in using the following information:**

Username *username*

Password *password*

3 **Click *Configure your BIG-IP 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 SSL and ServerSSL.
Proxy Address:	The IP address of Load Balancer 3.
Proxy Service:	**`1443`** The secure port number
Destination Address:	The IP address of Load Balancer 3.
Destination Service:	**`9443`** The secure port number
Destination Target:	Choose **`Local Virtual Server`**.
SSL Certificate:	Choose **`lb-3.example.com`**.
SSL Key:	Choose **`lb-3.example.com`**.
Enable ARP:	Check this box.

7 **Click Next.**
The Insert HTTP Header String page is displayed.

8 **Choose Matching for Rewrite Redirects.**

9 **Click Next.**
The Client Cipher List String page is displayed.

10 **Accept the defaults and click Next.**
The Server Chain File page is displayed.

11 **Select OpenSSL_CA_Cert.crt from the drop-down list.**

12 **Click Done.**
The new proxy server is now added to the Proxy Server list.

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

14 **Access `https://lb-3.example.com:1443/index.html` from a web browser to verify the configuration.**

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.

---

15 **Close the browser.**

7.4 Creating an Agent Profile with Custom User for the Distributed Authentication User Interface

Before installing and configuring the Distributed Authentication User Interface, create an agent profile with the OpenSSO Enterprise console. This agent profile allows OpenSSO Enterprise to store authentication and configuration information regarding the Distributed Authentication User Interface. The agent profile will be stored in the configuration data store.

Note – Although the Distributed Authentication User Interface is not an agent, it acts on behalf of OpenSSO Enterprise and therefore must have its own agent profile. This agent profile will be used by the Distributed Authentication User Interface to authenticate itself to OpenSSO Enterprise.

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

- "To Create an Agent Profile with Custom User for the Distributed Authentication User Interface" on page 162
- "To Verify that `authuiadmin` Was Created in Directory Server" on page 163

▼ To Create an Agent Profile with Custom User for the Distributed Authentication User Interface

The creation of the agent profile also creates a custom user that allows the Distributed Authentication User Interface to log into the OpenSSO Enterprise server. `authuiadmin` is the custom user created.

1 **Access** `https://osso-1.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 2.2 Agent tab.**

6 **Click New to create a new agent profile.**

The New Agent properties page is displayed.

7 **Type the following values and click Create.**

Name	**`authuiadmin`**
Password	**`authuiadmin`**
Password (confirm)	**`authuiadmin`**

`authuiadmin` is displayed in the list of Agent names.

8 **Log out of the console.**

To Verify that `authuiadmin` Was Created in Directory Server

This is an optional, verification step.

1 **Log in to either of the OpenSSO Enterprise host machines.**

2 **Run `ldapsearch` to verify that the `authuiadmin` entry was successfully created.**

```
# cd /var/opt/mps/serverroot/dsrk6/bin
# ./ldapsearch -b "dc=opensso,dc=java,dc=net" -h osso-1.example.com
-p 50389 -D "cn=Directory Manager" -w dsmanager "ou=authuiadmin"

version: 1
dn: ou=authuiadmin,ou=default,ou=OrganizationConfig,
ou=1.0,ou=AgentService,ou=services,dc=opensso,dc=java,dc=net
objectClass: top
objectClass: sunServiceComponent
sunserviceID: 2.2_Agent
ou: authuiadmin
sunKeyValue: userpassword=AQICrLO+CuXkZFllnTO/ISfA5UjKea1
 yVhgLpDj5QtqeiR/gWRF6w45Blh+hBjQfly7u
sunKeyValue: sunIdentityServerDeviceStatus=Active
```

```
sunKeyValue: sunIdentityServerDeviceKeyValue=
sunKeyValue: description=
sunsmspriority: 0
```

3 **Log out of the OpenSSO Enterprise host machine.**

4 **Access** `https://osso-1.example.com:1081/opensso/UI/Login` **from a web browser.**

5 **Log in to the OpenSSO Enterprise console as the agent user.**

User Name: **`authuiadmin`**

Password: **`authuiadmin`**

A successful login indicates that the Distributed Authentication User Interface will be successful in authentication during the configuration process.

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

7.5 Generating and Deploying the Distributed Authentication User Interface WAR

Use the following list of procedures as a checklist to create and deploy the Distributed Authentication User Interface WAR on both host machines.

1. "To Generate the Distributed Authentication User Interface WAR" on page 164
2. "To Deploy the Generated WAR as Distributed Authentication User Interface 1" on page 165
3. "To Configure Distributed Authentication User Interface 1" on page 168
4. "To Deploy the Generated WAR as Distributed Authentication User Interface 2" on page 169
5. "To Configure Distributed Authentication User Interface 2" on page 171
6. "To Configure Load Balancer Cookies for the Distributed Authentication User Interface" on page 173
7. "To Verify That Authentication Using the Distributed Authentication User Interface Load Balancer is Successful" on page 175

▼ To Generate the Distributed Authentication User Interface WAR

Create a WAR named `ossodistauth.war` that will be used to deploy the Distributed Authentication User Interface.

1. **As a root user, log in to the `osso–1` host machine.**

2. **Create a directory to serve as the staging area for the WAR.**

```
# cd /export/OSSO_BITS/opensso
# mkdir war-staging
# cd war-staging
```

3. **Extract the contents of `opensso.war` into the `war-staging` directory.**

```
# jar xvf /export/OSSO_BITS/opensso/deployable-war/opensso.war
```

4. **Generate the WAR using the Distributed Authentication User Interface file list.**

   `osso-distauth.list` is included with the OpenSSO Enterprise download.

```
# jar cvf /export/OSSO_BITS/opensso/deployable-war/ossodistauth.war
 @/export/OSSO_BITS/opensso/deployable-war/osso-distauth.list
```

5. **Update the generated WAR with additional files in the `/opensso/deployable-war/distauth` directory of the unzipped download.**

   See the README for more information.

```
# cd /export/OSSO_BITS/opensso/deployable-war/distauth
# jar uvf /export/OSSO_BITS/opensso/deployable-war/ossodistauth.war
```

   The WAR is updated and ready to be used to deploy the Distributed Authentication User Interface.

6. **Log out of the `osso–1` host machine.**

▼ To Deploy the Generated WAR as Distributed Authentication User Interface 1

Before You Begin This procedure assumes you have completed "To Generate the Distributed Authentication User Interface WAR" on page 164.

1. **As a root user, log in to the `da–1` host machine.**

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

```
# su da80adm
```

3. **Change to the directory into which `ossodistauth.war` will be copied.**

```
# cd /export/da80adm
```

4 **Copy** `ossodistauth.war` **from the** `osso–1` **host machine.**

```
# ftp osso-1.example.com

Connected to osso-1.example.com
220 osso-1.example.com FTP server ready.

Name (osso-1.example.com:username):username

Password: password
     ...
Using binary mode to transfer files

ftp> cd /export/OSSO_BITS/opensso/deployable-war

CWD command successful

ftp> mget ossodistauth.war

mget ossodistauth.war? y

200 PORT command successful

ftp> bye
```

5 **Verify that** `ossodistauth.war` **was successfully copied and is owned by the non-root user.**

```
# ls -al

total 17630
drwxr-xr-x   3 da80adm  staff        512 Jun 30 15:20 .
drwxr-xr-x   6 root     sys          512 May 13 11:22 ..
-rw-r--r--   1 da80adm  staff        144 May 13 11:22 .profile
drwx------   3 da80adm  staff        512 May 13 14:55 .sunw
-rw-r--r--   1 da80adm  staff   10017728 Jun 30 15:20 ossodistauth.war
-rw-r--r--   1 da80adm  staff        136 May 13 11:22 local.cshrc
-rw-r--r--   1 da80adm  staff        157 May 13 11:22 local.login
-rw-r--r--   1 da80adm  staff        174 May 13 11:22 local.profile
```

6 **Start the Web Server Administration Server.**

```
# cd /opt/SUNWwbsvr/admin-server/bin
# ./startserv
```

7 **Add the Distributed Authentication User Interface WAR using the** `wadm` **command line interface.**

```
# cd /opt/SUNWwbsvr/bin
# ./wadm add-webapp --user=admin
--host=da-1.example.com --port=8989
--config=da-1.example.com --vs=da-1.example.com
```

```
--uri=/distAuth
/export/da80adm/ossodistauth.war

Please enter admin-user-password: web4dmin

Do you trust the above certificate? [y|n] y

CLI201 Command 'add-webapp' ran successfully
```

8 **Deploy the Distributed Authentication User Interface WAR using the `wadm` command line interface.**

```
# ./wadm deploy-config --user=admin
--host=da-1.example.com --port=8989
da-1.example.com

Please enter admin-user-password: web4dmin

CLI201 Command 'deploy-config' ran successfully
```

9 **Verify that the `distAuth` web application has been deployed.**

```
# cd /opt/SUNWwbsvr/https-da-1.example.com/web-app/da-1.example.com
# ls -al

total 6
drwxr-xr-x   4 da80adm  staff        512 Jun 30 15:40 .
drwxr-xr-x   3 da80adm  staff        512 Jun 30 15:40 ..
drwxr-xr-x   6 da80adm  staff        512 Jun 30 15:40 distAuth
```

10 **Restart the Web Server instance.**

```
# cd /opt/SUNWwbsvr/https-da-1.example.com/bin
# ./stopserv; ./startserv

server has been shutdown
Sun Java System Web Server 7.0U2 B12/09/2008 09:02
info: CORE5076: Using [Java HotSpot(TM) Server VM, Version 1.5.0_12]
from [Sun Microsystems Inc.]
info: WEB0100: Loading web module in virtual server [da-1.example.com]
at [/distAuth]
info: HTTP3072: http-listener-1: http://da-1.example.com:1080 ready to
accept requests
info: HTTP3072: http-listener-2: https://da-1.example.com:1443 ready to
accept requests
info: CORE3274: successful server startup
```

The output indicates that the `distAuth` web application has been successfully loaded.

▼ To Configure Distributed Authentication User Interface 1

1. **Access `http://da-1.example.com:1080/distAuth` from a web browser.**

   The Configurator page is displayed the first time the Distributed Authentication User Interface is accessed.

2. **Provide the following configuration information and click Configure.**

Server Protocol	**`https`**
Server Host	**`lb-2.example.com`**
Server Port	**`1081`**
Server Deployment URI	**`opensso`**
`distAuth` Server Protocol	**`http`**
`distAuth` Server Host	**`da-1.example.com`**
`distAuth` Server Port	**`1080`**
`distAuth` Server Deployment URI	**`/distAuth`**
`distAuth` Server Cookie Name	**`AMDistAuthCookie`**
Debug Directory	**`/export/da80adm/Debug`**
Debug level	**`error`**
Encryption Key	Accept the default value.
Application User Name	**`authuiadmin`**
Application User Password	**`authuiadmin`**
Confirm Application User Password	**`authuiadmin`**

   These values will configure the Distributed Authentication User Interface web application to communicate with OpenSSO Enterprise through Load Balancer 2. You see the following message after a successful configuration.

```
DistAuth application is successfully configured.
AMDistAuthConfig.properties created at /export/da80adm/AMDistAuthConfig.properties

Click here to go to login page.
```

3 **Access** `http://da-1.example.com:1080/distAuth/UI/Login?goto=http://da-1.example.com:1080` **from a web browser.**

4 **Log in to the Distributed Authentication User Interface as** `testuser1`**.**

Username **`testuser1`**

Password **`password`**

After successful authentication, you should be redirected to the index page for the Web Server instance in which the Distributed Authentication User Interface is deployed. This confirms that the Distributed Authentication User Interface has authenticated to OpenSSO Enterprise using the load balancer's secure channel.

Caution – You may click the login link after configuration of the Distributed Authentication User Interface. If you do and provide valid administrator credentials you will get an error page indicating that the requested object does not exist on this server. This is because the success login URL configured on OpenSSO Enterprise is a relative URL.

▼ To Deploy the Generated WAR as Distributed Authentication User Interface 2

Before You Begin This procedure assumes you have completed "To Generate the Distributed Authentication User Interface WAR" on page 164.

1 **As a root user, log in to the** `da–2` **host machine.**

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

```
# su da80adm
```

3 **Change to the directory into which** `ossodistauth.war` **will be copied.**

```
# cd /export/da80adm
```

4 **Copy** `ossodistauth.war` **from the** `osso–1` **host machine.**

```
# ftp osso-1.example.com

Connected to osso-1.example.com
220 osso-1.example.com FTP server ready.

Name (osso-1.example.com:username):username

Password: password
     ...
```

```
Using binary mode to transfer files

ftp> cd /export/OSSO_BITS/opensso/deployable-war

CWD command successful

ftp> mget ossodistauth.war

mget ossodistauth.war? y

200 PORT command successful

ftp> bye
```

5 **Verify that `ossodistauth.war` was successfully copied and is owned by the non-root user.**

```
# ls -al

total 17630
drwxr-xr-x   3 da80adm  staff        512 Jun 30 15:20 .
drwxr-xr-x   6 root     sys          512 May 13 11:22 ..
-rw-r--r--   1 da80adm  staff        144 May 13 11:22 .profile
drwx------   3 da80adm  staff        512 May 13 14:55 .sunw
-rw-r--r--   1 da80adm  staff   10017728 Jun 30 15:20 ossodistauth.war
-rw-r--r--   1 da80adm  staff        136 May 13 11:22 local.cshrc
-rw-r--r--   1 da80adm  staff        157 May 13 11:22 local.login
-rw-r--r--   1 da80adm  staff        174 May 13 11:22 local.profile
```

6 **Start the Web Server Administration Server.**

```
# cd /opt/SUNWwbsvr/admin-server/bin
# ./startserv
```

7 **Add the Distributed Authentication User Interface WAR using the `wadm` command line interface.**

```
# cd /opt/SUNWwbsvr/bin
# ./wadm add-webapp --user=admin
--host=da-2.example.com --port=8989
--config=da-2.example.com --vs=da-2.example.com
--uri=/distAuth
/export/da80adm/ossodistauth.war

Please enter admin-user-password: web4dmin

Do you trust the above certificate? [y|n] y

CLI201 Command 'add-webapp' ran successfully
```

8 **Deploy the Distributed Authentication User Interface WAR using the `wadm` command line interface.**

```
# ./wadm deploy-config --user=admin
--host=da-2.example.com --port=8989
da-2.example.com

Please enter admin-user-password: web4dmin

CLI201 Command 'deploy-config' ran successfully
```

9 **Verify that the `distAuth` web application has been deployed.**

```
# cd /opt/SUNWwbsvr/https-da-2.example.com/web-app/da-2.example.com
# ls -al

total 6
drwxr-xr-x   4 da80adm  staff        512 Jun 30 15:40 .
drwxr-xr-x   3 da80adm  staff        512 Jun 30 15:40 ..
drwxr-xr-x   6 da80adm  staff        512 Jun 30 15:40 distAuth
```

10 **Restart the Web Server instance.**

```
# cd /opt/SUNWwbsvr/https-da-2.example.com/bin
# ./stopserv; ./startserv

server has been shutdown
Sun Java System Web Server 7.0U2 B12/09/2008 09:02
info: CORE5076: Using [Java HotSpot(TM) Server VM, Version 1.5.0_12]
from [Sun Microsystems Inc.]
info: WEB0100: Loading web module in virtual server [da-2.example.com]
at [/distAuth]
info: HTTP3072: http-listener-1: http://da-2.example.com:1080 ready to
accept requests
info: HTTP3072: http-listener-2: https://da-2.example.com:1443 ready to
accept requests
info: CORE3274: successful server startup
```

The output indicates that the `distAuth` web application has been successfully loaded.

▼ To Configure Distributed Authentication User Interface 2

1 **Access `http://da-2.example.com:1080/distAuth` from a web browser.**

The Configurator page is displayed the first time the Distributed Authentication User Interface is accessed.

