

Oracle® Application Server Single Sign-On

Administrator's Guide

10g Release 2 (10.1.2)

Part No. B14078-01

December 2004

Oracle Application Server Single Sign-On Administrator's Guide, 10g Release 2 (10.1.2)

Part No. B14078-01

Copyright © 1996, 2004, Oracle. All rights reserved.

Primary Author: Henry Abrecht

Contributor: Gaurav Bhatia, Kamalendu Biswas, Margaret Chou, Lee Cooper, Mike Hwa, Ganesh Kirti, Peifung Eric Lam, Jeffrey Levinger, Andrew Maywah, Mark Nelson, Saurabh Shrivastava, Arun Swaminathan, Huiping Wang, Tim Willard

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

This product includes software developed by the Apache Group for use in the Apache HTTP server project (<http://www.apache.org/>).

Contents

Send Us Your Comments	xi
Preface	xiii
Intended Audience	xiii
Documentation Accessibility	xiii
Structure.....	xiv
Related Documents	xv
Conventions	xvi
1 Components and Processes: an Overview	
Key Components in the Single Sign-On System	1-1
Single Sign-On Server	1-1
Partner Applications	1-2
External Applications	1-2
mod_osso	1-2
Oracle Internet Directory	1-3
Oracle Identity Management Infrastructure	1-3
Single Sign-On Processes	1-3
Accessing the Single Sign-On Server.....	1-3
Accessing a Partner Application.....	1-3
Accessing an External Application.....	1-4
Accessing the External Applications Portlet in OracleAS Portal	1-4
Authenticating to an External Application for the First Time.....	1-4
Authenticating to an External Application After the First Time	1-5
Logging Out of an External Application	1-5
Single Sign-Off	1-5
Changing Passwords	1-6
Global User Inactivity Timeout.....	1-6
Signing On Using the Wireless Option.....	1-6
2 Basic Administration	
The Single Sign-On Administrator’s Role	2-1
Granting Administrative Privileges	2-1
Changing the Single Sign-On Administration Group	2-3

policy.properties	2-4
Stopping and Starting Single Sign-On Components	2-4
Using the Application Server Control Console	2-4
Using the Command Line	2-5
Stopping and Starting the Oracle HTTP Server	2-5
Stopping and Starting the OC4J_SECURITY Instance	2-5
Stopping and Starting the Single Sign-On Middle Tier.....	2-5
Stopping and Starting All Components	2-6
Setting Browser Preferences for OracleAS Single Sign-On	2-6
Accessing the Administration Pages	2-7
Using the Edit SSO Server Page to Configure the Server	2-8
Configuring Globalization Support	2-8
Configuring the Global User Inactivity Timeout	2-9
Obtaining the Sample Files	2-11

3 Directory-Enabled Single Sign-On

Managing Users in Oracle Internet Directory	3-1
Password Policies	3-2
Password Rules	3-2
Configuring Password Life.....	3-2
Change Password Page Behavior	3-2
Password Has Expired	3-2
Password Is About to Expire.....	3-2
Grace Login Is in Force.....	3-3
Force Change Password.....	3-3
Configuring Account Lockout	3-3
Unlocking Users	3-3
Configuring Password Policies	3-3
Directory Tree for OracleAS Single Sign-On	3-3
Changing Single Sign-On Server Settings for Directory Access	3-5
Updating the Single Sign-On Server with Directory Changes	3-5

4 Configuring and Administering Partner Applications

Registering a Partner Application: What It Means	4-1
Registering mod_osso	4-1
Syntax and Parameters for ssoereg.....	4-2
Command Example	4-4
Restarting the Oracle HTTP Server	4-4
Deploying Multiple Partner Applications with a Load Balancer	4-4
Usage Scenario.....	4-4
Configuration Steps	4-5
Installing the Partner Applications	4-6
Configuring the Oracle HTTP Servers on the Partner Application Middle Tiers	4-6
Configuring the HTTP Load Balancer	4-7
Reregistering mod_osso on the Partner Application Middle Tiers	4-7
Configuring mod_osso with Virtual Hosts	4-8

5	Configuring and Administering External Applications	
	Using the Interface to Deploy and Manage External Applications.....	5-1
	Adding an External Application.....	5-1
	Editing an External Application	5-4
	Storing External Application Credentials in the Single Sign-On Database.....	5-4
	Proxy Authentication for Basic Authentication Applications.....	5-4
	Configuring the Oracle HTTP Server as a Proxy for Basic Authentication.....	5-5
	Configuration Requirements	5-6
	Configuration Steps	5-6
6	Multilevel Authentication	
	What Is Multilevel Authentication?	6-1
	How Multilevel Authentication Works	6-1
	Components of a Multilevel System	6-2
	Authentication Levels.....	6-2
	Authentication Plugins.....	6-3
	Configuring Multilevel Authentication.....	6-3
	Usage Scenario.....	6-3
	Configuration Steps	6-4
7	Signing On with Digital Certificates	
	How Certificate-Enabled Authentication Works.....	7-1
	System Requirements.....	7-2
	Configuring the Single Sign-On System for Certificates	7-2
	Oracle HTTP Server	7-2
	Setting SSL Parameters.....	7-2
	Choosing a Certificate Authority.....	7-3
	Single Sign-On Server.....	7-3
	Configure policy.properties with the Default Authentication Plugin	7-4
	Modify the Configuration File for the Authentication Plugin (Optional).....	7-4
	Customize the User Name Mapping Module (Optional)	7-5
	Restart the Single Sign-On Middle Tier.....	7-6
	Oracle Internet Directory	7-6
	Maintaining a Certificate Revocation List.....	7-7
8	Advanced Deployment Options	
	Enabling SSL.....	8-1
	Enable SSL on the Single Sign-On Middle Tier	8-2
	Reconfigure the Identity Management Infrastructure Database	8-3
	Change Single Sign-On URLs	8-3
	Update targets.xml.....	8-4
	Protect Single Sign-On URLs.....	8-4
	Restart the Oracle HTTP Server and the Single Sign-On Middle Tier	8-5
	Reregister Partner Applications.....	8-5
	Deployment Scenarios	8-5

One Single Sign-On Middle Tier, One Oracle Internet Directory	8-6
Multiple Single Sign-On Middle Tiers, One Oracle Internet Directory	8-7
Usage Scenario.....	8-7
Configuration Steps	8-8
Multiple Single Sign-On Middle Tiers, Replicated Oracle Internet Directory	8-13
Multiple, Geographically Distributed Single Sign-On Instances	8-13
Usage Scenario.....	8-13
Configuration Steps	8-14
Other High Availability Deployments.....	8-15
OracleAS Cold Failover Cluster (Infrastructure)	8-15
Disaster Recovery.....	8-15
Backup and Recovery	8-15
Replicating the Identity Management Database	8-15
The Replication Mechanism	8-16
Configuring the Identity Management Database for Replication.....	8-17
Adding a Node to a Replication Group,.....	8-18
Deleting a Node from a Replication Group	8-18
Deploying OracleAS Single Sign-On with a Proxy Server.....	8-18
Turn Off IP Checking.....	8-19
Enable the Proxy Server	8-19
Setting Up Directory Synchronization for User Nickname Changes	8-20

9 Enabling Support for Application Service Providers

Application Service Providers: Deciding to Deploy Multiple Realms	9-1
Setting Up and Enabling Multiple Realms	9-1
How the Single Sign-On Server Enables Authentication to Multiple Realms.....	9-2
Locating Realms in Oracle Internet Directory	9-2
Validating Realm-Affiliated Users to Partner Applications	9-3
Configuring the Single Sign-On Server for Multiple Realms.....	9-4
Granting Administrative Privileges for Multiple Realms	9-7

10 Monitoring the Single Sign-On Server

Accessing the Monitoring Pages	10-1
Interpreting and Using the Home Page on the Standalone Console	10-1
Interpreting and Using the Details of Login Failures Page	10-3
Updating the Port Property for the Single Sign-On Monitoring Target.....	10-3
Using the OracleAS Web Cache Instance to Monitor the Server.....	10-4
Monitoring a Single Sign-On Server Enabled for SSL.....	10-4

11 Creating Deployment-Specific Pages

How the Single Sign-On Server Uses Deployment-Specific Pages.....	11-1
How to Write Deployment-Specific Pages.....	11-2
Login Page Parameters	11-3
Forgot My Password.....	11-4
Change Password Page Parameters	11-4
Single Sign-Off Page Parameters	11-5

Page Error Codes	11-6
Login Page Error Codes	11-6
Change Password Page Error Codes	11-6
Adding Globalization Support	11-7
Deciding What Language to Display the Page In	11-7
Use the Accept-Language Header to Determine the Page	11-7
Use Page Logic to Determine the Language	11-7
Rendering the Page.....	11-8
Guidelines for Deployment-Specific Pages	11-8
Installing Deployment-Specific Pages	11-9
Using policy.properties to Install Login and Change Password Pages	11-9
Using policy.properties to Install Wireless Login and Change Password Pages	11-9
Using WSSO_LS_CONFIGURATION\$ to Install the Single Sign-Off Page.....	11-9
Examples of Deployment-Specific Pages	11-10

12 Integrating with Third-Party Access Management Systems

How Third-Party Access Management Works	12-1
Scenario 1: The user has not yet authenticated to the third-party server	12-2
Scenario 2: The user has already authenticated to the third-party server	12-3
Synchronizing the Third-Party Repository with Oracle Internet Directory	12-3
Third-Party Integration Modules	12-3
Using Vendor-Supplied Packages	12-4
Building Your Own Package	12-4
Guidelines for Using the Interfaces.....	12-4
The Interfaces.....	12-4
Configuration Steps	12-5
Logging Out of the Integrated System.....	12-6
Integration Case Study: SSOAcme	12-7
Sample Integration Package	12-7
Migrating the Release 9.0.2 Sample Implementation to Release 10.1.2.....	12-9
New Authentication Interface.....	12-9
Get User Name from HTTP Header.....	12-10
Error Handling if User Name Not Present.....	12-10
Return User Name to Single Sign-On Server.....	12-10

13 Exporting and Importing Data

What's Exported and Imported?	13-1
Export and Import Script: Syntax and Parameters	13-1
Script Syntax	13-1
Script Parameters	13-2
Exporting Data from One Server to Another	13-3
Export and Import Scenarios and Script Examples.....	13-3
Export Scenarios.....	13-3
Import Scenarios	13-4
Running the Script	13-4
Verifying That Export and Import Succeeded	13-5

Consolidating Multiple Servers	13-5
Error Messages	13-5

A Troubleshooting

Log Files	A-1
Error Messages and Other Problems	A-2
Basic Error Messages and Problems	A-2
Certificate Authentication.....	A-6
Debugging Certificate Sign-On.....	A-6
Error Messages	A-7
Password Policies.....	A-8
Type 41400 Errors.....	A-8
Increasing the Debug Level.....	A-10
Enabling the Debug Option in the Single Sign-On Database	A-10
Enabling LDAP Tracing for UI Operations	A-11
Managing Single Sign-On Audit Records.....	A-12
Refreshing the LDAP Connection Cache	A-13
Restarting OC4J After Modifying Oracle Internet Directory	A-13
Troubleshooting Replication.....	A-13
Verifying Oracle9i Advanced Replication Configuration.....	A-14
Verifying and Rectifying Oracle9i Advanced Replication Configuration.....	A-14
A Word About Non-GET Authentication	A-14
Need More Help?	A-14

B Obtaining the Single Sign-On Schema Password

Using the Command Line	B-1
Using Oracle Directory Manager	B-2

C policy.properties

Glossary

Index

List of Figures

1-1	Single Sign-On with mod_osso	1-4
2-1	iASAdmins Tab of Oracle Directory Manager	2-3
2-2	Using the Console to Restart the Single Sign-On Middle Tier	2-5
2-3	SSO Server Administration Page	2-7
3-1	Directory Information Tree for OracleAS Single Sign-On	3-4
4-1	Load Balancer with Multiple Partner Applications	4-5
5-1	External Application Login Page	5-4
5-2	Authentication Flow Using mod_osso/mod_proxy	5-5
6-1	Multilevel Authentication Flow	6-2
7-1	Certificate-Enabled Single Sign-On	7-1
8-1	Default Single Sign-On Installation: One Computer	8-6
8-2	Single Sign-On Installation: Two Computers	8-6
8-3	Two Single Sign-On Middle Tiers, One Oracle Internet Directory	8-8
8-4	A Highly Available, Geographically Distributed Single Sign-On System	8-14
8-5	Multimaster Replication Architecture	8-16
9-1	The Big Picture: Single Sign-On in Multiple Realms	9-3
9-2	mod_osso Headers for Users with the Same Name	9-4
10-1	Monitoring Home Page for OracleAS Single Sign-On	10-2
10-2	Details of Login Failures Page	10-3
12-1	Accessing Oracle Partner Applications Using a Third-Party Server	12-2

List of Tables

2-1	SSO Session Policy	2-8
5-1	External Application Login	5-2
5-2	Authentication Method	5-2
5-3	Additional Fields.....	5-2
6-1	Default Authentication Levels	6-3
7-1	HTTP Parameters for Certificate-Enabled Single Sign-On	7-3
8-1	Parameters for ssoReplSetup.....	8-17
9-1	Parameters for enblhstg.csh and addsub.csh.....	9-6
11-1	Login Page Parameters Submitted to the Page by the Single Sign-On Server	11-3
11-2	Login Page Parameters Submitted by the Page to the Single Sign-On Server	11-3
11-3	Change Password Parameters Submitted to the Page.....	11-4
11-4	Change Password Page Parameters Submitted by the Page	11-5
11-5	Parameters Submitted to the Single Sign-Off Page.....	11-5
11-6	Login Page Error Codes	11-6
11-7	Change Password Page Error Codes.....	11-6
13-1	Parameters Passed to ssomig	13-2
13-2	Error Codes for Export and Import	13-5
A-1	Parameters for the Replication Environment Management Tool	A-14

Send Us Your Comments

Oracle Application Server Single Sign-On Administrator's Guide, 10g Release 2 (10.1.2)

Part No. B14078-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appserverdocs_us@oracle.com
- FAX: (650) 506-7375. Attn: Oracle Application Server Documentation Manager
- Postal service:

Oracle Corporation
Oracle Application Server Documentation
500 Oracle Parkway, M/S 10p6
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Oracle Application Server Single Sign-On Administrator's Guide contains concepts and procedures for managing user authentication to Oracle Application Server (OracleAS). The material presented in this book applies to UNIX and Windows platforms.