2 **Provide the following configuration information and click Configure.**

Server Protocol	**`https`**
Server Host	**`lb-2.example.com`**
Server Port	**`1081`**
Server Deployment URI	**`opensso`**
`distAuth` Server Protocol	**`http`**
`distAuth` Server Host	**`da-2.example.com`**
`distAuth` Server Port	**`1080`**
`distAuth` Server Deployment URI	**`/distAuth`**
`distAuth` Server Cookie Name	**`AMDistAuthCookie`**
Debug Directory	**`/export/da80adm/Debug`**
Debug level	**`error`**
Encryption Key	Accept the default value.
Application User Name	**`authuiadmin`**
Application User Password	**`authuiadmin`**
Confirm Application User Password	**`authuiadmin`**

These values will configure the Distributed Authentication User Interface web application to communicate with OpenSSO Enterprise through Load Balancer 2. You see the following message after a successful configuration.

```
DistAuth application is successfully configured.
AMDistAuthConfig.properties created at /export/da80adm/AMDistAuthConfig.properties

Click here to go to login page.
```

3 **Access** `http://da-2.example.com:1080/distAuth/UI/Login?goto=http://da-2.example.com:1080` **from a web browser.**

4 **Log in to the Distributed Authentication User Interface as** `testuser1`**.**

Username **`testuser1`**

Password **`password`**

After successful authentication, you should be redirected to the index page for the Web Server instance in which the Distributed Authentication User Interface is deployed. This confirms that the Distributed Authentication User Interface has authenticated to OpenSSO Enterprise using the load balancer's secure channel.

Caution – You may click the login link after configuration of the Distributed Authentication User Interface. If you do and provide valid administrator credentials you will get an error page indicating that the requested object does not exist on this server. This is because the success login URL configured on OpenSSO Enterprise is a relative URL.

To Configure Load Balancer Cookies for the Distributed Authentication User Interface

Access to the Distributed Authentication User Interface is through Load Balancer 3. In order to maintain server affinity, the Distributed Authentication User Interface needs to specify *sticky* cookies. Towards this end, `AMDistAuthConfig.properties` is modified on both Distributed Authentication User Interface host machines.

1. **As a root user, log in to the `da–1` host machine.**

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

   ```
   # su da80adm
   ```

3. **Change to the non-root user directory.**

   ```
   # cd /export/da80adm
   ```

4. **Modify `AMDistAuthConfig.properties` as follows.**

   - Uncomment the last two lines at the end of the file.
   - Set the following property values:
     - `com.iplanet.am.lbcookie.name=`**`DistAuthLBCookie`**
     - `com.iplanet.am.lbcookie.value=`**`4131721920.41733.0000`**

Note – Use the same cookie name for the value of the `com.iplanet.am.lbcookie.name` property that was specified for load balancer persistence in "To Configure the Distributed Authentication User Interface Load Balancer" on page 158. Failure to do so might cause the OpenSSO Enterprise login page to go into a loop since stickiness could not be maintained based on the cookie name.

---

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

6 **Restart the Web Server instance.**

```
# cd /opt/SUNWwbsvr/https-da-1.example.com/bin
# ./stopserv; ./startserv
```

7 **Log out of the** `da–1` **host machine.**

8 **As a root user, log in to the** `da–2` **host machine.**

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

```
# su da80adm
```

10 **Change to the non-root user directory.**

```
# cd /export/da80adm
```

11 **Modify** `AMDistAuthConfig.properties` **as follows.**

- Uncomment the last two lines at the end of the file.
- Set the following property values:
  - `com.iplanet.am.lbcookie.name=`**`DistAuthLBCookie`**
  - `com.iplanet.am.lbcookie.value=`**`4148499136.41733.0000`**

Note – Use the same cookie name for the value of the `com.iplanet.am.lbcookie.name` property that was specified for load balancer persistence in "To Configure the Distributed Authentication User Interface Load Balancer" on page 158. Failure to do so might cause the OpenSSO Enterprise login page to go into a loop since stickiness could not be maintained based on the cookie name.

---

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

13 **Restart the Web Server instance.**

```
# cd /opt/SUNWwbsvr/https-da-2.example.com/bin
# ./stopserv; ./startserv
```

14 **Log out of the** `da–2` **host machine.**

▼ To Verify That Authentication Using the Distributed Authentication User Interface Load Balancer is Successful

1 **Access the load balancer's secure port at** `https://lb-3.example.com:1443/distAuth/UI/Login?goto=https://lb-3.example.com:1443` **from a web browser.**

2 **Log in to the OpenSSO Enterprise console as** `testuser1`**.**

Username **`testuser1`**

Password **`password`**

After successful login, you should be redirected to the index page for one of the Web Server instances in which the Distributed Authentication User Interface is deployed. If the load balancer configuration is incorrect, the OpenSSO Enterprise login page would not have been displayed in the previous step.

CHAPTER 8

Configuring the Protected Resource Host Machines

Each machine on which the protected resources will be hosted contain 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 Load Balancer 2. This chapter contains the following sections:

- "8.1 Configuring the Protected Resource Host Machines with a J2EE Policy Agent" on page 177
- "8.2 Configuring the Protected Resource Host Machines with a Web Policy Agent" on page 226

8.1 Configuring the Protected Resource Host Machines with a J2EE Policy Agent

We will install BEA WebLogic Server and a J2EE policy agent on the Protected Resource 1 host machine (`pr-1`) and on the Protected Resource 2 host machine (`pr-2`). The policy agents are then configured to access Load Balancer 2. Use the following list of procedures as a checklist for completing the task.

1. "8.1.1 Installing and Configuring the J2EE Container and J2EE Policy Agent on Protected Resource 1" on page 178
2. "8.1.2 Installing and Configuring the J2EE Container and J2EE Policy Agent on Protected Resource 2" on page 194
3. "8.1.3 Creating Groups Using the OpenSSO Enterprise Console" on page 210
4. "8.1.4 Setting Up a Test for the J2EE Policy Agent 1" on page 212
5. "8.1.5 Setting Up a Test for the J2EE Policy Agent 2" on page 217
6. "8.1.6 Configuring the J2EE Policy Agents to Access the Distributed Authentication User Interface" on page 223

8.1.1 Installing and Configuring the J2EE Container and J2EE Policy Agent on Protected Resource 1

Download the BEA WebLogic Server bits to the `pr-1` host machine 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 as J2EE Container 1 on Protected Resource 1" on page 178
2. "To Configure BEA WebLogic Server as J2EE Container 1 on Protected Resource 1" on page 179
3. "To Import the Certificate Authority Root Certificate into Application Server 1" on page 183
4. "To Install the J2EE Policy Agent 1 on Application Server 1" on page 184
5. "To Deploy the J2EE Policy Agent 1 Application" on page 189
6. "To Start the J2EE Policy Agent 1 Application" on page 190
7. "To Set Up the J2EE Policy Agent 1 Authentication Provider" on page 191
8. "To Deploy the J2EE Policy Agent 1 Sample Application" on page 192
9. "To Modify the J2EE Policy Agent 1 Configuration" on page 193

▼ To Install BEA WebLogic Server as J2EE Container 1 on Protected Resource 1

BEA WebLogic Server is the application server used as the J2EE container on Protected Resource 1.

1. **As a root user, log into the `pr-1` host machine.**

2. **Ensure that your system is properly patched.**

   Refer to the BEA web site to make sure that your system has the recommended patches.

3. **Create a directory into which you can download the WebLogic Server bits and change into it.**

```
# mkdir /export/BEAWL10
# cd /export/BEAWL10
```

4. **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
```

5. **Run the installer.**

```
# ./server100_solaris32.bin
```

6 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.

7 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 as J2EE Container 1 on Protected Resource 1

After installing the bits, WebLogic Server must be configured.

Before You Begin This procedure assumes you have just completed "To Install BEA WebLogic Server as J2EE Container 1 on Protected Resource 1" on page 178 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 **`pr-1`** 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: **`pr-1`** ■ Domain Location: **`/usr/local/bea/user_projects/domains`** (default)
`Creating Domain`	Click Done.

3 Start the WebLogic administration server.

```
# cd /usr/local/bea/user_projects/domains/pr-1
# ./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://pr-1.example.com:7001/console`.

---

5 **Change to the `AdminServer` directory.**

```
# cd /usr/local/bea/user_projects/domains/pr-1/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/pr-1/bin
# ./stopWebLogic.sh
# ./startWebLogic.sh
```

9 **Start the managed servers.**

```
# cd /usr/local/bea/user_projects/domains/pr-1/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/pr-1/
  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://pr-1.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 `pr–1` host machine.**

▼ To Import the Certificate Authority Root Certificate into Application Server 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 the OpenSSO Enterprise Load Balancer" on page 102 to the `/export/software` directory on the `pr-1` host machine.

1 **As a root user, log into the `pr–1` 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 `pr–1` host machine.**

To Install the J2EE Policy Agent 1 on Application Server 1

Before You Begin Set `JAVA_HOME` to `/usr/local/bea/jdk150_06`.

1 **As a root user, log into the `pr-1` host machine.**

2 **Stop the WebLogic Server 1 administration server and the WebLogic Server 1 managed instance.**

```
# cd /usr/local/bea/user_projects/domains/pr-1/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 **Create a text file that contains the Agent Administrator password.**

This text file should contain the password of the OpenSSO Enterprise administrator (by default, `amadmin`).

```
# cat > agentadm.pwd
```

```
ossoadmin
```

Hit Control D to terminate the command

```
^D
```

6 **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
```

7 **Unpack the J2EE policy agent bits.**

```
# unzip weblogic_v10_agent_3.zip
```

8 **Run the J2EE policy agent installer.**

```
# cd /export/J2EEPA1/j2ee_agents/weblogic_v10_agent/bin
# chmod 755 agentadmin
# ./agentadmin --custom-install
```

9 **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/pr-1/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://lb-2.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://pr-1.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 the path to a file that contains the password to be used for identifying the policy agent: **`/export/J2EEPA1/agent.pwd`**. **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.
`Enter the Agent Administrator's name:`	Enter **`amadmin`**
`Enter the path to the password file that contains the password of Agent Administrator:`	Enter **`/export/J2EEPA1/agentadm.pwd`**

```
-----------------------------------------------
SUMMARY OF YOUR RESPONSES
-----------------------------------------------
Startup script location :
  /usr/local/bea/user_projects/domains/
  pr-1/bin/startWebLogic.sh
WebLogic Server instance name :
  ApplicationServer-1
WebLogic home directory :
  /usr/local/bea/weblogic10
OpenSSO Server URL :
  https://lb-2.example.com:1081/opensso
Agent Installed on Portal domain : false
Agent URL :
  http://pr-1.example.com:1081/agentapp
Encryption Key :
  +Yr3K4K1/lWFe4H17SBHMNIUzLNRut7H
Agent Profile name : j2eeagent-1
Agent Profile Password file name :
  /export/J2EEPA1/agent.pwd
Agent Profile will be created right now
  by agent installer : true
Agent Administrator : amadmin
Agent Administrator's password file
  name : /export/J2EEPA1/agentadm.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`.

10 **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/pr-1/bin
```

b. **Insert the following line at the end of `setDomainEnv.sh`.**

```
. /usr/local/bea/user_projects/domains/pr-1/
bin/setAgentEnv_ApplicationServer-1.sh
```

c. **Save `setDomainEnv.sh` and close the file.**

11 **Change permissions for `setAgentEnv_ApplicationServer-1.sh`.**

```
# chmod 755 setAgentEnv_ApplicationServer-1.sh
```

12 **Start the WebLogic Server administration server and managed instance.**

```
# ./startWebLogic.sh &
# ./startManagedWebLogic.sh ApplicationSever-1 t3://localhost:7001
```

Watch for startup errors.

13 **Verify that the J2EE Policy Agent 1 was successfully created on the server using the following sub procedure.**

a. **Access** `https://osso-1.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.**

14 **Remove the password files.**

```
# cd /export/J2EEPA1
# rm agent.pwd
# rm agentadm.pwd
```

15 **Log out of the** `pr-1` **host machine.**

▼ To Deploy the J2EE Policy Agent 1 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://pr-1.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 `pr-1.example.com` link.**

7 **In the field named Location: `pr-1.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.**

▼ To Start the J2EE Policy Agent 1 Application

Before You Begin This procedure assumes that you have just completed "To Deploy the J2EE Policy Agent 1 Application" on page 189 and are still logged in to the WebLogic Server console as the administrator.

1 **In the WebLogic Server console, on the Settings for agentapp page, click Deployments.**

2 **On the Summary of Deployments page, mark the `agentapp` checkbox and click Start > Servicing all requests.**

3 **On the Start Application Assistant page, click Yes.**

Tip – If you encounter a JavaScript™ error, start the WebLogic Server instance and perform the steps again.

To Set Up the J2EE Policy Agent 1 Authentication Provider

Before You Begin This procedure assumes that you have just completed "To Start the J2EE Policy Agent 1 Application" on page 190 and are still logged in to the WebLogic Server console as the administrator.

1 **In the WebLogic Server console, on the Summary of Deployments page, under Domain Structure, click Security Realms.**

2 **On the Summary of Security Realms page, click Lock & Edit.**

3 **Click the `myrealm` link.**

4 **On the Settings for myrealm page, click the Providers tab.**

5 **Under Authentication Providers, click New.**

6 **On the Create a New Authentication Provider page, provide the following information and click OK.**

Name: **`Agent-1`**

Type: Select `AgentAuthenticator` from the drop down list.

`Agent-1` is now included in the list of Authentication Providers.

7 **In the list of Authentication Providers, click Agent-1.**

8 **In the Settings for Authentication Providers page, verify that the Control Flag is set to OPTIONAL.**

9 **In the navigation tree near the top of the page, click Providers.**

10 **In the list of Authentication Providers, click DefaultAuthenticator.**

11 **In the Settings for DefaultAuthenticator page, set the Control Flag to OPTIONAL and click Save.**

12 **In the navigation tree near the top of the page, click Providers again.**

13 **In the Change Center, click Activate Changes.**

14 **If indicated by the console, restart the servers with the following sub procedure.**

a. **Log out of the WebLogic Server console.**

b. **As a root user, log into the** `pr–1` **host machine.**

c. **Restart the administration server and the managed instance.**

```
# cd /usr/local/bea/user_projects/domains/pr-1/bin
# ./stopManagedWebLogic.sh ApplicationServer-1 t3://localhost:7001
# ./stopWebLogic.sh
# ./startWebLogic.sh
# ./startManagedWebLogic.sh ApplicationServer-1 t3://localhost:7001
```

d. **Log out of the** `pr–1` **host machine.**

▼ To Deploy the J2EE Policy Agent 1 Sample Application

1 **Access Application Server 1 at** `http://pr-1.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** `pr-1.example.com` **link.**

7 **In the list for Location:** `pr-1.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 Modify the J2EE Policy Agent 1 Configuration

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. In this deployment, J2EE policy agents are configured in centralized mode meaning that any configuration changes must be made using the OpenSSO Enterprise server. For more information, see "Centralized Agent Configuration" in *Sun OpenSSO Enterprise 8.0 Technical Overview*.

1 **Access `https://osso-1.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.**

8.1.2 Installing and Configuring the J2EE Container and J2EE Policy Agent on Protected Resource 2

Download the BEA WebLogic Server bits to the `pr-2` host machine 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 as J2EE Container 2 on Protected Resource 2" on page 194
2. "To Configure BEA WebLogic Server as J2EE Container 2 on Protected Resource 2" on page 196
3. "To Import the Certificate Authority Root Certificate into Application Server 2" on page 199
4. "To Install the J2EE Policy Agent 2 on Application Server 2" on page 200
5. "To Deploy the J2EE Policy Agent 2 Application" on page 205
6. "To Start the J2EE Policy Agent 2 Application" on page 206
7. "To Set Up the J2EE Policy Agent 2 Authentication Provider" on page 207
8. "To Deploy the J2EE Policy Agent 2 Sample Application" on page 208
9. "To Modify the J2EE Policy Agent 2 Configuration" on page 209

▼ To Install BEA WebLogic Server as J2EE Container 2 on Protected Resource 2

BEA WebLogic Server is the application server used as the J2EE container on Protected Resource 2.

1 **As a root user, log into the `pr-2` host machine.**

2 **Ensure that your system is properly patched.**

Refer to the BEA web site to make sure that your system has the recommended patches.

3 **Create a directory into which you can download the WebLogic Server bits and change into it.**

```
# mkdir /export/BEAWL10
# cd /export/BEAWL10
```

4 **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
```

5 **Run the installer.**

```
# ./server100_solaris32.bin
```

6 **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.

7 **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 as J2EE Container 2 on Protected Resource 2

After installing the bits, WebLogic Server must be configured.

Before You Begin This procedure assumes you have just completed "To Install BEA WebLogic Server as J2EE Container 2 on Protected Resource 2" on page 194 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-2`** ■ Listen Port: **`1081`**
`Configure Clusters`	Accept the default values and click Next.