This preface contains these topics:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Structure](#)
- [Related Documents](#)
- [Conventions](#)

Intended Audience

Oracle Application Server Single Sign-On Administrator's Guide is intended for the following users:

- Administrators charged with configuring and managing authentication to OracleAS.
- Developers of features for which OracleAS Single Sign-On is the authentication mechanism. The book is particularly for those who want to integrate these features with `mod_osso`, an authentication module on the Oracle HTTP Server.
- Anyone who wants to understand how to use OracleAS Single Sign-On to protect access to Web applications.

This document assumes that the reader has a rudimentary knowledge of OracleAS and has installed, or is able to install, release 10.1.2.

Documentation Accessibility

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

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Structure

This book has the following structure:

Chapter 1, "Components and Processes: an Overview"

Takes a high-level, abbreviated look at salient aspects of OracleAS Single Sign-On. Intended as a quick reference.

Chapter 2, "Basic Administration"

Examines essential administration tasks such as stopping and starting the single sign-on server, enabling applications for single sign-on, and assigning administrative privileges.

Chapter 3, "Directory-Enabled Single Sign-On"

Examines the role that Oracle Internet Directory plays in single sign-on. The directory is the native repository for OracleAS users. As such, it plays a key role in user management.

Chapter 4, "Configuring and Administering Partner Applications"

Explains how to register partner applications with the single sign-on server. Shows how to deploy multiple partner applications with a load balancer.

Chapter 5, "Configuring and Administering External Applications"

Explains how to use the single sign-on UI to add and delete external applications. Shows how to configure these applications for proxy authentication using the Oracle HTTP Server.

Chapter 6, "Multilevel Authentication"

Explains how to assign specific authentication levels and adapters to specific applications. This feature enables you to tailor authentication behavior to the security needs of the application.

Chapter 7, "Signing On with Digital Certificates"

Explains how to configure OracleAS Single Sign-On to use X.509 certificates over SSL.

Chapter 8, "Advanced Deployment Options"

Presents nondefault ways to configure OracleAS Single Sign-On. Shows how to deploy the single sign-on server in ways that make it more available. Other topics include SSL-enabled single sign-on and single sign-on using proxy servers.

Chapter 9, "Enabling Support for Application Service Providers"

Explains how OracleAS Single Sign-On enables multiple identity management realms to be deployed within one instance of the Oracle Identity Management infrastructure. Shows how to enable the server for login to multiple realms.

Chapter 10, "Monitoring the Single Sign-On Server"

Explains how to use Oracle Enterprise Manager, the Oracle system management console, to monitor server load and user activity.

Chapter 11, "Creating Deployment-Specific Pages"

Explains how single sign-on pages are invoked. Explains how to rework these pages to suit enterprise needs. Includes guidelines for globalizing pages.

Chapter 12, "Integrating with Third-Party Access Management Systems"

Explains how to integrate OracleAS Single Sign-On with a third-party single sign-on system. By integrating, the third-party system gains access to the OracleAS product complement. Includes a fictional case study.

Chapter 13, "Exporting and Importing Data"

Explains how to move data between two or more single sign-on servers. Uses different scenarios to describe the conditions under which data must be moved.

Appendix A, "Troubleshooting"

Provides tips for handling error messages and other problems. Groups error messages and problems by feature. Also lists and describes the single sign-on log files.

Appendix B, "Obtaining the Single Sign-On Schema Password"

Provides an LDAP command that returns the single sign-on schema password. You need this password to run single sign-on scripts.

Appendix C, "policy.properties"

Provides the `policy.properties` file in its entirety. This is a multipurpose configuration file that contains basic parameters. It is used to configure multilevel authentication.

Glossary

Defines terms used in the book.

Related Documents

For more information, see these Oracle resources:

- *Oracle Identity Management Application Developer's Guide*
- *Oracle Internet Directory Administrator's Guide*

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://www.oracle.com/technology/membership>

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://www.oracle.com/technology/documentation/>

To keep abreast of the latest developments in OracleAS Single Sign-On, see the following link:

http://www.oracle.com/technology/products/id_mgmt/osso/index.html

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.

Convention	Meaning	Example
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter <code>sqlplus</code> to open SQL*Plus. The password is specified in the <code>orapwd</code> file. Back up the datafiles and control files in the <code>/disk1/oracle/dbs</code> directory. The <code>department_id</code> , <code>department_name</code> , and <code>location_id</code> columns are in the <code>hr.departments</code> table. Set the <code>QUERY_REWRITE_ENABLED</code> initialization parameter to <code>true</code> . Connect as <code>oe</code> user. The <code>JRepUtil</code> class implements these methods.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the <code>parallel_clause</code> . Run <code>Uold_release.SQL</code> where <code>old_release</code> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	<code>DECIMAL (digits [, precision])</code>
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	<code>{ENABLE DISABLE}</code>
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	<code>{ENABLE DISABLE}</code> <code>[COMPRESS NOCOMPRESS]</code>
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> That we have omitted parts of the code that are not directly related to the example That you can repeat a portion of the code 	<code>CREATE TABLE ... AS subquery;</code> <code>SELECT col1, col2, ... , coln FROM employees;</code>

Convention	Meaning	Example
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	<pre>SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.</pre>
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	<pre>acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;</pre>
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	<pre>CONNECT SYSTEM/system_password DB_NAME = database_name</pre>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	<pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre>
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - HOME_NAME > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	<pre>c:\winnt\"system32 is the same as C:\WINNT\SYSTEM32</pre>

Convention	Meaning	Example
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.	C:\oracle\oradata>
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	C:\> net start OracleHOME_NAME\TNSListener
ORACLE_HOME and ORACLE_BASE	In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level ORACLE_HOME directory. For Windows NT, the default location was C:\orant. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level ORACLE_HOME directory. There is a top level directory called ORACLE_BASE that by default is C:\oracle. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\orann, where nn is the latest release number. The Oracle home directory is located directly under ORACLE_BASE. All directory path examples in this guide follow OFA conventions. Refer to <i>Oracle Database Platform Guide for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Go to the ORACLE_BASE\ORACLE_HOME\rdbms\admin directory.

Components and Processes: an Overview

OracleAS Single Sign-On enables you to use a single user name and password and, optionally, realm ID, to log in to all features of OracleAS as well as to other Web applications.

OracleAS Single Sign-On provides the following benefits:

- Reduced administrative costs
The single sign-on server eliminates the need to support multiple accounts and passwords.
- Convenient login
Users do not have to maintain a separate user name and password for each application that they access.
- Increased security
When a password is required only once, users are less likely to use simple, easily exposed passwords or to write these passwords down.

This chapter contains the following topics:

- [Key Components in the Single Sign-On System](#)
- [Single Sign-On Processes](#)

Key Components in the Single Sign-On System

OracleAS Single Sign-On interacts with the following components:

- [Single Sign-On Server](#)
- [Partner Applications](#)
- [External Applications](#)
- [mod_osso](#)
- [Oracle Internet Directory](#)
- [Oracle Identity Management Infrastructure](#)

Single Sign-On Server

The single sign-on server consists of program logic in the OracleAS database, Oracle HTTP Server, and OC4J server that enables you to log in securely to applications such as expense reports, mail, and benefits. These applications take two forms: partner

applications and external applications. In both cases, you gain access to several applications by authenticating only once.

Partner Applications

OracleAS applications delegate the authentication function to the single sign-on server. For this reason, they are called partner applications. An authentication module called `mod_osso` enables these applications to accept authenticated user information instead of a user name and password once you have logged in to the single sign-on server.

A partner application is responsible for determining whether a user authenticated by OracleAS Single Sign-On is authorized to use the application.

Examples of partner applications include OracleAS Portal, OracleAS Discoverer, and Oracle Delegated Administration Services.

External Applications

External applications do not delegate authentication to the single sign-on server. Instead, they display HTML login forms that ask for application user names and passwords. Each external application may require a unique user name and password. Yahoo! Mail is an example of an external application that uses HTML login forms.

You can configure the single sign-on server to provide user names and passwords to external applications on users' behalf once they have logged in to the single sign-on server. Users have the option of storing application credentials in the single sign-on database. The server uses the single sign-on user name to locate and retrieve application names and passwords and to log the user in. To save these credentials, the user selects the **Remember My Login Information For This Application** check box when first logging in.

`mod_osso`

`mod_osso` is an Oracle HTTP Server module that provides authentication to OracleAS applications. It replaces the single sign-on SDK, used in earlier releases of OracleAS Single Sign-On to integrate partner applications. Located on the application server, `mod_osso` simplifies the authentication process by serving as the sole partner application to the single sign-on server. In this way, `mod_osso` renders authentication transparent to OracleAS applications. The administrator for these applications is spared the burden of integrating them with an SDK.

After authenticating a user, `mod_osso` transmits the simple header values that applications may use to authorize the user:

- User name
- User GUID
- Language and territory

To learn more about the attributes that the single sign-on server passes to `mod_osso`, see the chapter about `mod_osso` in *Oracle Identity Management Application Developer's Guide*. This chapter explains how to develop applications for single sign-on.

`mod_osso` works only with the Oracle HTTP listener. You can use OracleAS SSO Plug-in to protect applications that work with third-party listeners such as Sun One and IIS. To learn how to use OracleAS SSO Plug-in, see the appendix about this tool in *Oracle HTTP Server Administrator's Guide*.

Oracle Internet Directory

Oracle Internet Directory is the repository for all single sign-on user accounts and passwords—administrative and nonadministrative. The single sign-on server authenticates users against their entries in the directory. At the same time, it retrieves user attributes from the directory that enable applications to validate users.

Oracle Identity Management Infrastructure

OracleAS Single Sign-On is just one link in an integrated infrastructure that also includes Oracle Internet Directory, Oracle Directory Integration and Provisioning, Oracle Delegated Administration Services, and OracleAS Certificate Authority. Working together, these components, called the Oracle Identity Management infrastructure, manage the security life cycle of users and other network entities in an efficient, cost-effective way.

To learn more about the benefits of Oracle Identity Management, see *Oracle Identity Management Concepts and Deployment Planning Guide*.

Single Sign-On Processes

This section describes the following processes:

- [Accessing the Single Sign-On Server](#)
- [Accessing an External Application](#)
- [Single Sign-Off](#)
- [Changing Passwords](#)
- [Global User Inactivity Timeout](#)
- [Signing On Using the Wireless Option](#)

Accessing the Single Sign-On Server

Nonadministrative users first gain access to the single sign-on server by entering the URL of a partner application such as OracleAS Portal. Entering such a URL invokes the single sign-on login screen. Once they have entered the correct user name and password, users gain access to other partner applications and to external applications without having to provide credentials again.

Administrative users can access the administration home page for single sign-on by typing a URL of this form:

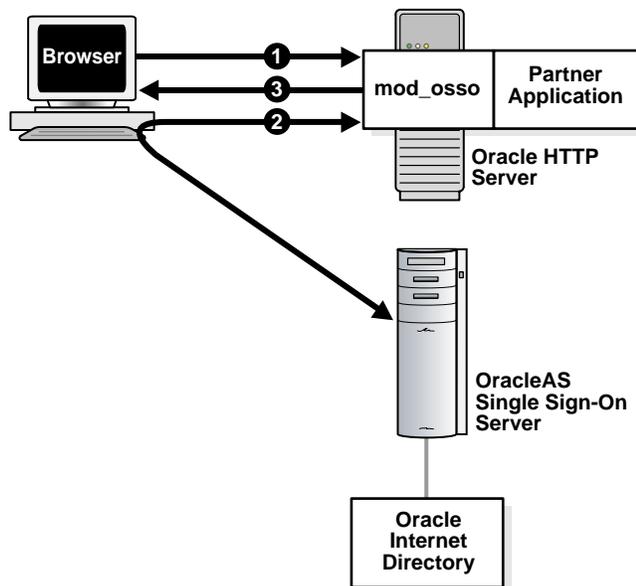
```
http://host:port/pls/orasso
```

where *host* is the computer where the single sign-on server is located, *port* is the port number of the server, and *orasso* is the database access descriptor for the single sign-on schema. If the server is enabled for SSL, `https` must be substituted for `http`. If the port number is 80 or 443 (SSL), it may be omitted from the URL. These numbers are the defaults.