`Configure Machines`	Select the Unix Machine tab, then select Add, type **`pr-2`** and click Next.
`Assign Servers to Machines`	From the left panel select *AdminServer* and *ApplicationServer-2*. From the right panel select *pr-2*. Click - -> and then click Next.
`Review WebLogic Domain`	Click Next.
`Create WebLogic Domain`	Add the following and click Create. ■ Domain name: **`pr-2`** ■ Domain Location: **`/usr/local/bea/user_projects/domains`** (default)
`Creating Domain`	Click Done.

3 Start the WebLogic administration server.

```
# cd /usr/local/bea/user_projects/domains/pr-2
# ./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://pr-2.example.com:7001/console`.

---

5 Change to the `AdminServer` directory.

```
# cd /usr/local/bea/user_projects/domains/pr-2/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 2 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/pr-2/bin
# ./stopWebLogic.sh
# ./startWebLogic.sh
```

9 **Start the managed servers.**

```
# cd /usr/local/bea/user_projects/domains/pr-2/bin
# ./startManagedWebLogic.sh ApplicationServer-2 t3://localhost:7001
```

You will be prompted for the administrative user credentials.

Username	**weblogic**
Password	**bea10admin**

10 **Change to the `ApplicationServer-2` directory.**

```
# cd /usr/local/bea/user_projects/domains/pr-2/
  servers/ApplicationServer-2
```

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 2 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-2/bin
# ./stopManagedWebLogic.sh ApplicationServer-2
   t3://localhost:7001
# ./startManagedWebLogic.sh ApplicationServer-2
   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://pr-2.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-2* are running and OK.

18 **Log out of the console.**

19 **Log out of the `pr–2` host machine.**

▼ To Import the Certificate Authority Root Certificate into Application Server 2

The 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 the OpenSSO Enterprise Load Balancer" on page 102 to the `/export/software` directory on the `pr-2` host machine.

1 **As a root user, log into the** `pr–2` **host machine.**

2 **Change to the directory where the** `cacerts` **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** `pr–2` **host machine.**

▼ To Install the J2EE Policy Agent 2 on Application Server 2

Before You Begin Set `JAVA_HOME` to `/usr/local/bea/jdk150_06`.

1 **As a root user, log into the** `pr-2` **host machine.**

2 **Stop the WebLogic Server 2 administration server and the WebLogic Server 2 managed server.**

```
# cd /usr/local/bea/user_projects/domains/pr-2/bin
# ./stopManagedWebLogic.sh ApplicationServer-2 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/J2EEPA2
# cd /export/J2EEPA2
```

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

j2eeagent2
```

Hit Control D to terminate the command

```
^D
```

5 **Create a text file that contains the Agent Administrator password.**

This text file should contain the password of the OpenSSO Enterprise administrator (by default, `amadmin`).

```
# cat > agentadm.pwd

ossoadmin
```

Hit Control D to terminate the command

```
^D
```

6 **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
```

7 **Unpack the J2EE policy agent bits.**

```
# unzip weblogic_v10_agent_3.zip
```

8 Run the J2EE policy agent installer.

```
# cd /export/J2EEPA2/j2ee_agents/weblogic_v10_agent/bin
# chmod 755 agentadmin
# ./agentadmin --custom-install
```

9 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/pr-2/bin/startwebLogic.sh`**
`Enter the WebLogic Server instance name: [AdminServer]`	Enter the name of the WebLogic Server instance secured by the agent **`ApplicationServer-2`**
`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://lb-2.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://pr-2.example.com:1081/agentapp`**
`Enter the Encryption Key [+Yr3K4K1/lWFe4H17SBHMNIUzLNRut7H]:`	Accept the default value.
`Enter the Agent Profile Name:`	**`j2eeagent-2`**
`Enter the path to the password file:`	Enter the path to a file that contains the password to be used for identifying the policy agent: **`/export/J2EEPA2/agent.pwd`**. **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.
`Enter the Agent Administrator's name:`	Enter **amadmin**
`Enter the path to the password file that contains the password of Agent Administrator:`	Enter **/export/J2EEPA2/agentadm.pwd**
`---------- SUMMARY OF YOUR RESPONSES ---------- Startup script location : /usr/local/bea/user_projects/domains/ pr-2/bin/startWebLogic.sh WebLogic Server instance name : ApplicationServer-2 WebLogic home directory : /usr/local/bea/weblogic10 OpenSSO Server URL : https://lb-2.example.com:1081/opensso Agent Installed on Portal domain : false Agent URL : http://pr-2.example.com:1081/agentapp Encryption Key : +Yr3K4K1/lWFe4H17SBHMNIUzLNRut7H Agent Profile name : j2eeagent-2 Agent Profile Password file name : /export/J2EEPA2/agent.pwd Agent Profile will be created right now by agent installer : true Agent Administrator : amadmin Agent Administrator's password file name : /export/J2EEPA2/agentadm.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/J2EEPA2/j2ee_agents/
  weblogic_v10_agent/Agent_001/
  config/FAMAgentBootstrap.properties
Agent Configuration file location
/export/J2EEPA2/j2ee_agents/
  weblogic_v10_agent/Agent_001/
  config/FAMAgentConfiguration.properties
Agent Audit directory location:
/export/J2EEPA2/j2ee_agents/
  weblogic_v10_agent/Agent_001/logs/audit
Agent Debug directory location:
/export/J2EEPA2/j2ee_agents/
  weblogic_v10_agent/Agent_001/logs/debug

Install log file location:
/export/J2EEPA2/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-2.sh`.

10 **Modify the startup script `setDomainEnv.sh` to reference `setAgentEnv_ApplicationServer-2.sh` using 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/pr-2/bin
```

b. **Insert the following line at the end of `setDomainEnv.sh`.**

```
. /usr/local/bea/user_projects/domains/pr-2/
bin/setAgentEnv_ApplicationServer-2.sh
```

c. **Save `setDomainEnv.sh` and close the file.**

11 **Change permissions for `setAgentEnv_ApplicationServer-2.sh`.**

```
# chmod 755 setAgentEnv_ApplicationServer-2.sh
```

12 **Start the WebLogic Server administration server and managed instance.**

```
# ./startWebLogic.sh &
# ./startManagedWebLogic.sh ApplicationSever-2 t3://localhost:7001
```

Watch for startup errors.

13 **Verify that the J2EE Policy Agent 2 was successfully created on the server using the following sub-procedure.**

a. **Access** `https://osso-1.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-2` is displayed under the Agent table.

f. **Click** `j2eeagent-2`**.**

The `j2eeagent-2` properties page is displayed.

g. **Log out of the OpenSSO Enterprise console and close the browser.**

14 **Remove the password files.**

```
# cd /export/J2EEPA2
# rm agent.pwd
# rm agentadm.pwd
```

15 **Log out of the** `pr-2` **host machine.**

▼ To Deploy the J2EE Policy Agent 2 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://pr-2.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 `pr-2.example.com` link.**

7 **In the field named Location: `pr-2.example.com`, click the root directory.**

8 **Navigate to `/export/J2EEPA2/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-2 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.**

▼ To Start the J2EE Policy Agent 2 Application

Before You Begin This procedure assumes that you have just completed "To Deploy the J2EE Policy Agent 2 Application" on page 205 and are still logged in to the WebLogic Server console as the administrator.

1 **In the WebLogic Server console, on the Settings for agentapp page, click Deployments.**

2 **On the Summary of Deployments page, mark the `agentapp` checkbox and click Start > Servicing all requests.**

3 **On the Start Application Assistant page, click Yes.**

Tip – If you encounter a JavaScript error, start the WebLogic Server instance and perform the steps again.

▼ To Set Up the J2EE Policy Agent 2 Authentication Provider

Before You Begin This procedure assumes that you have just completed "To Start the J2EE Policy Agent 2 Application" on page 206 and are still logged in to the WebLogic Server console as the administrator.

1 **In the WebLogic Server console, on the Summary of Deployments page, under Domain Structure, click Security Realms.**

2 **On the Summary of Security Realms page, click Lock & Edit.**

3 **Click the `myrealm` link.**

4 **On the Settings for myrealm page, click the Providers tab.**

5 **Under Authentication Providers, click New.**

6 **On the Create a New Authentication Provider page, provide the following information and click OK.**

Name: **`Agent-2`**

Type: Select `AgentAuthenticator` from the drop down list.

`Agent-2` is now included in the list of Authentication Providers.

7 **In the list of Authentication Providers, click Agent-2.**

8 **In the Settings for Authentication Providers page, verify that the Control Flag is set to OPTIONAL.**

9 **In the navigation tree near the top of the page, click Providers.**

10 **In the list of Authentication Providers, click DefaultAuthenticator.**

11 **In the Settings for DefaultAuthenticator page, set the Control Flag to OPTIONAL and click Save.**

12 **In the navigation tree near the top of the page, click Providers again.**

13 **In the Change Center, click Activate Changes.**

14 **If indicated by the console, restart the servers.**

a. **Log out of the WebLogic Server console.**

b. **As a root user, log into the** `pr–2` **host machine.**

c. **Restart the administration server and the managed instance.**

```
# cd /usr/local/bea/user_projects/domains/pr-2/bin
# ./stopManagedWebLogic.sh ApplicationServer-2 t3://localhost:7001
# ./stopWebLogic.sh
# ./startWebLogic.sh
# ./startManagedWebLogic.sh ApplicationServer-2 t3://localhost:7001
```

d. **Log out of the** `pr–2` **host machine.**

▼ To Deploy the J2EE Policy Agent 2 Sample Application

1 **Access Application Server 2 at** `http://pr-2.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** `pr-2.example.com` **link.**

7 **In the list for Location:** `pr-2.example.com`**, click the root directory.**

8 **Navigate to the application directory (**`/export/J2EEPA2/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-2` **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-2`.

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 2 console.**

▼ To Modify the J2EE Policy Agent 2 Configuration

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. In this deployment, J2EE policy agents are configured in centralized mode meaning that any configuration changes must be made using the OpenSSO Enterprise server. For more information, see "Centralized Agent Configuration" in *Sun OpenSSO Enterprise 8.0 Technical Overview*.

1 **Access `https://osso-1.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-2` is displayed under the Agent table.

6 **Click `j2eeagent-2`.**

The `j2eeagent-2` 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.**

8.1.3 Creating Groups Using the OpenSSO Enterprise Console

A *group* represents a collection of users with a common function, feature or interest. The groups created with this procedure will be replicated to OpenSSO Enterprise 2 and used in "8.1.4 Setting Up a Test for the J2EE Policy Agent 1" on page 212 and "8.1.5 Setting Up a Test for the J2EE Policy Agent 2" on page 217.

▼ To Create Manager and Employee Groups with OpenSSO Enterprise

1 **Access `https://osso-1.example.com:1081/opensso/console` from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

3 **Under the Access Control tab, click `/ (Top Level Realm)`.**

4 **Click the Subjects tab.**

5 **Click the Group tab.**

The Group page is displayed.

6. **Create a manager group using the following sub procedure.**

   a. **Click New on the Group page.**
      The New Group properties page is displayed.

   b. **Enter `Manager-Group` as the ID and click OK.**
      The Group page is displayed.

   c. **Click Manager-Group in the list.**

   d. **Click the User tab.**
      The test users are displayed.

   e. **Select Test User 1 from the list and click Add.**

   f. **Click Save.**

   g. **Click Back to Subjects.**

7. **Create an employee group using the following sub procedure.**

   a. **Click New on the Group page.**
      The New Group properties page is displayed.

   b. **Enter `Employee-Group` as the ID and click OK.**
      The Group page is displayed.

   c. **Click Employee-Group in the list.**

   d. **Click the User tab.**
      The test users are displayed.

   e. **Select Test User 2 from the list and click Add.**

   f. **Click Save.**

   g. **Click Back to Subjects.**

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

8.1.4 Setting Up a Test for the J2EE Policy Agent 1

The BEA Policy Agent comes with a sample application that was deployed in "To Deploy the J2EE Policy Agent 1 Sample Application" on page 192 and "To Deploy the J2EE Policy Agent 2 Sample Application" on page 208. The application was created to help test policies and will be used for that purpose in this section. Use the following list as a checklist for this task.

- "To Create a Test Policy in the OpenSSO Enterprise Root Realm" on page 212
- "To Configure OpenSSO Enterprise Properties for the J2EE Policy Agent 1 Sample Application" on page 213
- "To Verify that J2EE Policy Agent 1 is Configured Properly" on page 215

Note – For more information on the sample application, see `readme.txt` in the `/export/J2EEPA1/j2ee_agents/weblogic_v10_agent/sampleapp` directory.

▼ To Create a Test Policy in the OpenSSO Enterprise Root Realm

1. **Access `https://osso-1.example.com:1081/opensso/console` from a web browser.**

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

   Username **`amadmin`**

   Password **`ossoadmin`**

3. **Under the Access Control tab, click `/ (Top Level Realm)`.**

4. **Click the Policies tab.**

   The Policies page is displayed.

5. **Click New Policy.**

6. **Enter `URL Policy for Application Server-1` in the Name field.**

7. **Under Rules, click New.**

8. **On the resulting page, select URL Policy Agent (with Resource Name) and click Next.**

9. **On the resulting page, provide the following information and click Finish.**

   Name: **`agentsample`**

   Resource Name: **`http://pr-1.example.com:1081/agentsample/*`**

Note – Make sure the hostname is typed in lowercase.

GET	Mark this check box and verify that Allow is selected.
POST	Mark this check box and verify that Allow is selected.

The rule `agentsample` is now added to the list of Rules.

10 **Under Subjects, click New.**

11 **On the resulting page, select Access Manager Identity Subject and click Next.**

12 **On the resulting page, provide the following information and click Search.**

Name:	**`agentsampleGroup`**
Filter:	Select Group.

`Manager-Group` and `Employee-Group` are displayed in the Available list.

13 **Select Manager-Group and Employee-Group and click Add.**

`Manager-Group` and `Employee-Group` are displayed in the Selected list.

14 **Click Finish.**

15 **Click OK.**

The new policy is displayed in the list of policies.

16 **Click Back to Access Control.**

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

▼ To Configure OpenSSO Enterprise Properties for the J2EE Policy Agent 1 Sample Application

1 **Access `https://osso-1.example.com:1081/opensso/console` from a web browser.**

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

Username	**`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 Application tab.**

   The Application properties page is displayed.

8. **Provide the following information.**

Login Form URI:	Enter the following and click Add. **`/agentsample/authentication/login.html`**
Not Enforced URI Processing:	Enter each of the following and click Add. **`/agentsample/public/*`** **`/agentsample/images/*`** **`/agentsample/styles/*`** **`/agentsample/index.html`** **`/agentsample`**
Resource Access Denied URI:	Enter each of the following and click Add. Map Key: **`agentsample`** Corresponding Map Value: **`/agentsample/authentication/accessdenied.html`**

9. **Click Save.**

   The `j2eeagent-1` properties page is displayed.

10. **Map the attributes from the OpenSSO Enterprise embedded data store to those used by the Application Server with the following sub procedure.**

    a. **From the `j2eeagent-1` properties page, click Back to Main Page.**

    b. **Click the Subjects tab.**

c. **Click the Group tab.**

d. **Click Employee-Group in the list of Groups.**

e. **Copy and save** `id=Employee-Group,ou=group,dc=opensso,dc=java,dc=net`**, the value of the Universal ID attribute.**

f. **Click Back to Subjects.**

   You are returned to the Group tab.

g. **Click Manager-Group in the list of Groups.**

h. **Copy and save** `id=Manager-Group,ou=group,dc=opensso,dc=java,dc=net`**, the value of the Universal ID attribute.**

i. **Click Back to Subjects.**

j. **Click the Agents tab.**

k. **Click the J2EE tab.**

   `j2eeagent-1` is displayed under the Agent table.

l. **Click** `j2eeagent-1`**.**

   The `j2eeagent-1` properties page is displayed.

m. **Click the Application tab.**

   The Application properties page is displayed.

n. **Provide the identifiers previously saved as the manager and employee map keys and corresponding map values for Privileged Attribute Mapping and click Save.**

```
Map Key: [id=Manager-Group,ou=group,dc=opensso,dc=java,dc=net]
Corresponding Map Value: am_manager_role

Map Key: [id=Employee-Group,ou=group,dc=opensso,dc=java,dc=net]
Corresponding Map Value: am_employee_role
```

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

▼ To Verify that J2EE Policy Agent 1 is Configured Properly

Use these steps to access the agent sample application and test policies against it.

1. **Access `http://pr-1.example.com:1081/agentsample/index.html`, the sample application URL, from a web browser.**

   The Sample Application welcome page is displayed.