Accessing a Partner Application

Figure 1–1 shows what happens when the user requests the URL of a partner application that is protected by `mod_osso`.

Figure 1–1 Single Sign-On with mod_osso



1. Users try to access a partner application.
2. Users are redirected to the single sign-on server. The server challenges them for their credentials. After verifying these credentials in Oracle Internet Directory, the server passes these credentials on to the partner application.
3. The application serves up the requested content.

Accessing an External Application

External applications are available through OracleAS Portal, a single sign-on partner application.

This section contains these topics:

- [Accessing the External Applications Portlet in OracleAS Portal](#)
- [Authenticating to an External Application for the First Time](#)
- [Authenticating to an External Application After the First Time](#)
- [Logging Out of an External Application](#)

Accessing the External Applications Portlet in OracleAS Portal

To gain access to an external application, you select the External Applications portlet on the OracleAS Portal home page; then, from the list of external applications that appears, you select an application.

Authenticating to an External Application for the First Time

Selecting an application in the External Applications portlet initiates the external application login procedure. The following occurs if you are accessing the application for the first time:

1. The external application login procedure checks the single sign-on password store for your credentials. If it finds no credentials, the single sign-on server prompts you for them.

2. You enter your user name and password. You can save these credentials in the password store by selecting the **Remember My Login Information** check box on the application login screen.
3. If you elect to save your credentials in the password store, the server uses these credentials to construct a login form to submit to the login processing routine of the application. This routine has been preconfigured by the administrator and is associated with the requested application.
4. The server sends the form to the client browser, with a directive to post it immediately to the external application.
5. The client posts the form to the external application and logs you in.

If you decline to save your credentials in the password store, you must enter a user name and password each time that you log in.

Authenticating to an External Application After the First Time

If you saved your credentials when accessing an external application for the first time, the single sign-on server retrieves your credentials for you during subsequent logins. The process works like this:

1. You click one of the links in the External Applications portlet of OracleAS Portal.
2. The external application login procedure checks the password store for your credentials.
3. The single sign-on server finds your credentials and uses them to construct a login form to submit to the login processing routine of the application. This routine has been preconfigured by the administrator and is associated with the requested application.
4. The server sends the form to the client browser, with a directive to post it immediately to the external application.
5. The client posts the form to the external application and logs you in.

Logging Out of an External Application

Unlike partner applications, external applications do not cede logout control to the single sign-on server. It is the user's responsibility to log out of each of these applications.

Single Sign-Off

You can terminate a single sign-on session and log out of all active partner applications simultaneously by logging out of whatever application you are working in. Clicking **Logout** in a partner application takes you to the single sign-off page, where logout occurs.

If you signed off successfully, each of the applications listed on the single sign-off page has a check mark next to the application name. A broken image next to an application name denotes an unsuccessful logout.

Once all of the application names activated in a session have a check mark, you can click **Return** to go to the application from which you initiated logout.

Changing Passwords

The change password screen appears only when your password is about to expire and you fall within a grace login period. If the password is still valid, you can click **Cancel** on this screen and proceed with the login.

To change or reset a password under other circumstances, the nonadministrative user must go to Oracle Delegated Administration Services, a service of Oracle Internet Directory that performs user and group management functions.

The Oracle Delegated Administration Services home page is found at a URL of the following form:

```
http://host:port/oiddas/
```

where *host* is the name of the computer where Oracle Delegated Administration Services is located, and *port* is the port number of this server. Oracle Delegated Administration Services and OracleAS Single Sign-On generally have the same host name. If the Oracle HTTP Server hosting Oracle Delegated Administration Services and OracleAS Single Sign-On is enabled for SSL, `https` must be substituted for `http`. The port number may be omitted if it is 80 or 443 (SSL) because these numbers are the defaults.

Note: Unlike single sign-on user names, single sign-on passwords are case sensitive and must conform to the Oracle Internet Directory realms that users belong to.

Global User Inactivity Timeout

The global user inactivity timeout is a feature that enables applications to force you to reauthenticate if you have been idle for a preconfigured amount of time. This timeout is a useful feature for sensitive applications that require a shorter user inactivity timeout than the single sign-out session timeout.

When you exceed the global user inactivity timeout limit and try to access the application, the application sends the single sign-on server an authentication request as usual. The server, ascertaining that you have exceeded the timeout limit, prompts you to log in. If you have not exceeded the limit, the server uses the session cookie to authenticate you.

Note: You may have a valid single sign-on session, but if you have exceeded the global timeout limit, the server prompts you for credentials.

See Also: ["Configuring the Global User Inactivity Timeout"](#) in Chapter 2, "Basic Administration"

Signing On Using the Wireless Option

You can use mobile, or wireless, devices such as personal digital assistants, cellular phones, and voice recognition systems to access OracleAS applications. As in PC-based systems, the authentication mechanism is OracleAS Single Sign-On. You can select the wireless option when installing OracleAS. If you do, Portal-to-Go, the gateway for mobile devices, is registered with the single sign-on server automatically.

To learn more about OracleAS Wireless see *Oracle Application Server Wireless Administrator's Guide* and *Oracle Application Server Wireless Developer's Guide*.

Basic Administration

This chapter introduces you to the tasks involved in administering single sign-on. The chapter contains the following topics:

- [The Single Sign-On Administrator's Role](#)
- [Granting Administrative Privileges](#)
- [Changing the Single Sign-On Administration Group](#)
- [policy.properties](#)
- [Stopping and Starting Single Sign-On Components](#)
- [Setting Browser Preferences for OracleAS Single Sign-On](#)
- [Accessing the Administration Pages](#)
- [Using the Edit SSO Server Page to Configure the Server](#)
- [Configuring Globalization Support](#)
- [Configuring the Global User Inactivity Timeout](#)
- [Obtaining the Sample Files](#)

The Single Sign-On Administrator's Role

When the single sign-on server is accessed for the first time, only one single sign-on administrator exists: orcladmin, the OracleAS super user. The person installing OracleAS selects the password for this user at install time. The orcladmin account is used to create other accounts, including accounts for iASAdmins, the group that administers single sign-on.

As a single sign-on administrator, you can use the administration pages to do the following:

- Configure server settings
- Administer partner applications
- Administer external applications

Granting Administrative Privileges

To exercise your privileges as a single sign-on administrator, you must be a member of the administrative group iASAdmins. This means that an existing member of this group must add you to it.

To assign a user to iASAdmins:

1. Start Oracle Directory Manager. To learn how to start this tool, see *Oracle Internet Directory Administrator's Guide*.
2. Log in as `cn=orcladmin`, the directory super user. You must use the password that was assigned to this user when Oracle Internet Directory was installed.

Note: The directory superuser `cn=orcladmin` is not the same as the OracleAS super user `orcladmin`. These are separate, hierarchically unequal accounts.

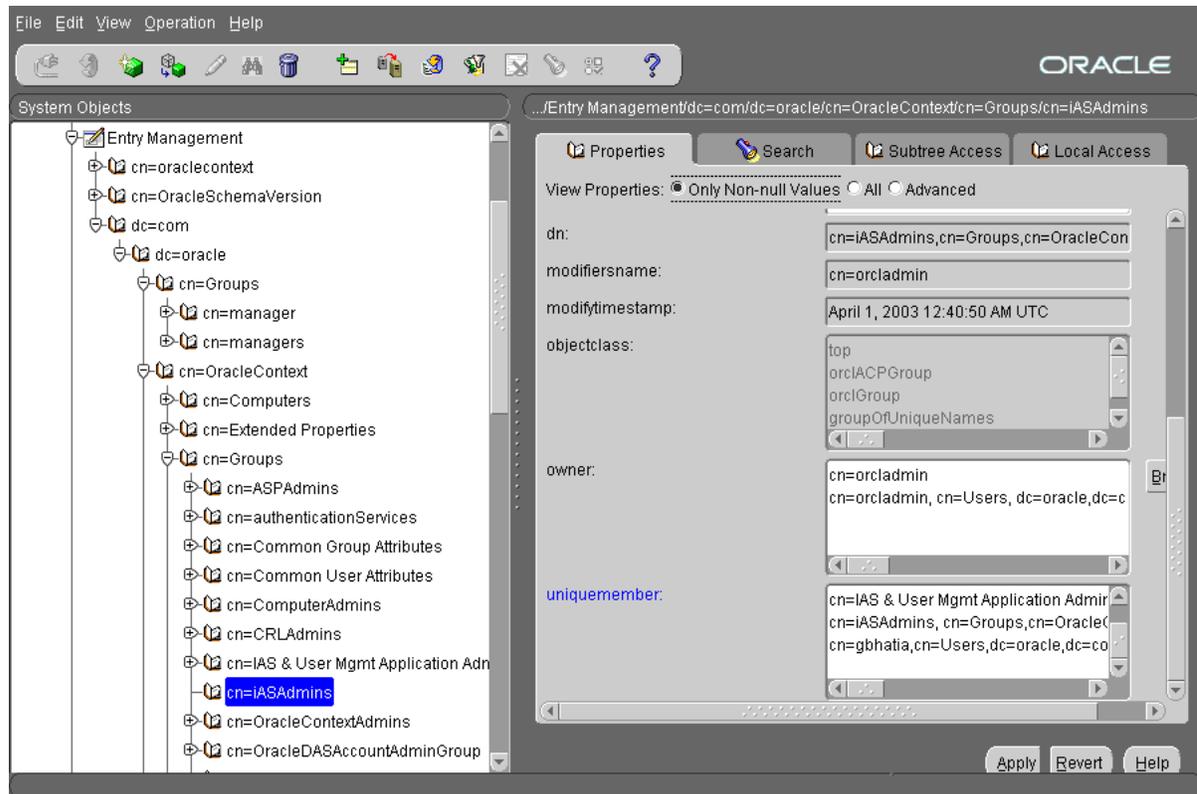
3. In the **System Objects** frame, click in succession the following entries:
 - Entry Management
 - `dc=default_identity_management_realm`
 - `cn=OracleContext`
 - `cn=Groups`
 - `cn=iASAdmins`

For example:

```
cn=iASAdmins,cn=Groups,cn=OracleContext,dc=oracle,dc=com
```

Where `dc=oracle,dc=com` is the default identity management realm. In reality, the default is likely the domain name of your installation.

4. In the **uniquemembers** text box of the **iASAdmins** tab, add an entry for the user. `uniquemembers` is an attribute of the entry `iASAdmins`. As such it defines members of the group `iASAdmins`. Be sure to add the user's full DN. In [Figure 2-1](#) on page 2-3 the user `cn=gbhatia,cn=users,cn=oracle,cn=oracle` has been added.
5. Click **Apply**.

Figure 2–1 *iASAdmins Tab of Oracle Directory Manager*

To create new users, use Oracle Delegated Administration Services. See the chapter about the Oracle Internet Directory Self-Service Console in *Oracle Identity Management Guide to Delegated Administration*.

Changing the Single Sign-On Administration Group

By default, the single sign-on server uses the directory entry `cn=iASAdmins, cn=Groups, cn=OracleContext, default_realm_dn` to determine whether a user has administrative privileges for OracleAS Single Sign-On. You can create a different default administration group by following these steps:

1. Create a new group in the directory relative to the default realm DN. If, for example, your default realm is `dc=us, dc=acme, dc=com`, and you want the default administration group to be called `sso_admins`, create this entry:

```
cn=sso_admins, dc=us, dc=acme, dc=com
```

Use either Oracle Directory Manager or LDAP command line tools to create the entry.

2. In the file `ORACLE_HOME/sso/admin/plsql/sso/ssoidi.pks` update the `IASADMINSGROUP` constant with the RDN that you created in step 1. Following the example given earlier, you would edit the line that contains this constant to look like this:

```
IASADMINSGROUP CONSTANT VARCHAR2(1024) := 'cn=sso_admins';
```

You do not have to include the default realm (`dc=us, dc=acme, dc=com` in the example). The single sign-on server appends the realm when it checks for group membership.

3. Save the file.
4. Load the file into the `orasso` schema:

```
sql> connect orasso/password
sql> @ssoidi.pks
sql> @ssoidi.pkb
```

Once you complete these steps, all users in the group `cn=sso_admins,dc=us,dc=acme,dc=com` can administer the single sign-on server. You may include groups as members too.

policy.properties

`policy.properties` is a multipurpose configuration file for OracleAS Single Sign-On. This file contains basic parameters required by the single sign-on server. The default values of these parameters are adequate for most installations. Hence the file requires no modification out of the box.

`policy.properties` is also used to implement advanced single sign-on features such as multilevel authentication. [Appendix C, "policy.properties"](#) contains a copy of the file. `policy.properties` is also in the single sign-on configuration directory at `ORACLE_HOME/sso/conf`.

Note: When editing `policy.properties`, take care not to insert blank space at the end of each line. After editing the file, restart the single sign-on server. See the section immediately following for instructions.

Stopping and Starting Single Sign-On Components

You can use either the command line or the Oracle Enterprise Manager Application Server Control console to start and stop single sign-on components. The console offers the benefit that you can stop or start several components at once. The command line requires several commands for the same tasks.

Using the Application Server Control Console

Use these steps to stop and start single sign-on components with the console:

1. Go to the standalone console for the infrastructure instance of Oracle Enterprise Manager that you want to administer. This is effected by entering the host name of the computer hosting the OracleAS instance and the port number of Oracle Enterprise Manager. The default port number is 1810, but it may be configured in increments of one, up to 1816.
2. Log in using the credentials of an OracleAS administrator.
3. From the **Standalone Instances** section of the Farm page, choose the appropriate OracleAS instance.
4. From the **System Components** list of the Application Server page, select the check boxes for the components that you want to stop, start, or restart; then click the appropriate button at the top of the list. [Figure 2-2](#) on page 2-5 shows the single sign-on middle tier being restarted. To stop or restart the entire identity management infrastructure, click **Stop All** or **Restart All** at the top of the page.

Figure 2–2 Using the Console to Restart the Single Sign-On Middle Tier

Page Refreshed Sep 30, 2004 6:39:45 PM

General

Status **Up**

Host [isunnae29](#)

Installation Type **Infrastructure_ID**

Oracle Home **/private/latest/oracle/ias1012**

Farm **ias1012.us.oracle.com**

[Stop All](#) [Restart All](#)

CPU Usage

Application Server (0%)
Idle (97%)
Other (3%)

Memory Usage

Application Server (84% 863MB)
Free (16% 161MB)
Other (0% 0MB)

System Components [Enable/Disable Components](#) [Create OC4J Instance](#)

[Start](#) [Stop](#) [Restart](#) [Delete OC4J Instance](#)

Select All | Select None

Select	Name	Status	Start Time	CPU Usage (%)	Memory Usage (MB)
<input checked="" type="checkbox"/>	HTTP_Server	↑	Sep 24, 2004 12:55:50 PM	0.00	70.12
<input checked="" type="checkbox"/>	OC4J_SECURITY	↑	Sep 24, 2004 12:56:03 PM	0.03	172.08
<input type="checkbox"/>	oca	↑	Sep 24, 2004 12:57:06 PM	0.03	149.19
<input type="checkbox"/>	Single_Sign-On:orasso	↑	N/A	N/A	N/A
<input type="checkbox"/>	Management	↑	Sep 24, 2004 12:58:18 PM	0.40	471.45

TIP This table contains only the enabled components of the application server. Only components that have the checkbox enabled can be started or stopped.

Related Links

- [Process Management](#)
- [All Metrics](#)

Using the Command Line

You can issue separate commands to stop and start just the Oracle HTTP Server or the entire single sign-on middle tier. Another command stops and starts just the OC4J_SECURITY instance. Still another command stops and starts all infrastructure components.

Stopping and Starting the Oracle HTTP Server

Issue these two commands to stop and then start the Oracle HTTP Server:

```
ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=HTTP_Server
```

You can also stop and start the server by issuing this command:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

Stopping and Starting the OC4J_SECURITY Instance

Issue these two commands to stop and then start the OC4J_SECURITY instance:

```
ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=OC4J_SECURITY
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=OC4J_SECURITY
```

You can also stop and start the OC4J_SECURITY instance by issuing this command:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Stopping and Starting the Single Sign-On Middle Tier

To stop and then start the single sign-on middle tier, stop and start both the Oracle HTTP Server and the OC4J_SECURITY instance:

```
ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=HTTP_Server
```

```
ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=OC4J_SECURITY
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=OC4J_SECURITY
```

You can also stop and start the single sign-on middle tier by issuing these commands:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Stopping and Starting All Components

Issue the following commands to stop and then start the Oracle HTTP Server, the single sign-on server, OC4J, and Oracle Internet Directory:

```
ORACLE_HOME/opmn/bin/opmnctl stopall
ORACLE_HOME/opmn/bin/opmnctl startall
```

This command assumes that infrastructure components are all in the same Oracle home.

Setting Browser Preferences for OracleAS Single Sign-On

Logging in and out of OracleAS Single Sign-On successfully requires that the following browser settings be in place:

Cache Settings

To enable the correct cache settings:

1. Go to the cache settings dialog box by clicking the following in succession:
 - Internet Explorer:
 - Tools
 - Internet Options
 - General
 - Settings
 - Netscape Communicator:
 - Edit
 - Preferences
 - Advanced
 - Cache
2. Select **Every visit to the page** in Internet Explorer or **Every time** in Netscape Communicator.

Image Settings

To ensure that images are automatically loaded:

1. Click the following in succession:
 - Internet Explorer:
 - Tools
 - Internet Options
 - Advanced
 - Netscape Communicator:
 - Edit

- Preference
 - Advanced
2. Select **Show pictures** in Internet Explorer or **Automatically load images** in Netscape Communicator.

Accessing the Administration Pages

You can use the administration pages within the single sign-on UI to set the single sign-on session length and to enable the server to verify IP addresses. You can also use these pages to administer partner applications and external applications.

To access the administration pages:

1. Enter a URL of the following form:

```
http://host:port/pls/orasso
```

where *host* is the name of computer on which the server is located, *port* is the port number of the server, and *orasso* is the database access descriptor for the single sign-on schema. If the server is enabled for SSL, `https` must be substituted for `http`. The port number may be omitted if it is 80 or 443 (SSL) because these numbers are the defaults.

The **Access Partner Applications** page appears.

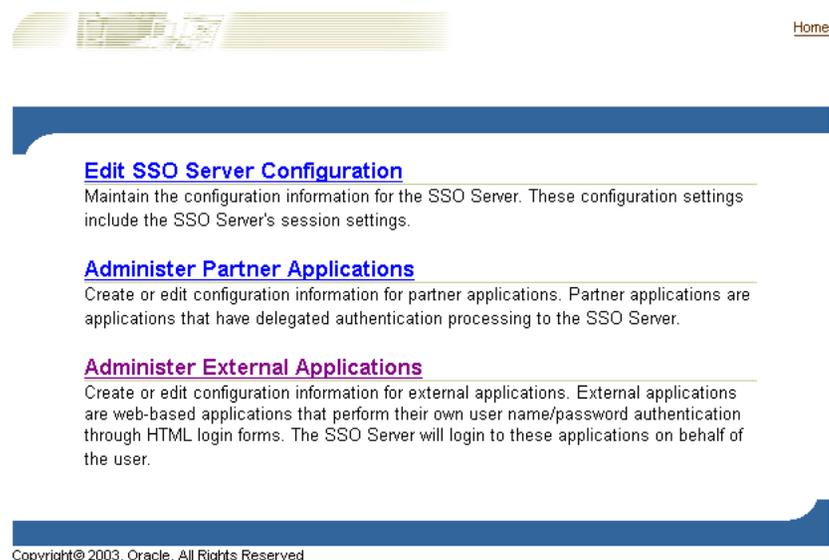
2. Click **Login** in the upper right corner of the **Access Partner Applications** page.

The login page appears.

3. Enter the administrator's user name and password; then click **Login**.
4. The home page appears. To perform administrative functions, click **SSO Server Administration**.

Figure 2–3 shows the **SSO Server Administration** page.

Figure 2–3 SSO Server Administration Page



Using the Edit SSO Server Page to Configure the Server

Use the Edit SSO Server page to fix the length of single sign-on sessions and to verify IP addresses. To access the Edit SSO Server page, click **Edit SSO Server Configuration** on the SSO Server Administration page.

The Edit SSO Server page contains the following heading and fields:

Table 2–1 SSO Session Policy

Field	Description
Single sign-on session duration	Enter the number of hours users can be logged in to the server before their session expires. The default is eight hours.
Verify IP addresses for requests made to the single sign-on server	Select to verify that the IP address of the browser is the same as the IP address in the authentication request. This check box is not selected by default.

If you change either of these parameters, restart the OC4J_SECURITY instance:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Configuring Globalization Support

You can enable the single sign-on UI to be rendered in any language that the user's browser is configured for. English and the language of the operating system are installed when OracleAS is installed. To install additional languages, click the **Product Languages** button on the Select a Product to Install screen. If you forget to install additional languages during the installation of OracleAS, you can still enable the single sign-on UI for additional languages by running the `ossoca.jar` tool.

To enable the single sign-on server for additional languages after installation:

1. Copy the desired language files from the CD home for the Repository Configuration Assistant (REPCD_HOME) to the Oracle home for OracleAS Single Sign-On:

```
cp REPCD_HOME/portal/admin/plsql/nlsres/ctl/lang/*.* ORACLE_
HOME/sso/nlsres/ctl/lang/
```

where *lang* is the desired language code. For example, this value would be `ja` for Japanese. Note that you must create the *lang* directory in the single sign-on home before running `ossoca.jar`.

2. Add `ORACLE_HOME/lib` to the library path environment variable.
3. Issue the following commands:

- UNIX:

```
$ORACLE_HOME/sso/bin/ssoca -jar $ORACLE_HOME/sso/lib/ossoca.jar langinst
lang make_lang_avail $ORACLE_HOME
```

- Windows:

```
set PATH=%ORACLE_HOME%/bin;%PATH%
```

```
%ORACLE_HOME%\jdk\bin\java -Dwindir=SYSTEM_ROOT -jar %ORACLE_
HOME%\sso\lib\ossoca.jar langinst lang make_lang_avail %ORACLE_HOME
```

SYSTEM_ROOT denotes the system root directory for Windows. Written as an environment variable, the system root is %SystemRoot%. By default, the value for this variable is C:\winnt.

For the variable *lang*, substitute the code for the language to be installed. For the variable *make_lang_avail*, substitute 1 if you want to make the language available.

In the following UNIX example, the Korean language is installed:

```
$ORACLE_HOME/jdk/bin/java -jar $ORACLE_HOME/sso/lib/ossoca.jar langinst ko 1
$ORACLE_HOME
```

For a complete list of the language codes supported, see Appendix A in *Oracle Application Server Globalization Guide*.

Configuring the Global User Inactivity Timeout

Before reading this section, read "[Global User Inactivity Timeout](#)" in Chapter 1, "Components and Processes: an Overview."

The global user inactivity timeout is applicable to one domain only. This means that computers enabled for the timeout must reside within the same cookie domain. The applications on these computers use the domain cookie to track user activity. If, for example, you use `login.acme.com` for the single sign on server, other computers in the system must have the `.acme.com` domain in their host name. One of these computers might be `host1.acme.com`. Another might be `host2.acme.com`. In addition, clocks on all of these computers, including the single sign-on server computer, must be synchronized with 10 seconds of one another.

The global user inactivity timeout is not configured by default. You must enable it by running the `ssogito.sql` script. The script is found at `ORACLE_HOME/sso/admin/plsql/sso`. The steps that follow include an example of `ssogito.sql`.

To configure the global user inactivity timeout:

1. Log in to SQL*Plus, using the single sign-on schema name and password. The default schema name is `orasso`. To obtain the password, see [Appendix B](#).
2. Run `ssogito.sql` by entering the following command:

```
SQL> @ssogito.sql
```

A list of fields appears.

3. In the **Enter value for timeout_cookie_domain field**, enter a domain name that is common to all of the applications enabled by the single sign-on server. Be sure to place a period before the domain name.

Note: If this field is left blank, the domain name defaults to the host name of the single sign-on server.

4. In the **Enter value for inactivity period field**, enter the length of the desired inactivity period in minutes.
5. To enable the new settings, press the **Return** or **Enter** key. To cancel the transaction, press the **Return** or **Enter** key twice.

Once you have completed a transaction, the script provides you with a summary of the new timeout settings. Here is an example of `ssogito.sql`:

```
SQL> @ssogito
=====
SSO Server Inactivity Timeout Configuration
=====
Timeout          : DISABLED
Cookie name      : OSSO_USER_CTX
Cookie domain    :
Inactivity period: 15 minutes
Encryption key   : 093D678526DAA66D
Note: timeout cookie domain will be defaulted
to the SSO Server hostname
-----
To disable timeout set inactivity period to 0, (zero)
Press return key twice if you do not want
to change timeout configuration.

PL/SQL procedure successfully completed.

Enter value for timeout_cookie_domain: .oracle.com
Enter value for inactivity_period: 15
Timeout          : ENABLED
New timeout cookie domain: .oracle.com
New inactivity period   : 15 minutes

PL/SQL procedure successfully completed.

No errors.
```

6. Restart the single sign-on middle tier:

```
ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=HTTP_Server

ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=OC4J_SECURITY
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=OC4J_SECURITY
```

7. On the application middle tiers where the inactivity timeout is to be enabled, edit the `mod_osso.conf` file. Make sure that the `OsoIdleTimeout` parameter exists and that it is set to on. The file is at `ORACLE_HOME/Apache/Apache/conf`. Here is an example file with the correct setting:

```
LoadModule osso_module libexec/mod_osso.so
<IfModule mod_osso.c>
    OsoIpCheck off
    OsoIdleTimeout on
    OsoConfigFile /u01/oracleas10g/Apache/Apache/conf/osso/osso.conf
#
#Insert Protected Resources
#
.
.
.
</IfModule>
```

8. Restart the Oracle HTTP Server on the application middle tiers.

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

If Oracle Delegated Administration Services and the single sign-on server are located on the same middle tier, and you want the global user inactivity timeout to apply to the former, perform steps seven and eight on the single sign-on middle tier.