2. **Click the J2EE Declarative Security link.**

3. **On the resulting page, click Invoke the Protected Servlet.**

   You are redirected to the OpenSSO Enterprise login page.

4. **Log in to OpenSSO Enterprise as `testuser1`.**

   Username **`testuser1`**

   Password **`password`**

   If you can successfully log in as `testuser1` and the J2EE Policy Agent Sample Application page is displayed, the first part of the test has succeeded and authentication is working as expected.

5. **Click the J2EE Declarative Security link again.**

6. **On the resulting page, click Invoke the Protected Servlet.**

   If the Success Invocation message is displayed, the second part of the test has succeeded as the sample policy for the `manager` role has been enforced as expected.

7. **Click the J2EE Declarative Security link to return.**

8. **On the resulting page, click Invoke the Protected EJB via an Unprotected Servlet.**

   If the Failed Invocation message is displayed, the third part of the test has succeeded as the sample policy for the `employee` role has been enforced as expected.

9. **Close the browser.**

10. **In a new browser session, access `http://pr-1.example.com:1081/agentsample/index.html`, the sample application URL, again.**

    The Sample Application welcome page is displayed.

11. **Click the J2EE Declarative Security link.**

12. **On the resulting page, click Invoke the Protected EJB via an Unprotected Servlet.**

    You are redirected to the OpenSSO Enterprise login page.

13. **Log in to OpenSSO Enterprise as `testuser2`.**

    Username **`testuser2`**

Password **`password`**

Note – The Failed Invocation message is displayed. This is a known issue.

---

14 **Click the J2EE Declarative Security link.**

15 **On the resulting page, click Invoke the Protected EJB via an Unprotected Servlet.**

The Successful Invocation message is displayed as the sample policy for the `employee` role has been enforced as expected.

16 **Click the J2EE Declarative Security link to return.**

17 **On the resulting page, click Invoke the Protected Servlet.**

If the Access to Requested Resource Denied message is displayed, this part of the test has succeeded as the sample policy for the `manager` role has been enforced as expected.

18 **Close the browser.**

8.1.5 Setting Up a Test for the J2EE Policy Agent 2

The BEA Policy Agent comes with a sample application that was deployed in "To Deploy the J2EE Policy Agent 1 Sample Application" on page 192 and "To Deploy the J2EE Policy Agent 2 Sample Application" on page 208. The application was created to help test policies and will be used for that purpose in this section. Use the following list as a checklist for this task.

- "To Create a Test Policy in the OpenSSO Enterprise Root Realm " on page 217
- "To Configure OpenSSO Enterprise Properties for the J2EE Policy Agent 2 Sample Application" on page 219
- "To Verify that J2EE Policy Agent 2 is Configured Properly" on page 221

Note – For more information on the sample application, see `readme.txt` in the `/export/J2EEPA2/j2ee_agents/weblogic_v10_agent/sampleapp` directory.

▼ To Create a Test Policy in the OpenSSO Enterprise Root Realm

1 **Access `https://osso-1.example.com:1081/opensso/console` from a web browser.**

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

Username **`amadmin`**

Password `ossoadmin`

3 **Under the Access Control tab, click `/ (Top Level Realm)`.**

4 **Click the Policies tab.**

The Policies page is displayed.

5 **Click New Policy.**

6 **Enter `URL Policy for Application Server-2` in the Name field.**

7 **Under Rules, click New.**

8 **On the resulting page, select URL Policy Agent (with Resource Name) and click Next.**

9 **On the resulting page, provide the following information and click Finish.**

Name: `agentsample`

Resource Name: `http://pr-2.example.com:1081/agentsample/*`

Note – Make sure the hostname is typed in lowercase.

GET Mark this check box and verify that Allow is selected.

POST Mark this check box and verify that Allow is selected.

The rule `agentsample` is now added to the list of Rules.

10 **Under Subjects, click New.**

11 **On the resulting page, select Access Manager Identity Subject and click Next.**

12 **On the resulting page, provide the following information and click Search.**

Name: `agentsampleGroup`

Filter: Select Group.

`Manager-Group` and `Employee-Group` are displayed in the Available list.

13 **Select Manager-Group and Employee-Group and click Add.**

`Manager-Group` and `Employee-Group` are displayed in the Selected list.

14 **Click Finish.**

15 **Click OK.**

The new policy is displayed in the list of policies.

16 **Click Back to Access Control.**

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

▼ To Configure OpenSSO Enterprise Properties for the J2EE Policy Agent 2 Sample Application

1 **Access** `https://osso-1.example.com:1081/opensso/console` **from a web browser.**

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

Username **`amadmin`**

Password **`ossoadmin`**

3 **Under the Access Control tab, click `/ (Top Level Realm)`.**

4 **Click the Agents tab.**

5 **Click the J2EE tab.**

`j2eeagent-2` is displayed under the Agent table.

6 **Click `j2eeagent-2`.**

The `j2eeagent-2` properties page is displayed.

7 **Click the Application tab.**

The Application properties page is displayed.

8 **Provide the following information.**

Login Form URI: Enter the following and click Add.

`/agentsample/authentication/login.html`

Not Enforced URI Processing: Enter each of the following and click Add.

`/agentsample/public/*`

`/agentsample/images/*`

`/agentsample/styles/*`

	`/agentsample/index.html` **`/agentsample`**
Resource Access Denied URI:	Enter each of the following and click Add. Map Key: **`agentsample`** Corresponding Map Value: **`/agentsample/authentication/accessdenied.html`**

9 **Click Save.**

The `j2eeagent-2` properties page is displayed.

10 **Map the attributes from the OpenSSO Enterprise embedded data store to those used by the Application Server with the following sub procedure.**

a. **From the `j2eeagent-2` properties page, click Back to Main Page.**

b. **Click the Subjects tab.**

c. **Click the Group tab.**

d. **Click Employee-Group in the list of Groups.**

e. **Copy and save `id=Employee-Group,ou=group,dc=opensso,dc=java,dc=net`, the value of the Universal ID attribute.**

f. **Click Back to Subjects.**

You are returned to the Group tab.

g. **Click Manager-Group in the list of Groups.**

h. **Copy and save `id=Manager-Group,ou=group,dc=opensso,dc=java,dc=net`, the value of the Universal ID attribute.**

i. **Click Back to Subjects.**

j. **Click the Agents tab.**

k. **Click the J2EE tab.**

`j2eeagent-2` is displayed under the Agent table.

l. **Click** `j2eeagent-2`**.**

The `j2eeagent-2` properties page is displayed.

m. **Click the Application tab.**

The Application properties page is displayed.

n. **Provide the identifiers previously saved as the manager and employee map keys and corresponding map values for Privileged Attribute Mapping and click Save.**

```
Map Key: [id=Manager-Group,ou=group,dc=opensso,dc=java,dc=net]
Corresponding Map Value: am_manager_role

Map Key: [id=Employee-Group,ou=group,dc=opensso,dc=java,dc=net]
Corresponding Map Value: am_employee_role
```

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

▼ To Verify that J2EE Policy Agent 2 is Configured Properly

Use these steps to access the agent sample application and test policies against it.

1 **Access** `http://pr-2.example.com:1081/agentsample/index.html`**, the sample application URL, from a web browser.**

The Sample Application welcome page is displayed.

2 **Click the J2EE Declarative Security link.**

3 **On the resulting page, click Invoke the Protected Servlet.**

You are redirected to the OpenSSO Enterprise login page.

4 **Log in to OpenSSO Enterprise as** `testuser1`**.**

Username **`testuser1`**

Password **`password`**

If you can successfully log in as `testuser1` and the J2EE Policy Agent Sample Application page is displayed, the first part of the test has succeeded and authentication is working as expected.

5 **Click the J2EE Declarative Security link again.**

6 **On the resulting page, click Invoke the Protected Servlet.**

If the Success Invocation message is displayed, the second part of the test has succeeded as the sample policy for the `manager` role has been enforced as expected.

7. **Click the J2EE Declarative Security link to return.**

8. **On the resulting page, click Invoke the Protected EJB via an Unprotected Servlet.**

   If the Failed Invocation message is displayed, the third part of the test has succeeded as the sample policy for the `employee` role has been enforced as expected.

9. **Close the browser.**

10. **In a new browser session, access `http://pr-2.example.com:1081/agentsample/index.html`, the sample application URL, again.**

    The Sample Application welcome page is displayed.

11. **Click the J2EE Declarative Security link.**

12. **On the resulting page, click Invoke the Protected EJB via an Unprotected Servlet.**

    You are redirected to the OpenSSO Enterprise login page.

13. **Log in to OpenSSO Enterprise as `testuser2`.**

Username	**`testuser2`**
Password	**`password`**

    Note – The Failed Invocation message is displayed. This is a known issue.

    ---

14. **Click the J2EE Declarative Security link.**

15. **On the resulting page, click Invoke the Protected EJB via an Unprotected Servlet.**

    The Successful Invocation message is displayed as the sample policy for the `employee` role has been enforced as expected.

16. **Click the J2EE Declarative Security link to return.**

17. **On the resulting page, click Invoke the Protected Servlet.**

    If the Access to Requested Resource Denied message is displayed, this part of the test has succeeded as the sample policy for the `manager` role has been enforced as expected.

18. **Close the browser.**

8.1.6 Configuring the J2EE Policy Agents to Access the Distributed Authentication User Interface

Configure the J2EE policy agent to point to the secure port of the Distributed Authentication User Interface Load Balancer 3. Use the following list as a checklist to complete this task.

1. "To Configure the J2EE Policy Agent 1 to Access the Distributed Authentication User Interface" on page 223
2. "To Configure the J2EE Policy Agent 2 to Access the Distributed Authentication User Interface" on page 224

▼ To Configure the J2EE Policy Agent 1 to Access the Distributed Authentication User Interface

1. **Access `https://osso-1.example.com:1081/opensso/console` from a web browser.**

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

   Username **`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 Services properties page is displayed.

8. **Make the following changes to the OpenSSO Login URL property value and click Save.**

   - Select `https://lb-2.example.com:1081/opensso/UI/Login` and click Remove.
   - Enter `https://lb-3.example.com:1443/distAuth/UI/Login` and click Add.

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

10 **Verify that the agent is configured properly using the following sub procedure.**

a. **Access** `http://pr-1.example.com:1081/agentsample/index.html`**, the sample application URL, from a web browser.**

The Sample Application Welcome page is displayed.

b. **Click the J2EE Declarative Security link.**

c. **On the resulting page, click Invoke the Protected Servlet.**

You are redirected to the Distributed Authentication User Interface at `https://lb-3.example.com:1443/distAuth/UI/Login`.

d. **(Optional) Double-click the gold lock in the lower left corner of the browser.**

In the Properties page, you see the certificate for `lb-3.example.com`.

e. **Log in to OpenSSO Enterprise as** `testuser1`**.**

Username **`testuser1`**

Password **`password`**

If you can successfully log in as `testuser1` and the J2EE Policy Agent Sample Application page is displayed, user authentication worked through the Distributed Authentication User Interface and the agent is configured properly.

f. **Close the browser.**

▼ To Configure the J2EE Policy Agent 2 to Access the Distributed Authentication User Interface

1 **Access** `https://osso-1.example.com:1081/opensso/console` **from a web browser.**

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

Username **`amadmin`**

Password **`ossoadmin`**

3 **Under the Access Control tab, click** **`/ (Top Level Realm)`.**

4 **Click the Agents tab.**

5 **Click the J2EE tab.**

`j2eeagent-2` is displayed under the Agent table.

6. **Click `j2eeagent-2`.**

   The `j2eeagent-2` properties page is displayed.

7. **Click the OpenSSO Services tab.**

   The Services properties page is displayed.

8. **Make the following changes to the OpenSSO Login URL value and click Save.**

   - Select `https://lb-2.example.com:1081/opensso/UI/Login` and click Remove.
   - Enter `https://lb-3.example.com:1443/distAuth/UI/Login` and click Add.

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

10. **Verify that the agent is configured properly using the following sub procedure.**

    a. **Access `http://pr-2.example.com:1081/agentsample/index.html`, the sample application URL, from a web browser.**

       The Sample Application Welcome page is displayed.

    b. **Click the J2EE Declarative Security link.**

    c. **On the resulting page, click Invoke the Protected Servlet.**

       You are redirected to the Distributed Authentication User Interface at `https://lb-3.example.com:1443/distAuth/UI/Login`.

    d. **(Optional) Double-click the gold lock in the lower left corner of the browser.**

       In the Properties page, you see the certificate for `lb-3.example.com`.

    e. **Log in to OpenSSO Enterprise as `testuser1`.**

       Username **`testuser1`**

       Password **`password`**

       If you can successfully log in as `testuser1` and the J2EE Policy Agent Sample Application page is displayed, user authentication worked through the Distributed Authentication User Interface and the agent is configured properly.

    f. **Close the browser.**

8.2 Configuring the Protected Resource Host Machines with a Web Policy Agent

We will install Sun Java System Web Server and a Web policy agent on the Protected Resource 1 host machine (`pr-1`) and on the Protected Resource 2 host machine (`pr-2`). The policy agents are then configured to access Load Balancer 2. Use the following list of procedures as a checklist for completing the task.

1. "8.2.1 Installing and Configuring the Web Container and Web Policy Agent on Protected Resource 1" on page 226
2. "8.2.2 Installing Web Server and a Web Policy Agent on Protected Resource 2" on page 238
3. "8.2.3 Configuring the Web Policy Agents to Access the Distributed Authentication User Interface" on page 250

8.2.1 Installing and Configuring the Web Container and Web Policy Agent on Protected Resource 1

Download the Sun Java System Web Server bits to the `pr-1` host machine 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.

1. "To Install and Configure Sun Java System Web Server as Web Container 1 on Protected Resource 1" on page 226
2. "To Import the Certificate Authority Root Certificate into Web Server 1" on page 231
3. "To Install and Configure Web Policy Agent 1 on Protected Resource 1" on page 232
4. "To Configure Policy for Web Policy Agent 1 on Protected Resource 1" on page 236
5. "To Verify that Web Policy Agent 1 is Working Properly" on page 237

▼ To Install and Configure Sun Java System Web Server as Web Container 1 on Protected Resource 1

Sun Java System Web Server is the web container used on the `pr-1` 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 `pr-1` host machine.

2 **Install the required patches if necessary.**

Patch results for your machines might be different.

a. **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.

b. **Make a directory for downloading the patch you need and change into it.**

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

c. **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.

---

d. **Unzip the patch file.**

```
# unzip 119963–08.zip
# unzip 120011–14.zip
```

e. **Run** `patchadd` **to install the patches.**

```
# patchadd /export/patches/119963–08
# patchadd /export/patches/120011–14
```

f. **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.

3 **Create a directory into which you can download the Web Server bits and change into it.**

```
# mkdir /export/WS7
# cd /export/WS7
```

4 **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.

5 **Unpack the Web Server package.**

```
# gunzip sjsws-7_0u3-solaris-sparc.tar.gz
# tar xvf sjsws-7_0u3-solaris-sparc.tar
```

6 **Run** `setup`**.**

```
# cd /export/WS7
# ./setup --console
```

7 **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 [pr-1.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 [pr-1.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-pr-1.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.
```

8 Start the Web Server administration server.

```
# cd /opt/SUNWwbsvr/admin-server/bin
# ./startserv
```

9 Run `netstat` to verify that the port is open and listening.

```
# netstat -an | grep 8989

*.8989               *.*                0      0 49152      0 LISTEN
```

10 **(Optional) Login to the Web Server administration console at** `https://pr-1.example.com:8989` **as the administrator.**

Username **admin**

Password **web4dmin**

You should see the Web Server administration console.

11 **(Optional) Log out of the Web Server console and close the browser.**

12 **Start the Protected Resource 1 Web Server instance.**

```
# cd /opt/SUNWwbsvr/https-pr-1.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://pr-1.example.com:1080 ready to
accept requests
info: CORE3274: successful server startup
```

13 **Run** `netstat` **to verify that the port is open and listening.**

```
# netstat -an | grep 1080

*.1080               *.*                0      0 49152      0 LISTEN
```

14 **(Optional) Access the Protected Resource 1 instance at** `http://pr-1.example.com:1080` **using a web browser.**

You should see the default Web Server index page.

15 **Log out of the** `pr–1` **host machine.**

▼ To Import the Certificate Authority Root Certificate into Web Server 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 the OpenSSO Enterprise Load Balancer" on page 102 to the `pr-1` 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** `pr-1` **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 `pr-1` host machine.**

▼ To Install and Configure Web Policy Agent 1 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 `pr–1` 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 **Create a text file that contains the Agent Administrator password.**

This text file should contain the OpenSSO Enterprise administrator (by default, `amadmin`) password. The Web policy agent installer requires this to create the agent profile on the server.

```
# cat > agentadm.pwd

ossoadmin
```

Hit Control D to terminate the command

```
^D
```

5 **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
```

6 **Unzip the downloaded file.**

```
# unzip sjsws_v70_SunOS_sparc_agent_3.zip
```

7 **Run the agent installer.**

```
# cd /export/WebPA1/web_agents/sjsws_agent/bin
# ./agentadmin --custom-install
```

8 **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-pr-1.example.com/config]:`	Type **/opt/SUNWwbsvr/ https-pr-1.example.com/config** and press Enter.

`Enter the OpenSSO Enterprise URL including the deployment URI (http://opensso.sample.com:58080/opensso)`	Type **`https://lb-2.example.com:1081/opensso`** and press Enter.
`Enter the Agent URL: (http://agent1.sample.com:1234)`	Type **`http://pr-1.example.com:1080`** and press Enter.
`Enter the Encryption Key[WSpf7aqc3AFIGvf2mCqvNBOsf44cDrfA]:`	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.
`This Agent Profile does not exist in OpenSSO Enterprise, will it be created by the installer? (Agent Administror's name and password are required) [true)`	Press Enter to accept the default and have the installer create the Agent Profile.
`Enter the Agent Administrator's name:`	Type **`amadmin`** and press Enter.
`Enter the path to the password file that contains the password of the Agent Administrator.`	Type **`/export/WebPA1/agentadm.pwd`** and press Enter.

``` ---------- SUMMARY OF YOUR RESPONSES ---------- Sun Java System Web Server Config Directory : /opt/SUNWwbsvr/https-pr-1.example.com/config OpenSSO Server URL : https://lb-2.example.com:1081/opensso Agent URL : http://pr-1.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.

9 Restart the Web Server 1 instance.

```
# cd /opt/SUNWwbsvr/https-pr-1.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://pr-1.example.com:1080 ready to
accept requests
info: CORE3274: successful server startup
```

10 Use the following sub-procedure to verify that the Web Policy Agent 1 was successfully created.

a. Access `https://osso-1.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.**

11 **Remove the password files.**

```
# cd /export/WebPA1
# rm agent.pwd
# rm agentadm.pwd
```

12 **Log out of the** `pr-2` **host machine.**

To Configure Policy for Web Policy Agent 1 on Protected Resource 1

Use the OpenSSO Enterprise console to configure policy for Web Policy Agent 1 that will be used to verify that the agent is working properly.

Note – You will add additional policies later when we add a load balancer in front of the Protected Resource 1 host machine.

---

1 **Access** `https://osso-1.example.com:1081/opensso/console` **from a web browser.**

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

Username **amadmin**

Password **ossoadmin**

3 **Under the Access Control tab, click / (Top Level Realm).**

4 **Click the Policies tab.**

5 **Click New Policy.**

6 **Enter URL Policy for Protected Resource 1 in the Name field.**

7 **Under Rules, click New.**

The Rules properties page is displayed.

8 **Select URL Policy Agent (with resource name) and click Next.**

9 **Provide the following information on the resulting page and click Finish.**

Name:	**`URL Rule for Protected Resource 1`**
Resource Name:	**`http://pr-1.example.com:1080/*`**
GET	Mark this check box and verify that Allow is selected.
POST	Mark this check box and verify that Allow is selected.

The rule URL Rule for Protected Resource 1 is added to the list of Rules.

10 **Under Subjects, click New.**

The Subjects properties page is displayed.

11 **Select Access Manager Identity Subject and click Next.**

12 **On the resulting page, provide the following information and click Search.**

Name:	**`Test Subject`**
Filter:	Choose User and click Search to display a list of available users.
Available:	From the available users, select `testuser1` and click Add.

13 **Click Finish.**

14 **Click OK.**

The new policy is included in the list of Policies.

15 **Click Back to Access Control.**

16 **Log out of the console.**

▼ To Verify that Web Policy Agent 1 is Working Properly

1 **Access `http://pr-1.example.com:1080/index.html` from a web browser.**

2 **Log in to OpenSSO Enterprise as `testuser1`.**

Username	**`testuser1`**
Password	**`password`**

You should see the default index page for Web Server 1 as `testuser1` was configured in the test policy to be allowed to access Protected Resource 1.

3 **Log out and close the browser.**

4 **Once again, access `http://pr-1.example.com:1080/index.html` from a web browser.**

Tip – If you are not redirected to the OpenSSO Enterprise login page for authentication, clear your browser's cache and cookies and try again.

---

5 **Log in to OpenSSO Enterprise as `testuser2`.**

Username **`testuser2`**

Password **`password`**

You should see the message, *You're not authorized to view this page*, (or *Your client is not allowed to access the requested object*) as `testuser2` was not included in the test policy that allows access to Protected Resource 1.

8.2.2 Installing Web Server and a Web Policy Agent on Protected Resource 2

Download the Sun Java System Web Server bits to the `pr-2` host machine 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.

1. "To Install Web Server as Web Container 2 on Protected Resource 2" on page 238
2. "To Import the Certificate Authority Root Certificate into Web Server 2" on page 243
3. "To Install and Configure Web Policy Agent 2 on Protected Resource 2" on page 244
4. "To Configure Policy for Web Policy Agent 2 on Protected Resource 2" on page 248
5. "To Verify that Web Policy Agent 2 is Working Properly" on page 249

▼ To Install Web Server as Web Container 2 on Protected Resource 2

Sun Java System Web Server is the web container used on the `pr-2` 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 the 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 `pr-2` host machine.**

2 **Install the required patches if necessary.**

Patch results for your machines might be different.

a. **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.

b. **Make a directory for downloading the patch you need and change into it.**

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

c. **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.

---

d. **Unzip the patch file.**

```
# unzip 119963–08.zip
# unzip 120011–14.zip
```

e. **Run** `patchadd` **to install the patches.**

```
# patchadd /export/patches/119963–08
# patchadd /export/patches/120011–14
```

f. **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.

3 **Create a directory into which you can download the Web Server bits and change into it.**

```
# mkdir /export/WS7
# cd /export/WS7
```

4 **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.

5 **Unpack the Web Server package.**

```
# gunzip sjsws-7_0u3-solaris-sparc.tar.gz
# tar xvf sjsws-7_0u3-solaris-sparc.tar
```

6 **Run** `setup`**.**

```
# cd /export/WS7
# ./setup --console
```

7 **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 [pr-2.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 [pr-2.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-pr-2.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.
```

8 Start the Web Server administration server.

```
# cd /opt/SUNWwbsvr/admin-server/bin
# ./startserv
```

9 Run `netstat` to verify that the port is open and listening.

```
# netstat -an | grep 8989

*.8989               *.*                0      0 49152      0 LISTEN
```

10 **(Optional) Login to the Web Server administration console at** `https://pr-2.example.com:8989` **as the administrator.**

Username **admin**

Password **web4dmin**

You should see the Web Server administration console.

11 **(Optional) Log out of the Web Server console and close the browser.**

12 **Start the Protected Resource 2 Web Server instance.**

```
# cd /opt/SUNWwbsvr/https-pr-2.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://pr-2.example.com:1080 ready to
accept requests
info: CORE3274: successful server startup
```

13 **Run** `netstat` **to verify that the port is open and listening.**

```
# netstat -an | grep 1080

*.1080               *.*                0      0 49152      0 LISTEN
```

14 **(Optional) Access the Protected Resource 2 instance at** `http://pr-2.example.com:1080` **using a web browser.**

You should see the default Web Server index page.

15 **Log out of the** `pr–2` **host machine.**

▼ To Import the Certificate Authority Root Certificate into Web Server 2

The web policy agent on Protected Resource 2 connects to OpenSSO Enterprise through Load Balancer 2. The load balancer is SSL-enabled, so the agent must be able to trust the load balancer SSL certificate to establish the SSL connection. For this reason, import the root certificate of the Certificate Authority (CA) that issued the Load Balancer 2 SSL server certificate into the policy agent certificate store.

Before You Begin

- Copy the same CA root certificate used in "To Install a CA Root Certificate to the OpenSSO Enterprise Load Balancer" on page 102 to the `pr-2` 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 `pr-2` host machine.**

2 **Import `ca.cer`, 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 `ca.cer` 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 `pr-2` host machine.**

▼ To Install and Configure Web Policy Agent 2 on Protected Resource 2

Before You Begin The `JAVA_HOME` environment variable should be set to `/opt/SUNWwbsvr/jdk/jre`.

1 **As a root user, log into the `pr–2` host machine.**

2 **Create a directory into which you can download the Web Server agent bits and change into it.**

```
# mkdir /export/WebPA2
# cd /export/WebPA2
```

3 **Create a text file that contains the Agent Profile password.**

The Web Policy Agent installer requires this for installation.

```
# cat > agent.pwd
```

```
webagent2
```

Hit Control D to terminate the command

```
^D
```

4 **Create a text file that contains the Agent Administrator password.**

This text file should contain the OpenSSO Enterprise administrator (by default, `amadmin`) password. The Web Policy Agent installer requires this to create the agent profile on the server.

```
# cat > agentadm.pwd

ossoadmin
```

Hit Control D to terminate the command

```
^D
```

5 **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
```

6 **Unzip the downloaded file.**

```
# unzip sjsws_v70_SunOS_sparc_agent_3.zip
```

7 **Run the agent installer.**

```
# cd /export/WebPA2/web_agents/sjsws_agent/bin
# ./agentadmin --custom-install
```

8 **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-pr-2.example.com/config]:`	Type **`/opt/SUNWwbsvr/ https-pr-2.example.com/config`** and press Enter.
`Enter the OpenSSO Enterprise URL including the deployment URI (http://opensso.sample.com:58080/opensso)`	Type **`https://lb-2.example.com:1081/opensso`** and press Enter.
`Enter the Agent URL: (http://agent2.sample.com:1234)`	Type **`http://pr-2.example.com:1080`** and press Enter.
`Enter the Encryption Key [WSpf7aqc3AFIGvf2mCqvNBOsf44cDrA3c]`	Accept the default value.
`Enter the Agent profile name [UrlAccessAgent]:`	Type **`webagent-2`** and press Enter.
`Enter the path to a file that contains the password to be used for identifying the Agent.`	Type **`/export/WebPA2/agent.pwd`** and press Enter. **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 Administror's name and password are required) [true)`	Press Enter to accept the default and have the installer create the Agent Profile.
`Enter the Agent Administrator's name:`	Type **`amadmin`** and press Enter.
`Enter the path to the password file that contains the password of the Agent Administrator.`	Type **`/export/WebPA2/agentadm.pwd`** and press Enter.

``` ---------- SUMMARY OF YOUR RESPONSES ---------- Sun Java System Web Server Config Directory : /opt/SUNWwbsvr/https-pr-2.example.com/config OpenSSO Server URL : https://lb-2.example.com:1081/opensso Agent URL : http://pr-2.example.com:1080 Encryption Key : WSpf7aqc3AFIGvf2mCqvNBOsf44cDrf3 Agent Profile name : webagent-2 Agent Profile Password file name : /export/WebPA2/agent.pwd Agent Profile will be created right now by agent installer : true Agent Administrator : amadmin Agent Administrator's password file name : /export/WebPA2/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.

9 Restart the Web Server 2 instance.

```
# cd /opt/SUNWwbsvr/https-pr-2.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://pr-2.example.com:1080 ready to
accept requests
info: CORE3274: successful server startup
```

10 Use the following sub-procedure to verify that the Web Policy Agent 2 was successfully created.

a. Access `https://osso-1.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-2` under the Agent table.

e. **Click** `webagent-2`**.**

The `webagent-2` properties page is displayed.

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

11 **Remove the password files.**

```
# cd /export/WebPA2
# rm agent.pwd
# rm agentadm.pwd
```

12 **Log out of the** `pr-2` **host machine.**

▼ To Configure Policy for Web Policy Agent 2 on Protected Resource 2

Use the OpenSSO Enterprise console to configure policy for Web Policy Agent 2 that will be used to verify that the agent is working properly.

Note – You will add additional policies later when we add a load balancer in front of the Protected Resource 2 host machine.

---

1 **Access** `https://osso-1.example.com:1081/opensso/console` **from a web browser.**

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

Username **amadmin**

Password **ossoadmin**

3 **Under the Access Control tab, click / (Top Level Realm).**

4 **Click the Policies tab.**

5 **Click New Policy.**

6 **Enter URL Policy for Protected Resource 2 in the Name field.**

7 **Under Rules, click New.**

The Rules properties page is displayed.

8 **Select URL Policy Agent (with resource name) and click Next.**

9 **Provide the following information on the resulting page and click Finish.**

Name:	**`URL Rule for Protected Resource 2`**
Resource Name:	**`http://pr-2.example.com:1080/*`**
GET	Mark this check box and verify that Allow is selected.
POST	Mark this check box and verify that Allow is selected.

The rule URL Rule for Protected Resource 2 is added to the list of Rules.

10 **Under Subjects, click New.**

The Subjects properties page is displayed.

11 **Select Access Manager Identity Subject and click Next.**

12 **On the resulting page, provide the following information and click Search.**

Name:	**`Test Subject`**
Filter:	Choose User and click Search to display a list of available users.
Available:	From the available users, select `testuser1` and click Add.

13 **Click Finish.**

14 **Click OK.**

The new policy is included in the list of Policies.

15 **Click Back to Access Control.**

16 **Log out of the console.**

▼ To Verify that Web Policy Agent 2 is Working Properly

1 **Access `http://pr-2.example.com:1080/index.html` from a web browser.**

2 **Log in to OpenSSO Enterprise as `testuser1`.**

Username	**`testuser1`**
Password	**`password`**

You should see the default index page for Web Server 2 as `testuser1` was configured in the test policy to be allowed to access Protected Resource 2.

3 **Log out and close the browser.**

4 **Once again, access `http://pr-2.example.com:1080/index.html` from a web browser.**

Tip – If you are not redirected to the OpenSSO Enterprise login page for authentication, clear your browser's cache and cookies and try again.

---

5 **Log in to OpenSSO Enterprise as `testuser2`.**

Username **`testuser2`**

Password **`password`**

You should see the message, *You're not authorized to view this page*, (or *Your client is not allowed to access the requested object*) as `testuser2` was not included in the test policy that allows access to Protected Resource 2.

8.2.3 Configuring the Web Policy Agents to Access the Distributed Authentication User Interface

Configure the web policy agents to point to the secure port of the Distributed Authentication User Interface Load Balancer 3. Use the following list of procedures as a checklist to complete the task.

1. "To Configure the Web Policy Agent 1 to Access the Distributed Authentication User Interface" on page 250
2. "To Configure the Web Policy Agent 2 to Access the Distributed Authentication User Interface" on page 251

▼ To Configure the Web Policy Agent 1 to Access the Distributed Authentication User Interface

1 **Access `https://osso-1.example.com:1081/opensso/console` from a web browser.**

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

Username **`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 Services properties page is displayed.

8 **Make the following changes to the OpenSSO Login URL value and click Save.**

- Select `https://lb-2.example.com:1081/opensso/UI/Login` and click Remove.
- Enter `https://lb-3.example.com:1443/distAuth/UI/Login` and click Add.

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

10 **Verify that the agent is configured properly using the following sub procedure.**

a. **Access** `http://pr-1.example.com:1080/index.html` **from a web browser.**

You are redirected to the Distributed Authentication User Interface at `https://lb-3.example.com:1443/distAuth/UI/Login`.

b. **(Optional) Double-click the gold lock in the lower left corner of the browser.**

In the Properties page, you see the certificate for `lb-3.example.com`.

c. **Log in to OpenSSO Enterprise as** `testuser1`**.**

Username **`testuser1`**

Password **`password`**

The default index page for Web Server 1 is displayed as `testuser1` is defined in the test policy as having permission to access Protected Resource 1.

d. **Close the browser.**

▼ To Configure the Web Policy Agent 2 to Access the Distributed Authentication User Interface

1 **Access** `https://osso-1.example.com:1081/opensso/console` **from a web browser.**

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

Username **amadmin**

Password **ossoadmin**

3 **Under the Access Control tab, click `/ (Top Level Realm)`.**

4 **Click the Agents tab.**

5 **Click the Web tab.**

`webagent-2` is displayed under the Agent table.