Obtaining the Sample Files

The `ipassample.jar` file contains sample code for single sign-on features such as certificate-enabled sign-on and deployment-specific pages. Use this command to extract the file:

```
ORACLE_HOME/jdk/bin/jar -xvf ORACLE_HOME/sso/lib/ipassample.jar
```

Directory-Enabled Single Sign-On

This chapter examines those aspects of OracleAS Single Sign-On that are dependent upon Oracle Internet Directory. The directory is the repository for all single sign-on user accounts and passwords—administrative and nonadministrative. All user and group management functions are handled by the directory.

Note: Oracle Internet Directory can be configured to authenticate to third-party repositories. To learn more, see *Oracle Identity Management Integration Guide*.

The chapter contains the following topics:

- [Managing Users in Oracle Internet Directory](#)
- [Password Policies](#)
- [Directory Tree for OracleAS Single Sign-On](#)
- [Changing Single Sign-On Server Settings for Directory Access](#)
- [Updating the Single Sign-On Server with Directory Changes](#)

Managing Users in Oracle Internet Directory

Use the following tools to manage single sign-on users:

- Oracle Delegated Administration Services

Oracle Delegated Administration Services is a self-service application that enables administrators to manage users and groups. For example, you can create and delete users and change passwords.

You can access Oracle Delegated Administration Services with a URL of this form:

`http://host:port/oiddas/`

where *host* is the name of the computer on which the Oracle Delegated Administration Services server is located, and *port* is the port number of the server. In a typical infrastructure installation, Oracle Delegated Administration Services and OracleAS Single Sign-On have the same host name.

- Oracle Directory Manager

Oracle Directory Manager is a Java-based tool for managing most functions in Oracle Internet Directory. Use it to configure password policies.

- LDAP Command-Line Tools

You can use command-line tools like `ldapmodify` in place of Oracle Delegated Administration Services and Oracle Directory Manager. These tools operate on text files. They take arguments that use the Lightweight Directory Interchange format.

Password Policies

The single sign-on user password is stored in Oracle Internet Directory as an attribute of the user's entry. Users can change their passwords in the single sign-on UI only when their passwords are about to expire. They may use Oracle Delegated Administration Services for this purpose at any time. The directory administrator can use Oracle Directory Manager to adjust password expiry behavior to suit enterprise needs.

This section covers the following topics:

- [Password Rules](#)
- [Configuring Password Life](#)
- [Change Password Page Behavior](#)
- [Configuring Account Lockout](#)
- [Unlocking Users](#)
- [Configuring Password Policies](#)

Password Rules

Oracle Directory Manager has fields that enable you to specify the minimum number of characters that a password requires. To learn what the defaults are, see the chapter about password policies in *Oracle Internet Directory Administrator's Guide*.

Configuring Password Life

Using either Oracle Directory Manager or LDAP command-line tools, you can configure password life and can specify when users are prompted to change their passwords. You can also configure a grace login period for users. This is a period after which users' passwords have expired. If they neglect to change their passwords within this period, they must have an administrator reset them.

Change Password Page Behavior

Users who try to log in when their passwords have expired or are about to expire experience the following server behavior:

Password Has Expired

Users are shown the password expiry screen. They must contact the directory administrator to have the password reset.

Password Is About to Expire

Users are shown an error message on the login page. They have the option of cancelling the page or changing their passwords. In either case, authentication proceeds in the same manner as it does when the change password page is not thrown.

Grace Login Is in Force

If a grace login period has been configured in the directory, users are presented the change password page after their passwords have expired. They have the option of cancelling the page or changing their passwords. In either case, the authentication sequence is the same as it is for users with valid passwords.

Force Change Password

This feature prompts users to change their password after it has been reset by an administrator. You enable force change password by setting the `pwdMustChange` attribute in the directory entry `cn=pwdpolicyentry, cn=common, cn=products, cn=OracleContext, dc=default_identity_management_realm`. You can use the command-line tool `ldapmodify` for this purpose. The value `TRUE` enables this feature. `FALSE` disables it. See the chapter about password policies in *Oracle Internet Directory Administrator's Guide* to learn how to run the tool.

Configuring Account Lockout

An account lockout occurs when users submit the incorrect user name and password combination more times than is permitted by Oracle Internet Directory. Once they are locked out, they are unable to access the single sign-on server from any number of workstations. By default, lockout occurs after 10 login attempts. Once this limit has been reached, even a valid user name and password combination fails to log a user in.

Because single sign-on user accounts are managed in the directory, the directory administrator determines account lockout policies. Oracle Directory Manager has fields for enabling and disabling lockout and for specifying lockout duration.

The default lockout duration is one day.

Unlocking Users

To learn how to unlock users, see the chapter about password policies in *Oracle Internet Directory Administrator's Guide*.

Configuring Password Policies

To learn how to configure password policies, see the chapter about these policies in *Oracle Internet Directory Administrator's Guide*.

Directory Tree for OracleAS Single Sign-On

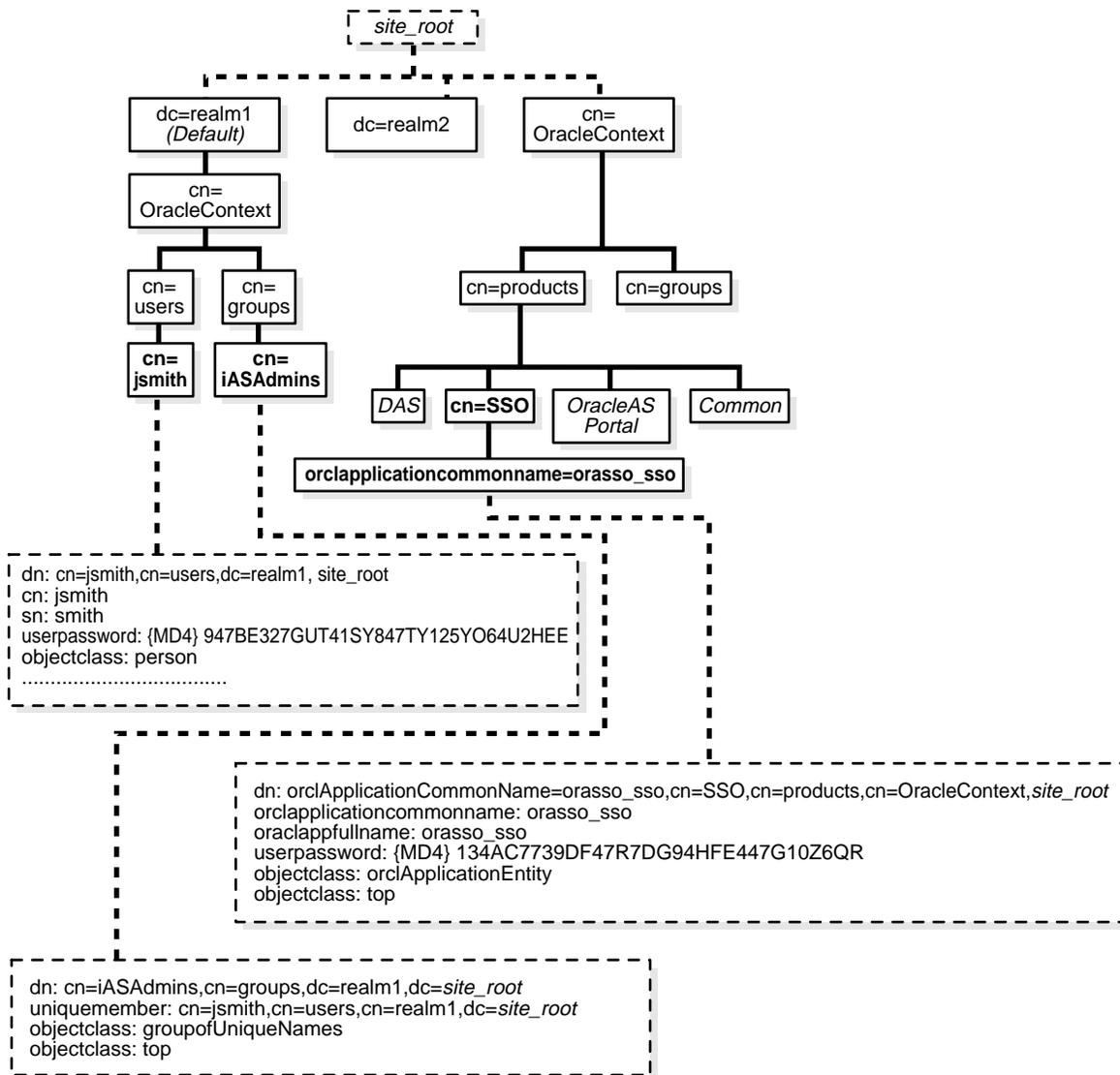
OracleAS Single Sign-On, like other components in the OracleAS complement, has its own "container" within the directory information tree (DIT). This container is found within the Oracle Context, an entry that serves as the root for all Oracle-specific data. In the simplified DIT shown in [Figure 3-1](#) on page 3-4, both the root Oracle Context and the realm-specific Oracle Context are expanded. The root Oracle Context is the repository for sitewide information—that is, information that applies to all identity management realms and products. Structurally, realm-specific Oracle Contexts are mirror images of the root context, but the information they contain pertains only to a particular realm. These realms store configuration information unique to specific users and other network entities. To learn more about realms, see [Chapter 9, "Enabling Support for Application Service Providers"](#).

In [Figure 3-1](#), the single sign-on container is identified by the entry `cn=SSO`. It contains a single entry, `orclApplicationCommonName=orasso_sso`. This is the

entry for the single sign-on server. In the illustration, the entry has been expanded to show the object classes and attributes that define the entry. For example, the `orclapplicationcommonname` attribute gives the default name of the single sign-on server, `orasso`. Note, too, that the single sign-on server has its own password, which, along with `orclapplicationcommonname`, the directory server uses to authenticate the single sign-on server when the latter performs user searches.

The container `Common` is a repository for information common to all OracleAS products. For instance, it contains attributes that enable products to identify the realm search base, or node, and the realm nickname. Realm-specific `Common` containers—not shown here—contain attributes that enable products to locate users within a realm subtree. In addition to expanding the `SSO` container, the illustration expands entries for an OracleAS user who is also an administrator.

Figure 3-1 Directory Information Tree for OracleAS Single Sign-On



Changing Single Sign-On Server Settings for Directory Access

The `ssooconf.sql` script enables you to change the following settings in the directory:

- directory host name
- directory port
- password for single sign-on server
- SSL connections to the directory

Note: You can change the host name and port number only if the new instance of Oracle Internet Directory is a replicated instance.

To change directory settings for the single sign-on server:

1. Navigate to the script at `ORACLE_HOME/sso/admin/plsql/sso`.
2. Log in to `SQL*Plus` as the schema `orasso`. To obtain the schema password, see [Appendix B](#).

Note: You can run the script only as `orasso`.

3. Run `ssooconf.sql` by issuing the following command:

```
SQL> @ssooconf.sql
```

This prompt appears:

```
Enter value for new_oid_host
```

4. Enter a value for the directory host name; then press **Return** or **Enter**. If, on the other hand, you do not want to change the directory host name, simply press **Return** or **Enter** to move to the next prompt.
5. Repeat step 4 for each of the remaining three prompts, which are `Enter value for new_oid_port`, `Enter value for new_ssoserver_password`, and `Enter value for new_ldapussl`. The last requires that you enter either `Y` (enable) or `N` (disable).

Note: An SSL connection between the single sign-on server and the directory exists by default.

6. To apply the changes, press **Return** or **Enter** one last time.

The script displays updated settings for the single sign-on server along with the old ones.

If you run the script and then decide not to make changes, press **Return** or **Enter** to retain existing values.

Updating the Single Sign-On Server with Directory Changes

The single sign-on server caches metadata about the Oracle Internet Directory DIT. This metadata includes the user search base, user nickname attribute, and

realm-related metadata. In the event that the directory DIT changes, the cache for the single sign-on server must be refreshed. This is effected by running the `ssoreoid.sql` script.