6 **Click `webagent-2`.**

The `webagent-2` properties page is displayed.

7 **Click the OpenSSO Services tab.**

The Services properties page is displayed.

8 **Make the following changes to the OpenSSO Login URL value and click Save.**

- Select `[0]=https://lb-2.example.com:1081/opensso/UI/Login` and click Remove.
- Enter `[0]=https://lb-3.example.com:1443/distAuth/UI/Login` and click Add.

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

10 **Verify that the agent is configured properly using the following sub procedure.**

a. **Access `http://pr-2.example.com:1080/index.html` from a web browser.**

You are redirected to the Distributed Authentication User Interface at `https://lb-3.example.com:1443/distAuth/UI/Login`.

b. **(Optional) Double-click the gold lock in the lower left corner of the browser.**

In the Properties page, you see the certificate for `lb-3.example.com`.

c. **Log in to OpenSSO Enterprise as `testuser1`.**

Username **testuser1**

Password **password**

The default index page for Web Server 2 is displayed as `testuser1` is defined in the test policy as having permission to access Protected Resource 2.

d. **Close the browser.**

CHAPTER 9

Setting Up Load Balancers for the Policy Agents

Two load balancers are configured for the policy agents in this deployment example. Load Balancer 4 balances traffic passing through the web policy agents. Load Balancer 5 balances traffic passing through the J2EE policy agents. Both load balancers are configured for *simple persistence*. Simple persistence guarantees that requests from the same user session will always be sent to the same policy agent that initially validated the user session and evaluated the applicable policies. This chapter contains the following sections.

- "9.1 Configuring the Web Policy Agents Load Balancer" on page 253
- "9.2 Configuring the J2EE Policy Agents Load Balancer" on page 262

9.1 Configuring the Web Policy Agents Load Balancer

Load Balancer 4 handles traffic for the web policy agents, and is configured for simple persistence.

Note – From a performance perspective, each policy agent validates user sessions and evaluates applicable policies. The results of those actions are cached by the policy agent that performed them. If simple persistence is **not** set, each agent builds its own cache, effectively doubling the workload on the OpenSSO Enterprise servers, and cutting overall system capacity. The problem will become more acute as the number of policy agents increases. In situations where each web policy agent instance is protecting identical resources, some form of load balancer persistence is highly recommended for these reasons. Although the actual type of persistence may vary when a different load balancer is used, it should achieve the goal of sending requests from the same user session to the same policy agent.

Note – When firewalls are configured, Load Balancer 4 can be located in a less secure zone.

Use the following list of procedures as a checklist for configuring the web policy agents' load balancer:

1. "To Configure the Web Policy Agents Load Balancer" on page 254
2. "To Create a Monitoring File on Each Host Machine for Load Balancer 4" on page 256
3. "To Add Load Balancer 4 as a Virtual Host by Modifying the Web Policy Agent Properties" on page 258
4. "To Configure Policy for the Web Policy Agents" on page 260
5. "To Verify the Web Policy Agents Load Balancer Configuration is Working Properly" on page 261

▼ To Configure the Web Policy Agents Load Balancer

Before You Begin The load balancer hardware and software used 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.

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

2. **Log in using the following credentials:**

   User name: *username*

   Password: *password*

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	**`WebAgent-Pool`**
Load Balancing Method	Round Robin
Resources	Add the IP address and port number of both Protected Resource host machines: `pr-1:1080` and `pr-2:1080`.

   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.

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. **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 `lb-4.example.com`
Service	**`90`**

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

e. **In the Pool Selection dialog box, assign the `WebAgent-Pool` Pool.**

f. **Click Done.**

6 **Add Monitors.**

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

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

b. **Click Add.**

In the Add Monitor dialog provide the following information:

Name:	**`WebAgent-http`**
Inherits From:	Choose http.

c. **Click Next.**

d. **On the resulting Configure Basic Properties page, click Next.**

e. **In the Send String field under Configure ECV HTTP Monitor, enter `GET /monitor.html` and click Next.**

f. **On the Destination Address and Service (Alias) page, click Done.**

The monitor just added is in the list of monitors under the Monitors tab.

g. **Click the Basic Associations tab.**

h. **Mark the Add checkbox next to the IP addresses for `pr-1:1080` and `pr-2:1080`.**

i. **At the top of the Node column, choose the monitor that you just added, `WebAgent-http`.**

j. **Click Apply.**

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

All requests sent *within a specified interval* from the same user are routed to the same agent. This significantly reduces the number of agent requests sent to OpenSSO Enterprise for validation thus reducing the load on the servers.

Note – Simple persistence tracks connections based on the client IP address only, returning a client to the same node to which it connected previously.

---

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

b. **Click the `WebAgent-Pool` link.**

c. **Click the Persistence tab.**

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

e. **Set the timeout interval.**

In the Timeout field, enter 300 seconds.

f. **Click Apply.**

8 **Log out of the console.**

▼ To Create a Monitoring File on Each Host Machine for Load Balancer 4

In order to configure the web policy agents to point to Load Balancer 4, create a file to be used by Load Balancer 4 for monitoring and modify the web agent properties — adding Load Balancer 4 as the virtual host. Instructions on how to create a monitoring file are in the following procedure. Instructions on how to modify the web agent properties are in "To Add Load Balancer 4 as a Virtual Host by Modifying the Web Policy Agent Properties" on page 258.

Note – We can alternately use the default Web Server `index.html` rather than create `monitor.html` but in this deployment, `index.html` is used to represent the resource protected by the web policy agent.

---

1. **As a root user, log in to the `pr–1` host machine.**

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

   ```
   # cd /opt/SUNWwbsvr/https-pr-1.example.com/docs
   ```

3. **Create a `monitor.html` file to be used by the load balancer.**

   ```
   # cat > monitor.html

   <HTML>
   </HTML>
   ```

   Hit Control D to terminate the command

   ```
   ^D
   ```

4. **Run the `tail` command.**

   ```
   # cd /opt/SUNWwbsvr/https-pr-1.example.com/logs
   # tail -f access
   ```

   If you see frequent entries similar to the one below, the custom monitor is configured properly.

   ```
   IP_address - - [30/Jul/2008:13:59:48 -0700] "GET /monitor.html" 200 15
   ```

Tip – If you do not see "`GET /monitor.html`", troubleshoot the load balancer configuration.

---

5. **Log out of the `pr–1` host machine.**

6. **As a root user, log in to the `pr–2` host machine.**

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

   ```
   # cd /opt/SUNWwbsvr/https-pr-2.example.com/docs
   ```

8. **Create a `monitor.html` file to be used by the load balancer.**

   ```
   # cat > monitor.html

   <HTML>
   </HTML>
   ```

Hit Control D to terminate the command

```
^D
```

9 **Run the `tail` command.**

```
# cd /opt/SUNWwbsvr/https-pr-2.example.com/logs
# tail -f access
```

If you see frequent entries similar to the one below, the custom monitor is configured properly.

```
IP_address - - [30/Jul/2008:13:59:48 -0700] "GET /monitor.html" 200 15
```

Tip – If you do not see "`GET /monitor.html`", troubleshoot the load balancer configuration.

---

10 **Log out of the `pr–2` host machine.**

▼ To Add Load Balancer 4 as a Virtual Host by Modifying the Web Policy Agent Properties

In order to configure the web policy agents to point to Load Balancer 4, create a file to be used by Load Balancer 4 for monitoring and modify the web agent properties — adding Load Balancer 4 as the virtual host.

Before You Begin This procedure assumes you have completed "To Create a Monitoring File on Each Host Machine for Load Balancer 4" on page 256.

1 **Access `https://osso-1.example.com:1081/opensso/console` from a web browser.**

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

Username **`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` and `webagent-2` is displayed under the Agent table.

6. **Click** `webagent-1`

   The Global tab is displayed.

7. **Enter a value for the FQDN Virtual Host Map and click Add.**

   The value is the name of the host machine in which Load Balancer 4 is installed.

Map Key	**`valid`**
Corresponding Map Value	**`lb-4.example.com`**

8. **Click Save.**

9. **Click the Application tab.**

   The Application properties page is displayed.

10. **On the resulting page, provide values for Not Enforced URL Processing.**

    Enter each of the following and click Add.

    ```
    http://lb-4.example.com:90/monitor.html
    http://pr-1.example.com:1080/monitor.html
    ```

11. **Click Save.**

12. **Click Back to Main Page.**

13. **Click** `webagent-2`

    The Global tab is displayed.

14. **Enter a value for the FQDN Virtual Host Map and click Add.**

    The value is the name of the host machine in which Load Balancer 4 is installed.

Map Key	**`valid`**.
Corresponding Map Value	**`lb-4.example.com`**

15. **Click Save.**

16. **Click the Application tab.**

    The Application properties page is displayed.

17. **On the resulting page, provide values for Not Enforced URL Processing.**

    Enter each of the following and click Add.

    ```
    http://lb-4.example.com:90/monitor.html
    http://pr-2.example.com:1080/monitor.html
    ```

18 **Click Save.**

19 **Click Back to Main Page.**

20 **Log out of the OpenSSO Enterprise console and close the browser.**

▼ To Configure Policy for the Web Policy Agents

Use the OpenSSO Enterprise console to configure policy for the web policy agents. The policies you create here are used in "To Verify the Web Policy Agents Load Balancer Configuration is Working Properly" on page 261.

1 **Access** `https://osso-1.example.com:1081/opensso/console` **from a web browser.**

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

Username	**`amadmin`**
Password	**`ossoadmin`**

3 **Under the Access Control tab, click `/ (Top Level Realm)`.**

4 **Click the Policies tab.**

5 **Click New Policy.**

The New Policy page is displayed.

6 **On the New Policy page, enter `URL Policy for LoadBalancer-4` in the Name field.**

7 **Click New under Rules.**

The New Rules page is displayed.

8 **On the New Rules page, accept the default** `URL Policy Agent (with resource name)` **and click Next.**

9 **On the resulting page, provide the following information.**

Name:	**`Rule for LoadBalancer-4`**.
Resource Name:	**`http://lb-4.example.com:90/*`**
GET	Mark this checkbox and verify that Allow is selected.
POST	Mark this checkbox and verify that Allow is selected.

10 **Click Finish.**

The New Policy page is displayed again.

11 **On the New Policy page, click New under Subjects.**

The New Subjects page is displayed.

12 **On the New Subjects page, verify that Access Manager Identity Subject is selected and click Next.**

13 **On the resulting page, provide the following information.**

Name	**`Subject for LoadBalancer-4`**.
Filter	From the drop-down list, select User and click Search. The search returns a list of available users.
Available	From the generated User list, select `testuser1` and click Add. `testuser1` is displayed in the Selected List.

14 **Click Finish.**

The New Policy page is displayed again.

15 **On the New Policy page, click OK.**

The completed policy is now included in the list of Policies.

16 **Log out of the OpenSSO Enterprise console and close the browser.**

▼ To Verify the Web Policy Agents Load Balancer Configuration is Working Properly

1 **Access `http://lb-4.example.com:90/index.html`, the OpenSSO Enterprise load balancer, from a web browser.**

2 **Log in to OpenSSO Enterprise as `testuser1`.**

Username	**`testuser1`**
Password	**`password`**

If the default Web Server `index.html` page is displayed, the load balancer is configured properly.

3 **Close the browser.**

4 **Access the OpenSSO Enterprise load balancer at** `http://lb-4.example.com:90/index.html` **from a web browser again.**

Tip – If not redirected to the OpenSSO Enterprise login page for authentication, clear your browser's cache and cookies and try again.

---

5 **Log in to OpenSSO Enterprise as** `testuser2`**.**

Username **`testuser2`**

Password **`password`**

You should see the message *You're not authorized to view this page* or *Your client is not allowed to access the requested object* as `testuser2` was not included in the test policy.

9.2 Configuring the J2EE Policy Agents Load Balancer

From a performance perspective, each policy agent validates user sessions and evaluates applicable policies. The results of those actions are cached by the policy agent that performed them. If simple persistence is **not** set, each agent builds its own cache, effectively doubling the workload on the OpenSSO Enterprise servers, and cutting overall system capacity. The problem will become more acute as the number of policy agents increases. In situations where each web policy agent instance is protecting identical resources, some form of load balancer persistence is highly recommended for these reasons. Although the actual type of persistence may vary when a different load balancer is used, it should achieve the goal of sending requests from the same user session to the same policy agent. Thus we deploy Load Balancer 5 to handle traffic for the J2EE policy agents, and configure the Load Balancer for simple persistence. Use the following list of procedures as a checklist for configuring the J2EE policy agents' load balancer.

1. "To Configure the J2EE Policy Agents Load Balancer" on page 262
2. "To Add Load Balancer 5 as a Virtual Host by Modifying the J2EE Policy Agent Properties" on page 264
3. "To Configure Policy for the J2EE Policy Agents" on page 265
4. "To Verify the J2EE Policy Agent Load Balancer Configuration is Working Properly" on page 267

▼ To Configure the J2EE Policy Agents Load Balancer

Before You Begin The load balancer hardware and software used 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.

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

2 **Log in using the following information:**

User name: **`username`**

Password: **`password`**

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	`J2EEAgent-Pool`
Load Balancing Method	Round Robin
Resources	Add the Application Server IP addresses and port numbers: `pr-1:1081` and `pr-2:1081`.

d. **Click Done.**

e. **In the List of Pools, click** `J2EEAgent-Pool`**.**

f. **Click the Persistence tab and provide the following information:**

Persistence Type: Choose `Active Http Cookie`

Note – `Active Http Cookie` persistence uses an HTTP cookie stored on a client computer to allow the client to reconnect to the same server previously visited.

Method: Choose `Insert`

g. **Click Apply.**

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 for this step.

---

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 `lb-5.example.com`
Services Port	**91**

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

e. **In the Pool Selection dialog box, assign the `J2EEAgent-Pool` pool.**

f. **Click Done.**

6 **Add Monitors.**

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

a. **Click Monitors in the left frame.**

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

c. **Mark the Add checkbox for the IP address for `pr–1:1081` and `pr–2:1081`.**

d. **At the top of the Node column, select `tcp`.**

e. **Click Apply.**

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

▼ To Add Load Balancer 5 as a Virtual Host by Modifying the J2EE Policy Agent Properties

In order to configure the J2EE policy agents to point to Load Balancer 5, modify the J2EE agent properties — adding Load Balancer 5 as the virtual host.

1 **Access `https://osso-1.example.com:1081/opensso/console` from a web browser.**

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

Username	**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` and `j2eeagent-2` are displayed under the Agent table.

6 **Click** `j2eeagent-1`

The Global tab is displayed.

7 **Enter a key and value for the FQDN Virtual Host Map and click Add.**

The key and the value is the name of the host machine in which Load Balancer 5 is installed.

Map Key	**lb-5.example.com**.
Corresponding Map Value	**lb-5.example.com**

8 **Click Save.**

9 **Click Back to Main Page.**

10 **Click** `j2eeagent-2`

The Global tab is displayed.

11 **Enter a key and value for the FQDN Virtual Host Map and click Add.**

The key and the value is the name of the host machine in which Load Balancer 5 is installed.

Map Key	**lb-5.example.com**.
Corresponding Map Value	**lb-5.example.com**

12 **Click Save.**

13 **Click Back to Main Page.**

▼ To Configure Policy for the J2EE Policy Agents

The policies you create here are used in "To Verify the J2EE Policy Agent Load Balancer Configuration is Working Properly" on page 267.

Before You Begin This procedure assumes that you have just completed "To Add Load Balancer 5 as a Virtual Host by Modifying the J2EE Policy Agent Properties" on page 264 and are still logged into the OpenSSO Enterprise console.

1. **Under the Access Control tab, click `/ (Top Level Realm)`.**

2. **Click the Policies tab.**

3. **Click New Policy.**

   The New Policy page is displayed.

4. **On the New Policy page, enter `URL Policy for LoadBalancer-5` in the Name field.**

5. **Click New under Rules.**

   The New Rules page is displayed.

6. **On the New Rules page, accept the default `URL Policy Agent (with resource name)` and click Next.**

7. **On the resulting page, provide the following information.**

Name:	**`Rule for LoadBalancer-5`**.
Resource Name:	**`http://lb-5.example.com:91/*`**
GET	Mark this checkbox and verify that Allow is selected.
POST	Mark this checkbox and verify that Allow is selected.

8. **Click Finish.**

9. **On the New Policy page again, under Subjects, click New.**

10. **On the resulting page, verify that Access Manager Identity Subject is selected, and click Next.**

11. **On the resulting page, provide the following information:**

    Name: **`LoadBalancer-5_Groups`**

    Filter: In the drop-down list, select Group and click Search.

    The search returns a list of available groups.

12. **Select `Employee-Group` and `Manager-Group` and click Add.**

    The `Employee-Group` and `Manager-Group` groups are in the Selected List.

13. **Click Finish.**

14 **On the resulting page, click OK.**

The created policy is displayed in the list of Policies.

15 **Log out of the OpenSSO Enterprise console and close the browser.**

▼ To Verify the J2EE Policy Agent Load Balancer Configuration is Working Properly

1 **Access** `http://lb-5.example.com:91/agentsample/index.html` **from a web browser.**

The Sample Application welcome page is displayed.

2 **Click the J2EE Declarative Security link.**

3 **On the resulting page click Invoke the Protected Servlet.**

The policy agent redirects to the OpenSSO Enterprise login page.

4 **Log in to OpenSSO Enterprise as** `testuser1`**.**

Username **`testuser1`**

Password **`password`**

If you can successfully log in as `testuser1` and the J2EE Policy Agent Sample Application page is displayed, this first part of the test succeeded and authentication is working as expected.

5 **Click the J2EE Declarative Security link to return.**

6 **On the resulting page, click Invoke the Protected Servlet.**

If the Successful Invocation message is displayed, this second part of the test has succeeded and the sample policy for the employee role has been enforced as expected.

7 **Close the browser.**

8 **Open a new browser and access** `http://lb-5.example.com:91/agentsample/index.html`**.**

The Sample Application welcome page is displayed.

9 **Click the J2EE Declarative Security link.**

10 **On the resulting page click Invoke the Protected Servlet.**

The policy agent redirects to the OpenSSO Enterprise login page.

11 **Log in to OpenSSO Enterprise as** `testuser2`**.**

Username `testuser2`

Password `password`

If the Access to Requested Resource Denied message is displayed, this third part of the test succeeded and the sample policy for the manager role has been enforced as expected.

12 **Click the J2EE Declarative Security link to return.**

13 **On the resulting page, click Invoke the Protected EJB via an Unprotected Servlet.**

If the Successful Invocation message is displayed, the sample policy for the employee role has been enforced as expected.

14 **Close the browser.**

CHAPTER 10

Implementing Session Failover

Sun OpenSSO Enterprise provides a web container-independent session failover feature that uses Sun Java™ System Message Queue, a messaging middleware product that enables distributed applications to communicate and exchange data by sending and receiving messages. OpenSSO Enterprise uses Message Queue as a communications broker, and the BerkeleyDB by Sleepycat Software, Inc. for backend session store databases. This chapter contains the following sections:

- "10.1 Session Failover Architecture" on page 269
- "10.2 Installing the Session Failover Components" on page 270
- "10.3 Configuring and Verifying Session Failover" on page 278

10.1 Session Failover Architecture

When session failover is implemented for OpenSSO Enterprise, session information is replicated in two backend session store databases. This ensures that if one OpenSSO Enterprise server fails or stops, the information stored in the backend session databases can be used to keep the user continuously authenticated. If session failover is not implemented and the OpenSSO Enterprise server in which a user's session was created fails, the user will have to reauthenticate to perform an operation that requires a session token.

Note – For more information about OpenSSO Enterprise and session failover, see Chapter 7, "Implementing OpenSSO Enterprise Session Failover," in *Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide*.

10.2 Installing the Session Failover Components

Install the OpenSSO Enterprise session failover components on the `mq-1` host machine and the `mq-2` host machine. Use the following list of procedures as a checklist for completing the task.

1. "To Install Session Failover Components on Message Queue 1" on page 270
2. "To Install Session Failover Components on Message Queue 2" on page 274

▼ To Install Session Failover Components on Message Queue 1

1 **As a root user, log in to the `mq–1` host machine.**

2 **Create a directory into which the Message Queue and Berkeley Database bits can be downloaded and change into it.**

```
# mkdir /export/SFO
# cd /export/SFO
```

3 **Copy `ssoSessionTools.zip` from the `osso–1` host machine to the `mq–1` host machine.**

`ssoSessionTools.zip` is included in the `opensso.zip` file downloaded in "To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine" on page 108 under the `tools` directory.

4 **Unzip `ssoSessionTools.zip`.**

```
# cd /export/SFO
# unzip ssoSessionTools.zip -d ssoSessionTools
```

5 **Modify the permissions on the `setup` script and run it to initialize the session failover tools.**

```
# cd /export/SFO/ssoSessionTools
# chmod +x setup
# ./setup
```

6 **When prompted, enter `opensso` as the** *Directory to install the scripts (example: opensso)***.**

Note – The directory location should be relative to the current directory.

When the script is finished, the following messages are displayed:

```
The scripts are properly setup under directory
   /export/SFO/ssoSessionTools/opensso
JMQ is properly setup under directory
   /export/SFO/ssoSessionTools/jmq
```

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

```
# cd /export/SFO/ssoSessionTools/jmq/mq/bin
```

8 **Run the `imqbrokerd` command to create a new broker instance named `msgqbroker`.**

```
# ./imqbrokerd -name msgqbroker -port 7777 &
```

9 **Run `netstat` to verify that the new Message Queue broker instance is up and running.**

```
# netstat -an | grep 7777

*.7777        *.*           0            0    49152        0    LISTEN
```

10 **Add a new user named `msgquser`.**

This user will connect to the Message Queue broker instance on servers where Message Queue is installed. This user will be used only for session failover purposes, and does not assume the privileges of the guest user. It is a good practice to create a custom user for such purposes, and not to rely on the known user accounts or default user accounts to help prevent brute force or DOS attacks.

```
# ./imqusermgr add -u msgquser -g admin -p m5gqu5er -i msgqbroker

User repository for broker instance: msgqbroker

User msgquser successfully added.
```

11 **Disable the guest user.**

This step ensures that the guest user will not be able to access the OpenSSO Enterprise server.

```
# ./imqusermgr update -u guest -a false -i msgqbroker

User repository for broker instance: msgqbroker

Are you sure you want to update user guest? (y/n) y

User guest successfully updated.
```

12 **Modify the `amsfo.conf` file.**

`amsfo.conf` has parameters that are consumed by the OpenSSO Enterprise session failover startup script, `amsfo`.

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

```
# cd /export/SFO/ssoSessionTools/opensso/config/lib
```

Tip – Backup `amsfo.conf` before you modify it.

---

b. **Set the following properties:**

```
CLUSTER_LIST=mq-1.example.com:7777,mq-2.example.com:7777
BROKER_INSTANCE_NAME=msgqbroker
USER_NAME=msgquser
BROKER_PORT=7777
```

Note – The port used for `BROKER_PORT` should be the same as the one used in the value of the `CLUSTER_LIST`.

---

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

13 **Generate an encrypted password in a `.password` file with the following sub procedure.**

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

```
# cd /export/SFO/ssoSessionTools/opensso/bin
```

b. **Run `amsfopassword`.**

This command generates an encrypted password, creates a new file named `.password`, and stores the encrypted password in the new file.

Caution – `amsfopassword` creates the `.password` file in a default location based on where the scripts were installed. If a different location is used, the `PASSWORDFILE` property in `amsfo.conf` should be changed accordingly.

```
# ./amsfopassword -e m5gqu5er -f /export/SFO/ssoSessionTools/opensso/.password

os.name=SunOS
SUCCESSFUL
```

c. **(Optional) View the encrypted password for verification.**

```
# more /export/SFO/ssoSessionTools/opensso/.password

M27OGb6U4ufRu+oWAzBdWw==
```

14 **(Optional) Modify the `amsessiondb` script if necessary.**

The `amsessiondb` script (located in the `/export/SFO/ssoSessionTools/opensso/bin` directory) starts the Berkeley Database client, creates the database, and sets specific database values. It is called when the `amsfo` script is run for the first time. The `amsessiondb` script

contains variables that specify default paths and directories. If any of the following components are not installed in their default directories, edit the `amsessiondb` script to set the variables to the correct locations.

```
JAVA_HOME=/usr/jdk/entsys-j2se
IMQ_JAR_PATH=/export/SFO/ssoSessionTools/jmq/mq/lib
JMS_JAR_PATH=/export/SFO/ssoSessionTools/jmq/mq/lib
AM_HOME=/export/SFO/ssoSessionTools
```

Tip – Backup `amsessiondb` before you modify it.

---

15 Restart the session failover components with the following sub procedure.

a. Change to the `bin` directory.

```
# cd /export/SFO/ssoSessionTools/jmq/mq/bin
```

b. Stop the Message Queue instance using the product's command line interface.

See the Message Queue documentation for more information.

c. Run the `netstat` command to verify that the `mq-1` broker instance is stopped.

```
# netstat -an | grep 7777
```

If `netstat` returns no result, the `mq-1` broker instance is stopped.

Tip – If the `mq-1` broker instance is not stopped, kill the process using the following procedure.

a. Get the Java process IDs.

```
# ps -ef | grep java
```

b. Kill the Java process IDs that were returned.

```
# kill -9 #### ####
```

c. Run `netstat` again.

---

d. Restart the `mq-1` broker instance.

```
# cd /export/SFO/ssoSessionTools/opensso/bin
# ./amfso start
```

e. Run the `netstat` command to verify that the Message Queue port is open and listening.

```
# netstat -an | grep 7777

*.7777           *.*           0           0    49152           0    LISTEN
```

16 **Log out of the `mq-1` host machine.**

▼ To Install Session Failover Components on Message Queue 2

1 **As a root user, log in to the `mq–2` host machine.**

2 **Create a directory into which the Message Queue and Berkeley Database bits can be downloaded and change into it.**

```
# mkdir /export/SFO
# cd /export/SFO
```

3 **Copy `ssoSessionTools.zip` from the `osso–1` host machine to the `mq–2` host machine.**

Note – `ssoSessionTools.zip` is included in the `opensso.zip` file downloaded in "To Generate an OpenSSO Enterprise WAR on the OpenSSO Enterprise 1 Host Machine" on page 108 under the `tools` directory.

---

4 **Unzip `ssoSessionTools.zip`.**

```
# cd /export/SFO
# unzip ssoSessionTools.zip -d ssoSessionTools
```

5 **Modify the permissions on the `setup` script and run it to initialize the session failover tools.**

```
# cd /export/SFO/ssoSessionTools
# chmod +x setup
# ./setup
```

6 **When prompted, enter `opensso` as the** *Directory to install the scripts (example: opensso)***.**

Note – The directory location should be relative to the current directory.

When the script is finished, the following messages are displayed:

```
The scripts are properly setup under directory
   /export/SFO/ssoSessionTools/opensso
JMQ is properly setup under directory
   /export/SFO/ssoSessionTools/jmq
```

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

```
# cd /export/SFO/ssoSessionTools/jmq/mq/bin
```

8 **Run the `imqbrokerd` command to create a new broker instance named `msgqbroker`.**

```
# ./imqbrokerd -name msgqbroker -port 7777 &
```

9 **Run `netstat` to verify that the new Message Queue broker instance is up and running.**

```
# netstat -an | grep 7777

*.7777        *.*           0           0    49152        0    LISTEN
```

10 **Add a new user named `msgquser`.**

This user will connect to the Message Queue broker instance on servers where Message Queue is installed. This user will be used only for session failover purposes, and does not assume the privileges of the guest user. It is a good practice to create a custom user for such purposes, and not to rely on the known user accounts or default user accounts to help prevent brute force or DOS attacks.

```
# ./imqusermgr add -u msgquser -g admin -p m5gqu5er -i msgqbroker

User repository for broker instance: msgqbroker

User msgquser successfully added.
```

11 **Disable the guest user.**

This step ensures that the guest user will not be able to access the OpenSSO Enterprise server.

```
# ./imqusermgr update -u guest -a false -i msgqbroker

User repository for broker instance: msgqbroker

Are you sure you want to update user guest? (y/n) y

User guest successfully updated.
```

12 **Modify the `amsfo.conf` file with the following sub procedure.**

`amsfo.conf` has parameters that are consumed by the OpenSSO Enterprise session failover startup script, `amsfo`.

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

```
# cd /export/SFO/ssoSessionTools/opensso/config/lib
```

Tip – Backup `amsfo.conf` before you modify it.

---

b. **Set the following properties:**

```
CLUSTER_LIST=mq-1.example.com:7777,mq-2.example.com:7777
BROKER_INSTANCE_NAME=msgqbroker
```

```
USER_NAME=msgquser
BROKER_PORT=7777
```

Note – The port used for `BROKER_PORT` should be the same as the one used in the value of the `CLUSTER_LIST`.

---

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

13 **Generate an encrypted password in a `.password` file with the following sub procedure.**

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

```
# cd /export/SFO/ssoSessionTools/opensso/bin
```

b. **Run `amsfopassword`.**

This command generates an encrypted password, creates a new file named `.password`, and stores the encrypted password in the new file.

Caution – `amsfopassword` creates the `.password` file in a default location based on where the scripts were installed. If a different location is used, the `PASSWORDFILE` property in `amsfo.conf` should be changed accordingly.

```
# ./amsfopassword -e m5gqu5er -f /export/SFO/ssoSessionTools/opensso/.password

os.name=SunOS
SUCCESSFUL
```

c. **(Optional) View the encrypted password for verification.**

```
# more /export/SFO/ssoSessionTools/opensso/.password

M27OGb6U4ufRu+oWAzBdWw==
```

14 **(Optional) Modify the `amsessiondb` script if necessary.**

The `amsessiondb` script (located in the `/export/SFO/ssoSessionTools/opensso/bin` directory) starts the Berkeley Database client, creates the database, and sets specific database values. It is called when the `amsfo` script is run for the first time. The `amsessiondb` script contains variables that specify default paths and directories. If any of the following components are not installed in their default directories, edit the `amsessiondb` script to set the variables to the correct locations.

```
JAVA_HOME=/usr/jdk/entsys-j2se
IMQ_JAR_PATH=/export/SFO/ssoSessionTools/jmq/mq/lib
JMS_JAR_PATH=/export/SFO/ssoSessionTools/jmq/mq/lib
AM_HOME=/export/SFO/ssoSessionTools
```

Tip – Backup `amsessiondb` before you modify it.

---

15 Restart the session failover components.

a. Change to the `bin` directory.

```
# cd /export/SFO/ssoSessionTools/jmq/mq/bin
```

b. Stop the Message Queue instance using the product's command line interface.

See the Message Queue documentation for more information.

c. Run the `netstat` command to verify that the `mq-2` broker instance is stopped.

```
# netstat -an | grep 7777
```

If `netstat` returns no result, the `mq-2` broker instance is stopped.

Tip – If the `mq-2` broker instance is not stopped, kill the process using the following procedure.

a. Get the Java process IDs.

```
# ps -ef | grep java
```

b. Kill the Java process IDs that were returned.

```
# kill -9 #### ####
```

c. Run `netstat` again.

---

d. Restart the `mq-2` broker instance.

```
# cd /export/SFO/ssoSessionTools/opensso/bin
# ./amfso start
```

e. Run the `netstat` command to verify that the Message Queue port is open and listening.

```
# netstat -an | grep 7777

*.7777           *.*           0           0    49152           0    LISTEN
```

16 Log out of the `mq-2` host machine.

10.3 Configuring and Verifying Session Failover

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

1. "To Configure OpenSSO Enterprise for Session Failover" on page 278
2. "To Verify That the Administrator Session Fails Over" on page 280
3. "To Verify that the User Session Fails Over" on page 281

▼ To Configure OpenSSO Enterprise for Session Failover

1. **Access** `https://osso-1.example.com:1081/opensso/console` **from a web browser.**

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

   Username **`amadmin`**

   Password **`ossoadmin`**

3. **Click the Configuration tab.**

4. **Under Global properties, click Session.**

5. **Under Secondary Configuration Instance, click New.**

6. **In the Add Sub Configuration page, provide the following information.**

Name	Select External
Session Store User	Enter **`msgquser`**
Session Store Password	Enter **`m5gqu5er`**
Session Store Password (confirm)	Enter **`m5gqu5er`**
Maximum Wait Time	Keep the default value of 5000.
Database URL	Enter **`mq-1.example.com:7777,mq-2.example.com:7777`**. This is the Message Queue broker address list. Enter multiple values using a comma and no space.