1. Navigate to the script at `ORACLE_HOME/sso/admin/plsql/sso`.

2. Log in to the single sign-on schema:

```
SQL> connect orasso/orasso_password
```

See [Appendix B](#) to obtain the `orasso` schema password.

Note: This script cannot be run as `sys`.

3. Run the script:

```
SQL> @ssoreoid.sql
```

4. Restart the single sign-on server.

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server  
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

These are just a few of the DIT changes that require that the script be run:

- The default realm name or realm DN changes or both change
- A new default realm is created
- The user search base or group search base for the default realm changes or both change
- The user nickname attribute changes

To learn how realm information is changed in Oracle Internet Directory, see *Oracle Internet Directory Administrator's Guide*.

Configuring and Administering Partner Applications

This chapter explains how to enable partner applications for single sign-on. This process involves registering the authentication module `mod_osso` with the single sign-on server. See "[Partner Applications](#)" in Chapter 1 for a definition of partner applications.

The chapter contains the following topics:

- [Registering a Partner Application: What It Means](#)
- [Registering `mod_osso`](#)
- [Deploying Multiple Partner Applications with a Load Balancer](#)
- [Configuring `mod_osso` with Virtual Hosts](#)

Registering a Partner Application: What It Means

Single sign-on partner applications are integrated with `mod_osso`, which is registered automatically by the OracleAS installer. In essence, partner applications are registered *by way* of `mod_osso`. Registering the module creates an entry for it in the identity management infrastructure database as well as on the application computer.

`mod_osso`-integrated applications are registered either by the `ssoreg.sh` script or by the `ssoreg.bat` batch file, depending upon whether the platform is UNIX or Windows. OracleAS Portal, an application enabled for single sign-on not by `mod_osso` but by an SDK, is registered by the `ptlconfig` script. All three tools are invoked by the installer. Only `ssoreg.sh` and `ssoreg.bat` are discussed here. To learn how to use `ptlconfig`, see the appendix devoted to this tool in *Oracle Application Server Portal Configuration Guide*.

Registering `mod_osso`

Under certain circumstances, you must reregister `mod_osso` manually, using the single sign-on registration tool. For instance, the host name and port number of the Oracle HTTP Server on either the infrastructure or the application tier may change after OracleAS is installed. Or SSL may be enabled after installation.

Running the single sign-on registration tool updates the `mod_osso` registration record in `osso.conf`. The tool generates this file whenever it runs.

This section contains the following topics:

- [Syntax and Parameters for `ssoreg`](#)

- [Command Example](#)
- [Restarting the Oracle HTTP Server](#)

Syntax and Parameters for ssoreg

The `ssoreg.sh` and `ssoreg.bat` scripts share the same parameters. The syntax for both commands is provided here. Before running the script, set the `ORACLE_HOME` environment variable to point to the directory where OracleAS is installed. See the section "[Command Example](#)" if you need help. In the example, the home directory is called `gitml`.

Here is the command:

- **UNIX:**

```

$ORACLE_HOME/sso/bin/ssoreg.sh
-oracle_home_path orcl_home_path
-site_name site_name
-config_mod_osso TRUE
-mod_osso_url mod_osso_url
[-virtualhost]
[-update_mode CREATE | DELETE | MODIFY]
[-config_file config_file_path]
[-admin_info admin_info]
[-admin_id adminid]

```
- **Windows:**

```

%ORACLE_HOME%\sso\bin\ssoreg.bat
-oracle_home_path orcl_home_path
-site_name site_name
-config_mod_osso TRUE
-mod_osso_url mod_osso_url
[-virtualhost]
[-update_mode CREATE | DELETE | MODIFY]
[-config_file config_file_path]
[-admin_info admin_info]
[-admin_id adminid]

```

oracle_home_path

Absolute path to the Oracle home.

site_name

Name of the site—typically, the effective host name and port of the partner application. For example, `application.mydomain.com`.

config_mod_osso

If set to `TRUE`, this parameter indicates that the application being registered is `mod_osso`. You must include `config_mod_osso` for `osso.conf` to be generated.

mod_osso_url

The effective URL of the partner application. This is the URL that is used to access the partner application. The value should be specified in this URL format:

```
http://oracle_http_host.domain:port
```

For example:

```
http://application.mydomain.com:7777
```

Omit the port number if the partner Oracle HTTP Server is listening on the default HTTP port of 80 or on the default HTTPS port of 443.

virtualhost

Optional. Include this parameter only if you are registering an Oracle HTTP virtual host with the single sign-on server. Omit the parameter if you are not registering a virtual host.

If you are creating an HTTP virtual host, use the `httpd.conf` file to fill in the following directive for each protected URL:

```
<VirtualHost host_name>
    OsoConfigFile $ORACLE_HOME/Apache/Apache/conf/osso/host_name/osso.conf
    OsoIpCheck off
    #<Location /your_protectedORACLE_HOME_url>
    # AuthType basic
    # Require valid-user
    #</Location>
    #Other configuration information for the virtual host
</VirtualHost>
```

If, on the other hand, you are creating an HTTPS virtual host, use the `ssl.conf` file to fill in the same directive. Note that the commented lines must be uncommented before the application is deployed. Both `httpd.conf` and `ssl.conf` are in `ORACLE_HOME/Apache/Apache/conf`.

After creating a virtual host, run this command to update the Distributed Cluster Management schema:

```
ORACLE_HOME/dcm/bin/dcmctl updateConfig -v -d
```

update_mode

Optional. Creates, deletes, or modifies the partner registration record. `CREATE`, the default, generates a new record. `DELETE` removes the existing record. `MODIFY` deletes the existing record and then creates a new one.

config_file

Location of the `osso.conf` file for the virtual host if one is being configured. It may, for example, be `ORACLE_HOME/Apache/Apache/conf/osso/virtual_host_name/osso.conf`.

This parameter is mandatory if you are registering a virtual host. If you omit `config_file`, the assumption is that you are registering a nonvirtual host. In this case, `ssoreg` creates a file with the name `osso.conf` in `ORACLE_HOME/Apache/Apache/conf/osso`.

admin_info

Optional. User name of the `mod_osso` administrator. If you omit this parameter, the **Administer Information** field on the Edit Partner Application page is left blank.

admin_id

Optional. Any additional information, such as email address, about the administrator. If you omit this parameter, the **Administrator E-mail** field on the Edit Partner Application page is left blank.

Command Example

This command sequence that follows shows a `mod_osso` instance being reregistered with the single sign-on server. In the example, the Oracle home is a directory called `gitml`

- UNIX (csh and tcsh):

```
setenv ORACLE_HOME /private/oracle/gitml

$ORACLE_HOME/sso/bin/ssoreg.sh
-oracle_home_path $ORACLE_HOME
-site_name myhost.mydomain.com
-config_mod_osso TRUE
-mod_osso_url http://myhost.mydomain.com
```

- UNIX (Bourne and ksh)

```
ORACLE_HOME=/private/oracle/gitml; export ORACLE_HOME

$ORACLE_HOME/sso/bin/ssoreg.sh
-oracle_home_path $ORACLE_HOME
-site_name myhost.mydomain.com
-config_mod_osso TRUE
-mod_osso_url http://myhost.mydomain.com
```

- Windows:

```
set ORACLE_HOME=c:\private\oracle\gitml

%ORACLE_HOME%\sso\bin\ssoreg.bat
-oracle_home_path %ORACLE_HOME%
-site_name myhost.mydomain.com
-config_mod_osso TRUE
-mod_osso_url http://myhost.mydomain.com
```

Restarting the Oracle HTTP Server

After running `ssoreg`, restart the Oracle HTTP Server:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

Deploying Multiple Partner Applications with a Load Balancer

You can configure two or more partner application instances in a highly available deployment by placing a load balancer in front of them. The load balancer publishes a single address for partner applications while providing a farm of application servers that actually service requests. The HTTP load balancer can detect when one of the Oracle HTTP Server instances has failed and can then fail over requests to another instance.

The usage scenario presented here takes you through the steps required to configure partner applications with a load balancer.

Usage Scenario

This scenario assumes the following hypothetical configurations:

- There are two partner application computers: `pa1.mydomain.com` and `pa2.mydomain.com`. Both application servers listen on non-SSL port 7777.

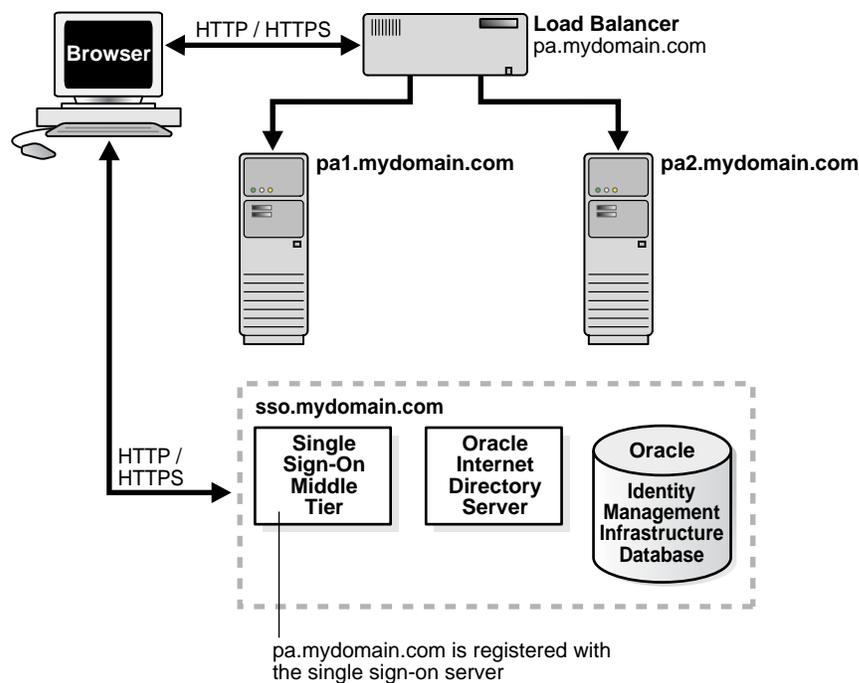
- The partner application computers are configured to use the single sign-on server located at `sso.mydomain.com`.
- The effective host name of the partner application published to users is `pa.mydomain.com`. An HTTP load balancer is configured to listen at this address, on port 80. It load balances and fails over user requests between `pa1.mydomain.com` and `pa2.mydomain.com`.
- The single sign-on server, directory server, and identity management infrastructure database are located at `sso.mydomain.com`.

Note:

- In this scenario, the load balancer is listening on port 80, a non-SSL port number. If the load balancer is configured to use SSL to interact with the browser, a different port number must be selected. The default SSL port number is 443.
 - Two partner application computers are deployed. There can, in fact, be any number of them.
 - The host names presented are examples only. These names may not work in an actual implementation. Substitute values that apply to your installation.
-
-

Figure 4–1 shows what this hypothetical system looks like.

Figure 4–1 Load Balancer with Multiple Partner Applications



Configuration Steps

Setting up the system depicted in Figure 4–1 involves the following tasks:

- [Installing the Partner Applications](#)

- [Configuring the Oracle HTTP Servers on the Partner Application Middle Tiers](#)
- [Configuring the HTTP Load Balancer](#)
- [Reregistering mod_osso on the Partner Application Middle Tiers](#)

Installing the Partner Applications

Install the partner applications on `pa1.mydomain.com` and `pa2.mydomain.com`. When prompted by the installer for a directory location, choose the server located at `sso.mydomain.com`.

Note: The partner application mentioned here can be any Web-based application. In a simple case, it can be an OracleAS core installation that includes the Oracle HTTP Server and OC4J. Consult application-specific installation documentation.

Configuring the Oracle HTTP Servers on the Partner Application Middle Tiers

When a load balancer is placed between the user and the Oracle HTTP servers on the OracleAS middle tier, the effective URL of the partner application changes. The configuration file `httpd.conf` on both middle tiers must be modified to reflect this change. This file is found at `ORACLE_HOME/OracleHTTPServer/conf`.

Complete the following steps:

1. Modify the Oracle HTTP servers at the OracleAS middle tier to listen at the externally published name, which, in the scenario presented, is `pa.mydomain.com`.

Add the following lines to the `httpd.conf` file on `pa1.mydomain.com` and `pa2.mydomain.com`:

```
ServerName pa.mydomain.com
Port 80
```

Note: If multiple ports are listed in `httpd.conf`, the effective port must appear last.

2. If you configure SSL between the browser and the load balancer, and the SSL connection terminates at the load balancer, configure `mod_certheaders` on both `pa1.mydomain.com` and `pa2.mydomain.com`. This module enables the Oracle HTTP Server to treat requests that it receives over HTTP as SSL requests. Add the lines that follow to `httpd.conf`. You can place them at the end of the file. Where they appear is unimportant.

- a. Enter this line to load the module:

```
LoadModule certheaders_module libexec/mod_certheaders.so
```

- b. If you are using OracleAS Web Cache as a load balancer, enter this line:

```
AddCertHeader HTTPS
```

If you are using a hardware load balancer, enter this line:

```
SimulateHttps on
```

Configuring the HTTP Load Balancer

The HTTP load balancer that you use can be either hardware or software. If you elect to use the latter, you can use OracleAS WebCache.