7. **Click Add.**

8. **Click Save.**

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

10 **Restart the Application Server 1 instance with the following sub procedure.**

a. **As a root user, log in to the** `osso–1` **host machine.**

b. **Switch to the non-root user and change to the** `bin` **directory.**

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

c. **Restart the Application Server 1 instance.**

```
# ./stopserv; ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

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

d. **Log out of the** `osso–1` **host machine.**

11 **Restart the Application Server 2 instance with the following sub procedure.**

a. **As a root user, log in to the** `osso–2` **host machine.**

b. **Switch to the non-root user and change to the** `bin` **directory.**

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

c. **Restart the Application Server 2 instance.**

```
# ./stopserv; ./startserv

admin username:domain2adm

admin password:domain2pwd

master password:domain2master

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

d. **Log out of the** `osso-2` **host machine.**

▼ To Verify That the Administrator Session Fails Over

Before You Begin Both OpenSSO Enterprise 1 and OpenSSO Enterprise 2 should be up and running before you begin this verification procedure.

1. **As a root user, log in to the `osso–2` host machine.**

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

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

3. **Stop OpenSSO Enterprise 2.**

   ```
   # ./stopserv
   ```

4. **Access `https://lb-2.example.com:1081/opensso/console` from a web browser.**

   a. **Log in to the OpenSSO Enterprise console as the administrator.**

      Username **`amadmin`**

      Password **`ossoadmin`**

   b. **Click the Sessions tab.**

   c. **In the View field, select `osso-1.example.com:1081` from the drop down list.**

      Verify that only `amadmin` exists in the Sessions table.

   d. **In the View field, select `osso-2.example.com:1081` from the drop down list.**

      You will see an error message indicating the server is down.

   e. **Leave this browser window 1 open.**

5. **Start OpenSSO Enterprise 2.**

   ```
   # ./startserv

   admin username:domain2adm

   admin password:domain2pwd

   master password:domain2master

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

6. **As a root user, log in to the `osso-1` host machine.**

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

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

8 **Stop OpenSSO Enterprise 1.**

```
# ./stopserv
```

9 **Going back to the OpenSSO Enterprise console in browser window 1, under the Sessions tab, select `osso-1.example.com:1081` from the View drop down list.**

You will see an error message indicating the server is down.

10 **Now select `osso-2.example.com:1081` from the View drop down list.**

Verify that only `amadmin` exists in the Sessions table. This indicates that although OpenSSO Enterprise 1 was stopped, the OpenSSO Enterprise Load Balancer 2 directed the request to OpenSSO Enterprise 2 and a session for `amadmin` was successfully created by OpenSSO Enterprise 2. If session failover was not enabled, it would have resulted in a login page.

▼ To Verify that the User Session Fails Over

Before You Begin This procedure assumes that you have just completed "To Verify That the Administrator Session Fails Over" on page 280.

1 **Access `https://lb-2.example.com:1081/opensso/UI/Login` from a second browser window.**

2 **Log in to the OpenSSO Enterprise console as `testuser1`.**

Username **`testuser1`**

Password **`password`**

A page with a message that reads *You're logged in* is displayed. Since the User Profile attribute was set to Ignored, the user's profile is not displayed following a successful login. Because OpenSSO Enterprise 1 was stopped, the user session is created in OpenSSO Enterprise 2.

3 **Leave browser window 2 open.**

4 **Using browser window 1, click the Sessions tab.**

5 **In the View field, select `osso-2.example.com:1081` from the drop down list.**

Verify that `amadmin` and `testuser1` exist in the Sessions table.

6 **On the `osso–1` host machine, change to the `bin` directory.**

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

7 **Start OpenSSO Enterprise 1.**

```
# ./startserv
```

Both OpenSSO Enterprise 1 and OpenSSO Enterprise 2 are up and running.

8 **On the `osso–2` host machine, change to the `bin` directory.**

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

9 **Stop OpenSSO Enterprise 2.**

```
# ./stopserv
```

10 **Using browser window 1, click the Sessions tab and do the following sub procedure.**

a. **In the View field, select `osso-1.example.com:1081`.**

Verify that `amadmin` and `testuser1` exist in the Sessions table. This indicates that the session successfully failed over to OpenSSO Enterprise 1.

Tip – If `testuser1` is not displayed, refresh the browser window 2 page.

---

b. **In the View field, select `osso-2.example.com:1081`**

You will see an error message indicating the server is down.

11 **Log out of the consoles and the host machines.**

PART III

Reference: Summaries of Server and Component Configurations

This final section of *Deployment Example: Single Sign-On, Load Balancing and Failover Using Sun OpenSSO Enterprise 8.0* contains component descriptions and configurations for the software and hardware used.

- Appendix A, "Directory Server Host Machines, Test Users and Load Balancer"
- Appendix B, "OpenSSO Enterprise Host Machines and Load Balancer"
- Appendix C, "OpenSSO Enterprise Distributed Authentication User Interface Host Machines and Load Balancer"
- Appendix D, "Protected Resource Host Machine Web Containers, Policy Agents and Load Balancers"
- Appendix E, "Message Queue Servers"
- Appendix F, "Known Issues and Limitations"

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

Directory Server Host Machines, Test Users and Load Balancer

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
- User Test Entries

TABLE A–1 Sun Java System Directory Server 1 Host Machine

Components	Description	
Host Name	ds–1.example.com	
Installation Directory	/var/opt/mps/serverroot/	
Administrator User	cn=Directory Manager	
Administrator Password	dsmanager	
User Data Instance	Instance Name	am-users
	Instance Directory	/var/opt/mps/am-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	ds–2.example.com	
Installation Directory	/var/opt/mps/serverroot/	
Administrator User	cn=Directory Manager	
Administrator Password	dsmanager	
User Data Instance	Instance Name	am-users
	Instance Directory	/var/opt/mps/am-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	lb-1.example.com
Method	Round Robin
Protected Servers	ds-1.example.com:1736
	ds-2.example.com:1736
Virtual Servers	lb-1.example.com:489
Monitors	ds-1.example.com:1736
	ds-2.example.com:1736

TABLE A–4 User Test Entries

UserID	Description	
testuser1	Password	password
	DN	uid=testuser1,ou=users,dc=company,dc=com
testuser2	Password	password
	DN	uid=testuser2,ou=users,dc=company,dc=com

APPENDIX B

OpenSSO Enterprise Host Machines and Load Balancer

This appendix collects the information regarding the OpenSSO Enterprise host machines. It contains the following tables:

- OpenSSO Enterprise 1 Host Machine
- OpenSSO Enterprise 2 Host Machine
- Load Balancer for OpenSSO Enterprise Host Machines

TABLE B–1 OpenSSO Enterprise 1 Host Machine

Component	Description	
Host Name	osso-1.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://osso–1.example.com:4848

TABLE B–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://osso–1.example.com:8989
OpenSSO Enterprise	Administrative User	amadmin
	Administrative User Password	ossoadmin
	Configuration Data Store	Embedded
	User Data Store	lb-1.example.com:489
	Agent User	agentuser
	Agent User Password	agentuser
	Administrative Console URL	https://osso–1.example.com:1081/opensso/console

TABLE B–2 OpenSSO Enterprise 2 Host Machine

Component	Description	
Host Name	osso-2.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://osso–2.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://osso–2.example.com:8989
OpenSSO Enterprise	Administrative User	amadmin
	Administrative User Password	ossoadmin

TABLE B–2 OpenSSO Enterprise 2 Host Machine *(Continued)*

Component	Description	
	Configuration Data Store	Embedded
	User Data Store	lb-1.example.com:489
	Agent User	agentuser
	Agent User Password	agentuser
	Administrative Console URL	https://osso–2.example.com:1081/opensso/console

TABLE B–3 Load Balancer for OpenSSO Enterprise Host Machines

Component	Description
URL	lb-2.example.com
Method	Round Robin
Protected Servers	osso-1.example.com:1081
	osso-2.example.com:1081
Virtual Servers	lb-2.example.com:489
Monitors	osso-1.example.com:1081
	osso-2.example.com:1081
Cookie Name	amlbcookie

APPENDIX C

OpenSSO Enterprise Distributed Authentication User Interface Host Machines and Load Balancer

This appendix collects the information regarding the instances of the OpenSSO Enterprise Distributed Authentication User Interface. It contains the following tables:

- Distributed Authentication User Interface Host Machine 1
- Distributed Authentication User Interface Host Machine 2
- Load Balancer for the Distributed Authentication User Interface Host Machines

TABLE C–1 Distributed Authentication User Interface Host Machine 1

Component	Description	
Host Name	da-1.example.com	
Non-Root User	da80adm	
Non-Root User Password	da80a6m	
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	da80adm
	Ports	8989 (SSL) 1080 (HTTP)
Sun Java System Web Server Instance	Instance Name	da-1.example.com

TABLE C–1 Distributed Authentication User Interface Host Machine 1 *(Continued)*

Component	Description	
	Instance Directory	/opt/SUNWwbsvr/https-da-1.example.com
	Port	1080 (HTTP)
		1443 (SSL)
	Service URL	http://da-1.example.com:1080
		https://da-1.example.com:1443
Distributed Authentication User Interface	Server Protocol	https
	Server Host	lb-2.example.com
	Server Port	1081
	Server Deployment URI	opensso
	distAuth Protocol	http
		https
	distAuth Host	da–1.example.com
	distAuth Port	1080 (HTTP)
		1443 (SSL)
	distAuth Deployment URI	distAuth
	distAuth Cookie Name	AMDistAuthCookie
	Application User Name	authuiadmin
	Application User Password	authuiadmin

TABLE C–2 Distributed Authentication User Interface Host Machine 2

Component	Description	
Host Name	da-2.example.com	
Non-Root User	da80adm	
Non-Root User Password	da80a6m	
Sun Java System Web Server Administration Server	Installation Directory	/opt/SUNWwbsvr/
	Default Administration Directory	/opt/SUNWwbsvr/admin-server
	Default Administrator	admin

TABLE C–2 Distributed Authentication User Interface Host Machine 2 *(Continued)*

Component	Description	
	Default Administrator Password	web4dmin
	Runtime User ID	da80adm
	Ports	8989 (SSL)
		1080 (HTTP)
Sun Java System Web Server Instance	Instance Name	da-2.example.com
	Instance Directory	/opt/SUNWwbsvr/https-da-2.example.com
	Port	1080 (HTTP)
		1443 (SSL)
	Service URL	http://da-2.example.com:1080
		https://da-2.example.com:1443
Distributed Authentication User Interface	Server Protocol	https
	Server Host	lb-2.example.com
	Server Port	1081
	Server Deployment URI	opensso
	distAuth Protocol	http
		https
	distAuth Host	da–2.example.com
	distAuth Port	1080 (HTTP)
		1443 (SSL)
	distAuth Deployment URI	distAuth
	distAuth Cookie Name	AMDistAuthCookie
	Application User Name	authuiadmin
	Application User Password	authuiadmin

TABLE C–3 Load Balancer for the Distributed Authentication User Interface Host Machines

Component	Description
URL	lb-3.example.com
Method	Round Robin
Protected Servers	da-1.example.com:1443
	da-2.example.com:1443
Virtual Servers	lb-3.example.com:1443
Monitors	da-1.example.com:1443
	da-2.example.com:1443
Cookie Name	DistAuthLBCookie
OpenSSO Enterprise Agent Profile	authuiadmin
OpenSSO Enterprise Agent Profile Password	authuiadmin

APPENDIX D

Protected Resource Host Machine Web Containers, Policy Agents and Load Balancers

This appendix collects the information regarding the web containers and policy agents installed on the Protected Resource host machines. It contains the following tables:

- Protected Resource 1 Host Machine
- Protected Resource 2 Host Machine
- Load Balancer for the Web Policy Agents
- Load Balancer for the J2EE Policy Agents

TABLE D–1 Protected Resource 1 Host Machine

Component	Description	
Host Name	pr-1.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/pr-1
	Administration Server Directory	/usr/local/bea/user_projects/domains/pr-1/servers/AdminServer
	Administrator	weblogic
	Administrator Password	bea10admin
	Port	7001
	Administration Console URL	http://pr-1.example.com:7001/console
BEA WebLogic Server Managed Server	Managed Server Directory	/usr/local/bea/user_projects/domains/pr-1/servers/ApplicationServer-1
	Port	1081

TABLE D–1 Protected Resource 1 Host Machine *(Continued)*

Component	Description	
	OpenSSO Enterprise URL	https://lb-2.example.com:1081/opensso
	Distributed Authentication User Interface URL	https://lb-3.example.com:1443/distAuth/UI/Login
J2EE Policy Agent for BEA WebLogic Server	J2EE Agent Profile Name	j2eeagent–1
	J2EE Agent Profile Password	j2eeagent1
	J2EE Agent URL	http://pr-1.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	pr-1.example.com
	Instance Directory	/opt/SUNWwbsvr/https-pr-1.example.com
	Port	1080
	Service URL	http://pr-1.example.com:1080
Web Policy Agent for Sun Java System Web Server	Web Agent Profile Name	webagent-1
	Web Agent Profile Password	webagent1

TABLE D–2 Protected Resource 2 Host Machine

Component	Description	
Host Name	pr-2.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/pr-2
	Administration Server Directory	/usr/local/bea/user_projects/domains/pr-2/servers/AdminServer
	Administrator	weblogic
	Administrator Password	bea10admin
	Port	7001
	Administration Console URL	http://pr-2.example.com:7001/console
BEA WebLogic Server Managed Server	Managed Server Directory	/usr/local/bea/user_projects/domains/pr-2/servers/ApplicationServer-2
	Port	1081
	OpenSSO Enterprise URL	https://lb-2.example.com:1081/opensso
	Distributed Authentication User Interface URL	https://lb-3.example.com:1443/distAuth/UI/Login
J2EE Policy Agent for BEA WebLogic Server	J2EE Agent Profile Name	j2eeagent–2
	J2EE Agent Profile Password	j2eeagent2
	J2EE Agent URL	http://pr-2.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

TABLE D–2 Protected Resource 2 Host Machine *(Continued)*

Component	Description	
	Ports	8989 (SSL)
		1080 (HTTP)
Sun Java System Web Server Instance	Instance Name	pr-2.example.com
	Instance Directory	/opt/SUNWwbsvr/https-pr-2.example.com
	Port	1080
	Service URL	http://pr-2.example.com:1080
Web Policy Agent for Sun Java System Web Server	Web Agent Profile Name	webagent-2
	Web Agent Profile Password	webagent2

TABLE D–3 Load Balancer for the Web Policy Agents

Component	Description
URL	lb-4.example.com
Method	Round Robin
Protected Servers	pr-1.example.com:1080
	pr-2.example.com:1080
Virtual Servers	lb-4.example.com:90
Monitors	pr-1.example.com:1080
	pr-2.example.com:1080

TABLE D–4 Load Balancer for the J2EE Policy Agents

Component	Description
URL	lb-5.example.com
Method	Round Robin
Protected Servers	pr-1.example.com:1081
	pr-2.example.com:1081
Virtual Servers	lb-5.example.com:91

TABLE D–4 Load Balancer for the J2EE Policy Agents *(Continued)*

Component	Description
Monitors	pr-1.example.com:1081
	pr-2.example.com:1081

APPENDIX E

Message Queue Servers

This appendix collects the information regarding the Message Queue host machines. It contains the following tables:

- Message Queue 1 Host Machine
- Message Queue 2 Host Machine

TABLE E–1 Message Queue 1 Host Machine

Component	Description	
Host Name	mq-1.example.com	
Sun Java System Message Queue	Session Tools Scripts Directory	/export/SFO/ssoSessionTools/opensso
	Message Queue Directory	/export/SFO/ssoSessionTools/jmq
	Berkeley Database Directory	/tmp/amsession/sessiondb
Message Queue Broker Instance	Name	msgqbroker
	Port	7777
	Instance User	msgquser
	Instance User Password	m5gqu5er
	Database URL	http://mq-1.example.com:7777

TABLE E–2 Message Queue 2 Host Machine

Component	Description	
Host Name	mq-2.example.com	
Sun Java System Message Queue	Session Tools Scripts Directory	/export/SFO/ssoSessionTools/opensso
	Message Queue Directory	/export/SFO/ssoSessionTools/jmq
	Berkeley Database Directory	/tmp/amsession/sessiondb
Message Queue Broker Instance	Name	msgqbroker
	Port	7777
	Instance User	msgquser
	Instance User Password	m5gqu5er
	Database URL	http://mq-2.example.com:7777

APPENDIX F

Known Issues and Limitations

The issues in this appendix will be updated as more information becomes available.

TABLE F–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/`.