- Hardware Load Balancer

If you use a hardware load balancer, configure one pool of real servers with the addresses `pa1.mydomain.com` and `pa2.mydomain.com`. Configure one virtual server with the address `pa.mydomain.com`. This virtual server is the external interface of the load balancer. For instructions, consult the documentation provided by your load balancer vendor.

- Software Load Balancer

If you use OracleAS Web Cache to load balance connection requests, see the following links:

- "Leveraging Oracle Identity Management Infrastructure" in *Oracle Application Server Web Cache Administrator's Guide*.
- "Routing Single Sign-On Server Requests," also in *Oracle Application Server Web Cache Administrator's Guide*.

Note: For optimal performance, use a hardware load balancer.

Reregistering mod_osso on the Partner Application Middle Tiers

On both partner application instances, reregister `mod_osso` as the partner application `pa.mydomain.com`.

To reregister `mod_osso` on `pa1.mydomain.com`, run `ssoreg`, the registration script. See "Registering `mod_osso`" earlier in the chapter to learn how to run the script. Following the example, the script creates a partner application called `pa.mydomain.com`.

Note: If you are configuring the partner application computers for Distributed Cluster Management, omit the remaining steps. Instead, run this command on `pa1.mydomain.com`:

```
ORACLE_HOME/dcm/bin/dcmctl updateConfig -v -d
```

This command saves your changes to the Distributed Cluster Management repository, which serves as a backup facility for OracleAS configuration files.

To reregister `mod_osso` on `pa2.mydomain.com`:

1. On `pa2.mydomain.com`, log in to the single sign-on administration pages as the single sign-on administrator. Be sure to log in to this URL:

```
http://sso.mydomain.com/pls/orasso
```

2. Use the Administer Partner Applications page to delete the existing entry for the partner application `pa2.mydomain.com`.
3. Copy the `osso.conf` file from `pa1.mydomain.com`. Make sure that you use binary mode if you FTP the file. The default location of the file is `ORACLE_HOME/Apache/Apache/conf/osso`.

4. Synchronize the Distributed Cluster Management repository with the file copy. You do this by running the following command on `pa2.mydomain.com`:

```
ORACLE_HOME/Apache/Apache/bin/ssotransfer ORACLE_
HOME/Apache/Apache/conf/osso/osso.conf
```

Note: The `ssotransfer` command should not be used to synchronize the Distributed Cluster Management repository with the `mod_osso` configuration file created for a virtual host. To learn how to register `mod_osso` for a virtual host, see "[Configuring mod_osso with Virtual Hosts](#)".

5. Restart the Oracle HTTP Server:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

6. Test the partner application, using the effective URL:

```
http://pa.mydomain.com
```

For more information about integrating partner applications with `mod_osso`, see the chapter about developing applications for single sign-on in *Oracle Identity Management Application Developer's Guide*.

Configuring mod_osso with Virtual Hosts

Some deployments may require more than one Web site to be deployed on a single Oracle HTTP Server. An application may, for example, have to be available both over HTTP and HTTPS. In the scenario that follows, an SSL virtual host is configured to be protected by `mod_osso`. Although the virtual host is an SSL host, the scenario applies to any virtual host.

The scenario assumes the following conditions:

- The host name of the application middle tier is `app.mydomain.com`.
- The middle tier is already configured as a non-SSL partner application. This is typically done by the OracleAS Installer when the application is first installed.
- The default SSL port number of the application middle tier is 4443.

To configure `app.mydomain.com` as an SSL virtual host:

1. Make sure that Oracle Identity Management components are up and running—especially Oracle Internet Directory and the single sign-on server.
2. Check that `app.mydomain.com` has been defined as an SSL virtual host. The OracleAS installer does this in the `Virtual Host Context` section of the `ssl.conf` file. The file is in `ORACLE_HOME/Apache/Apache/conf`.
3. Create a partner application for the SSL site:
 - a. Make sure that the Oracle home of the middle tier is set with the correct path. If you need help, see the command examples in the section "[Registering mod_osso](#)".
 - b. Run this command on the middle tier:

- UNIX:

```
ORACLE_HOME/sso/bin/ssoreg.sh
```

```
-oracle_home_path $ORACLE_HOME
-site_name app.mydomain.com
-config_mod_osso TRUE
-mod_osso_url https://app.mydomain.com:4443
-virtualhost
-config_file $ORACLE_HOME/Apache/Apache/conf/osso/osso-https.conf
```

– Windows:

```
%ORACLE_HOME%\sso\bin\ssoreg.bat
-oracle_home_path %ORACLE_HOME%
-site_name app.mydomain.com
-config_mod_osso TRUE
-mod_osso_url https://app.mydomain.com:4443
-virtualhost
-config_file $ORACLE_HOME/Apache/Apache/conf/osso/osso-https.conf
```

4. Go to the `mod_osso.conf` file at `ORACLE_HOME/Apache/Apache/conf`. Once there, comment this directive:

- UNIX:

```
LoadModule osso_module libexec/mod_osso.so
```

- Windows:

```
LoadModule osso_module modules\ApacheModuleOsso.dll
```

5. In `httpd.conf`, found in the `conf` directory, add the directive that you just commented in the preceding step. In a default setup, place the directive right after `LoadModule wchandshake_module libexec/mod_wchandshake.so`.
6. In `ssl.conf`, which is also in the `conf` directory, update `VirtualHost` to include the `osso.conf` file for the virtual host. Name the file `osso-https.conf` to avoid conflict with the default `osso.conf` file. Check that the file name matches the name used in the registration script.

```
<VirtualHost _default_:4443>
.
.
.
OssConfigFile ORACLE_HOME/Apache/Apache/conf/osso/osso-https.conf
OssIpCheck off
<Location /your_protected_url_for_the_virtual_site>
  AuthType basic
  Require valid-user
</Location>
.
.
.
</VirtualHost>
```

7. Update the Distributed Cluster Management Repository:

```
ORACLE_HOME/dcm/bin/dcmctl updateConfig -v -d
```

8. Restart the Oracle HTTP Server on the application middle tier:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

9. Test both the SSL and the non-SSL site.

Configuring and Administering External Applications

This chapter describes how to configure external applications for single sign-on support. These are Web applications that are not modified to delegate authentication to the single sign-on server. Configuring a Web application as an external application enables it to be single sign-on enabled without having to change its interface. See "External Applications" in Chapter 1 for more about these applications.

The chapter contains the following topics:

- [Using the Interface to Deploy and Manage External Applications](#)
- [Proxy Authentication for Basic Authentication Applications](#)

Using the Interface to Deploy and Manage External Applications

The Administer External Applications page, accessible as a link on the SSO Server Administration page, is used to add, edit, or delete external applications. Once you add these applications, users can access them in the External Applications portlet of OracleAS Portal.

This section covers the following topics:

- [Adding an External Application](#)
- [Editing an External Application](#)
- [Storing External Application Credentials in the Single Sign-On Database](#)

Adding an External Application

Clicking the **Add External Application link** takes you to the Create External Application page. This page contains the following headings and fields:

Table 5–1 External Application Login

Field	Description
Application Name	Enter a name that identifies the external application. This is the default name for the external application.
Login URL	Enter the URL to which the HTML login page for the external application is submitted for authentication. This, for example, is the login URL for Yahoo! Mail: <code>http://login.yahoo.com/config/login?6p4f5s403j3h0</code>
Username/ID Field Name	Enter the term that identifies the user name or user ID field of the HTML login form for the application. You find this term by viewing the HTML source of the form. (See the example after the steps immediately following). This field is not applicable if you are using basic authentication.
Password Field Name	Enter the term that identifies the password field of the HTML login form for the application. You find this term by viewing the HTML source of the form. (See the example after the steps immediately following). This field is not applicable if you are using basic authentication.

Table 5–2 Authentication Method

Field	Description
Type of Authentication Use	<p>Use the pulldown menu to select the form submission method for the application. This method specifies how message data is sent by the browser. You find this term by viewing the HTML source for the login form. Select one of the following three methods:</p> <p>POST: Posts data to the single sign-on server and submits login credentials within the body of the form.</p> <p>GET: Presents a page request to a server, submitting the login credentials as part of the login URL.</p> <p>Basic authentication: Submits the login credentials in the application URL, which is protected by HTTP basic authentication.</p> <p>Notes:</p> <ul style="list-style-type: none"> ■ Basic authentication uses pop-up windows, which by default are blocked by Windows XP, service pack 2. If you use this service pack, make sure that you reconfigure browser settings to display the window for the single sign-on login page. Use the pop-up blocker item in the Tools menu of Internet Explorer. Other browsers and browser plugins are able to block popups. Mozilla is one of these. Make sure that these do not block the single sign-on login page. ■ If you use Internet Explorer 5.0 or a later version, basic authentication may not work with external applications. This version of Internet Explorer includes Microsoft MS04-004 Cumulative Security Update (832894). See this link for a workaround: <code>http://support.microsoft.com</code>

Table 5–3 Additional Fields

Field	Description
Field Name	Enter the name of any additional fields on the HTML login form that may require user input to log in. This field is not applicable if you are using basic authentication.
Field Value	Enter a default value for a corresponding field name value, if applicable. This field is not applicable if you are using basic authentication.

Use the following steps to add an external application:

1. From the Administer External Applications page, select **Add External Application**.

The Create External Application page appears.

2. In the **External Application Login** field, enter the name of the external application and the URL to which the HTML login form is submitted. If you are using basic authentication, enter the protected URL.
3. If the application uses HTTP POST or HTTP GET authentication, in the **User Name/ID Field Name** field, enter the term that identifies the user name or user ID field of the HTML login form. You can find the name by viewing the HTML source of the login form.

If the application uses the basic authentication method, the **User Name/ID Field Name** field should be empty.

4. If the application uses HTTP POST or HTTP GET authentication, in the **Password Field Name** field, enter the term that identifies the password field of the application. See the HTML source of the login form.

If the application uses the basic authentication method, the **Password Field Name** field should be empty.

5. In the **Additional Fields** field, enter the name and default values for any additional fields on the HTML login form that may require user input.

If the application uses the basic authentication method, these fields should be empty.

6. Select the **Display to User** check box to allow the default value of an additional field to be changed by the user on the HTML login form.
7. Click **OK**. The new external application appears under the **Edit/Delete External Application** heading on the Administer External Applications page, along with the other external applications.

8. Click the application link to test the login.

The following example shows the source of the values that are used for Yahoo! Mail.

```
<form method=post action="http://login.yahoo.com/config/login?6p4f5s403j3h0"
autocomplete=off name=a>
...
<td><input name=login size=20 maxlength=32></td>
....
<td><input name=passwd type=password size=20 maxlength=32></td>
...
<input type=checkbox name=".persistent" value="Y" >Remember my ID & password
...
</form>
```

The source provides values for the following:

- Login URL :
http://login.yahoo.com/config/login?6p4f5s403j3h0
- Username/ID Field Name: login
- Password Field Name: passwd
- Type of Authentication Used: POST
- Field Name: .persistent Y

- Field Value: [off]

Editing an External Application

Clicking the pencil icon next to an application takes you to the Edit External Application page, where you can edit the values that you entered when you added the application. When you are finished editing, click **Apply** to enter the changes and to redisplay the page with the updated values.

Storing External Application Credentials in the Single Sign-On Database

Each external application expects to receive a user name and password each time the user logs in to the application. To enable single sign-on to these applications, users are given the option of storing their credentials in the single sign-on database when they log in.

If single sign-on users are logging in to an external application for the first time, they are presented with the External Application Login page. After entering credentials, they can select the check box **Remember My Login Information for This Application**. If they choose this option, the next time they access the application, the single sign-on server logs in on their behalf.

Figure 5–1 reproduces the External Application Login page.

Figure 5–1 External Application Login Page

External Application Login

Enter your user name (or other form of application identification) and password for this application. You may also enter custom values for any additional login parameters shown. The SSO Server uses this information to login on your behalf. If you click Remember My Login Information For This Application, you will then be logged in automatically each time you access this application.

Application Name: www.yahoo.com

User Name/ID

Password

Remember My Login Information For This Application

Login Close

Copyright © 2004, Oracle. All Rights Reserved

Note:

- If you change your password, you must also update the password on the External Application Login page. If you neglect to do so, this page returns an error message when you try to log in.
 - Your password cannot contain the following characters: &, {, }, <, >, ", ', (, and).
-
-

Proxy Authentication for Basic Authentication Applications

The standard way to access external applications enabled by single sign-on is through the External Applications portlet of OracleAS Portal, an SDK-enabled partner

application. Applications accessed in this way can be configured for GET, POST, or basic authentication.

An alternative method is to use the Oracle HTTP Server as a secure proxy for applications that reside on a separate Web server. This method involves configuring the modules `mod_osso` and `mod_proxy` to support single-sign-on-enabled basic authentication. The advantage of the proxy approach is that it eliminates the brief screen flicker that occurs when external applications are accessed in the standard way.

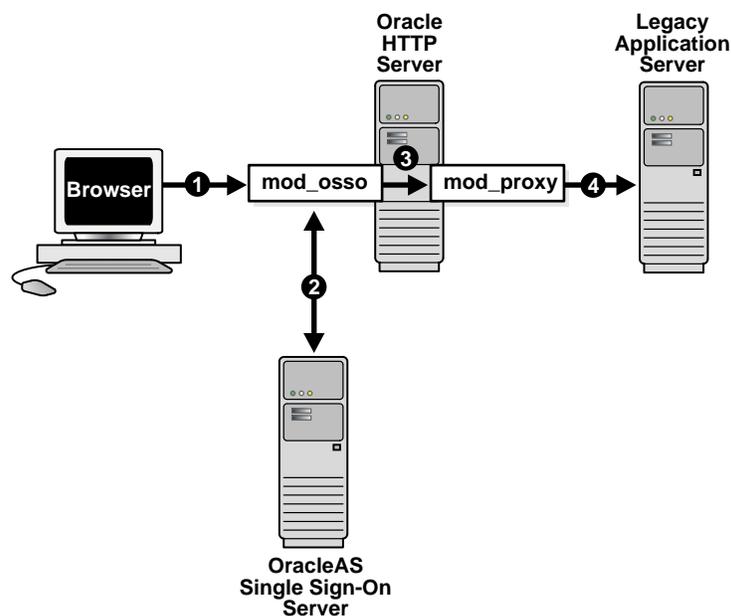
This section contains the following topics:

- [Configuring the Oracle HTTP Server as a Proxy for Basic Authentication](#)
- [Configuration Requirements](#)
- [Configuration Steps](#)

Configuring the Oracle HTTP Server as a Proxy for Basic Authentication

Configured correctly, authentication to `mod_osso`-enabled external applications is similar to what it is for partner applications: `mod_osso` intercepts a URL request and redirects it to the single sign-on server. [Figure 5–2](#) illustrates the process.

Figure 5–2 Authentication Flow Using `mod_osso/mod_proxy`



1. The single sign-on user requests an external application by selecting a bookmark or by entering a virtual URL. This URL enables the Oracle HTTP Server to intercept the request.
2. `mod_osso` adds an authentication header to the intercepted request and retrieves the user's credentials from the single sign-on server.
3. `mod_osso` sets the header value with the user's credentials, retrieved from the single sign-on server. `mod_osso` then passes this header to `mod_proxy`.
4. `mod_proxy` passes the user's credentials—in the form of a basic authentication header—to the real URL. `mod_proxy` does this by using directives that map the virtual URL to the real URL.

Configuration Requirements

The following criteria must be met before the Oracle HTTP Server can be configured for basic authentication to legacy applications:

- The application to be proxied must be registered as a basic authentication application with the single sign-on server. See ["Adding an External Application"](#) for instructions.
- The Oracle HTTP Server must have `mod_osso` installed and enabled.
- The Oracle HTTP Server must have the default `mod_proxy` installed and enabled.
- If the Web server that hosts the external application uses the Oracle HTTP Server as a proxy, the Web server must not have `mod_osso` enabled.

Configuration Steps

To configure the Oracle HTTP Server for basic authentication to external applications, complete these steps:

1. Add the section that follows to `mod_osso.conf` on the application tier. The file is at `ORACLE_HOME/Apache/Apache/conf`.

```
<IfModule mod_proxy.c>
<Location /application_virtual_path>
    require valid-user
    AuthType Basic
    OsoLegacyApp on | off
</Location>

ProxyPass /application_virtual_path/ http://host:port/application_real_path/
ProxyPassReverse /application_virtual_path/ http://host:port/application_real_path/
</IfModule>
```

The `OsoLegacyApp` directive indicates whether the protected URL is a legacy application. If the directive is missing or is set to `off`, the code that retrieves the application user name and password from the single sign-on database is not executed. The two `mod_proxy` directives `ProxyPass` and `ProxyPassReverse` map the virtual URL to the real URL.

2. Add this line to `httpd.conf`:

```
Listen 5000
```

This parameter instructs `mod_osso` to use the non-SSL port 5000 to access information about external applications.

3. Restart the Oracle HTTP Server:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

4. Update the Distributed Cluster Management schema:

```
ORACLE_HOME/dcm/bin/dcmctl updateConfig -v -d
```

Note:

- The directory where the virtual URL resides need not be specified. For convenience, this URL may consist of only the application name.
 - If SSL is enabled, substitute `https` for `http` in the real URL of the application.
-
-

Multilevel Authentication

This document explains how to configure a single sign-on system that assigns different authentication levels to different partner applications. Such a system enables the administrator to tailor authentication behavior to the security level of the application requested.

The document contains the following topics:

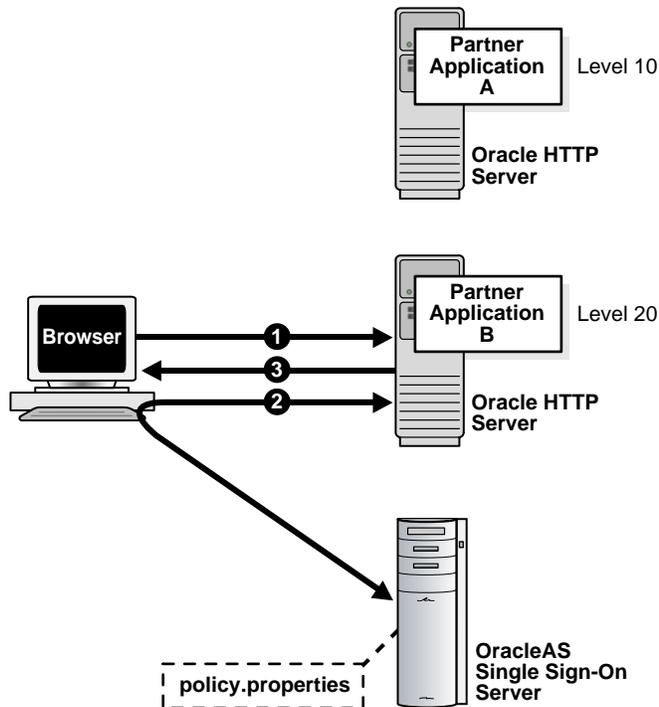
- [What Is Multilevel Authentication?](#)
- [How Multilevel Authentication Works](#)
- [Components of a Multilevel System](#)
- [Configuring Multilevel Authentication](#)

What Is Multilevel Authentication?

OracleAS Single Sign-On enables you to assign different authentication levels to the applications that it protects. You can then map these authentication levels to specific authentication plugins. You may, for example, configure a highly sensitive application to require a user certificate and a less sensitive application to require a user name and password.

How Multilevel Authentication Works

[Figure 6-1](#) on page 6-2 illustrates how multilevel authentication works.

Figure 6–1 Multilevel Authentication Flow

1. The user has already authenticated to Application A. He or she now goes to Application B.
2. Application B redirects the user to the single sign-on server.
3. Because Application B has a higher authentication level than Application A, the single sign-on server forces the user to authenticate again, this time with a higher credential.

Note: In release 10.1.2, authentication is at the root level of a partner application. You cannot assign authentication levels to URLs under the root.

Components of a Multilevel System

The following topics are key to understanding how multilevel authentication works:

- [Authentication Levels](#)
- [Authentication Plugins](#)

Authentication Levels

Authentication levels are parameters that enable you to specify a particular authentication behavior for an application. You use the `policy.properties` file to configure the authentication level names and values that make up these parameters. This file is at `ORACLE_HOME/sso/conf`. A copy of it appears in [Appendix C](#).

[Table 6–1](#) provides examples of authentication levels. You can customize these to suit your deployment requirements and can provide additional ones.

Table 6–1 Default Authentication Levels

Authentication Level Names	Authentication Level Values
LowSecurity	20
LowMediumSecurity	30
MediumSecurity	40
MediumHighSecurity	50
HighSecurity	60

The authentication level names must be unique. For example, a system that includes both `NoSecurity=10` and `NoSecurity=20` is unacceptable. The lower the numeric value of a level, the lower the level of security. Values must, however, be positive integers.

Users who log in at a high level such as `MediumHighSecurity` and then attempt to access a lower-level application are not rechallenged for credentials. Conversely, users who log in at a low-level application such as `LowMediumSecurity` and then attempt to access a higher-level one are challenged with the required level.

Authentication Plugins

An authentication plugin is an implementation of a specific authentication method. This method collects credentials from users and authenticates them.

You can pair one of the authentication levels introduced in the preceding section with one of the authentication methods described in the bulleted list that follows. The authentication level that an authentication plugin maps to is deployment specific. You use `policy.properties` to achieve the pairing.

- Password authentication
 - This is the default, standard method.
- Digital certificates
 - See [Chapter 7](#) for a discussion of certificate authentication.
- Windows native authentication
 - See the chapter about integrating with Microsoft Active Directory in *Oracle Identity Management Integration Guide*.

Configuring Multilevel Authentication

Applications not configured for a specific authentication level default to password authentication and are assigned an authentication level of `MediumSecurity`. If you require a different authentication level, you must modify `policy.properties`. Use the configuration scenario that follows for guidance.

Usage Scenario

This usage scenario explains how two hypothetical partner applications are configured to use different authentication levels and plugins. It assumes these conditions:

- Application `pa1` is deployed on host `pa1.mydomain.com`. It listens on port `7777`.
- `pa1` is already registered with the single sign-on server.

- pa1 must use certificate authentication.
- Application pa2 is deployed on host pa2.mydomain.com. It listens on port 7777.
- pa2 is already registered with the single sign-on server.
- pa2 must use password authentication.

Configuration Steps

Modify `policy.properties` with the following configurations.

1. Choose the name of the authentication level from `policy.properties`. If necessary, add a new authentication level and corresponding name to the file.
2. Assign authentication levels to the root URLs of the two partner applications:

```
pa1.mydomain.com\:7777 = HighSecurity
pa2.mydomain.com\:7777 = MediumSecurity
```

Note: Be sure to include the backslash after the domain name.

3. Assign authentication plugins to the authentication level names that you assigned in step 1:

```
HighSecurity_AuthPlugin = oracle.security.sso.server.auth.SSOX509CertAuth
MediumSecurity_AuthPlugin = oracle.security.sso.server.auth.SSOServerAuth
```

Note that the authentication plugin name is a combination of the authentication level name that you assigned in step 1 and the suffix `_AuthPlugin`.

4. Save `policy.properties`; then restart the single sign-on middle-tier.

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

5. Test the partner applications.

Signing On with Digital Certificates

Single sign-on with X.509 client certificates provides a stronger degree of security than simple authentication. It offers the benefit that partner applications are, by default, PKI enabled when the single sign-on server is PKI enabled.

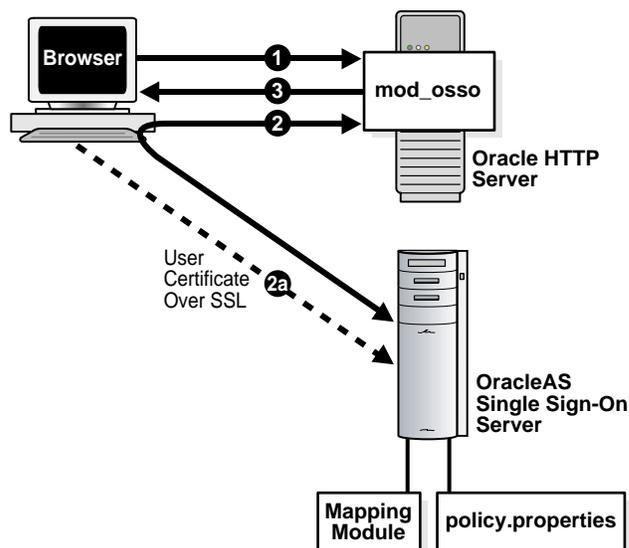
This chapter contains the following topics:

- [How Certificate-Enabled Authentication Works](#)
- [System Requirements](#)
- [Configuring the Single Sign-On System for Certificates](#)
- [Maintaining a Certificate Revocation List](#)

How Certificate-Enabled Authentication Works

Figure 7-1 depicts the authentication flow for certificate-enabled sign-on.

Figure 7-1 Certificate-Enabled Single Sign-On



1. The user tries to access a partner application.
2. The partner application redirects the user to the single sign-on server for authentication. As part of this redirection, the browser sends the user's certificate over SSL to the OracleAS Single Sign-On Server.

to the login URL of the server (2a). If it is able to verify the certificate, the server returns the user to the requested application.

3. The application delivers content.

Note: Users whose browsers are configured to prompt for a certificate-store password may only have to present this password once, depending upon how their browser is configured. If they log out and then attempt to access a partner application, the browser passes their certificate to the single sign-on server automatically. This means that they never really log out. To effectively log out, they must close the browser.

System Requirements

The following criteria must be met before certificate-enabled single sign-on can proceed:

- The single sign-on server and Oracle Internet Directory must be installed.
- The Oracle HTTP Server must have a valid server certificate installed.
- The client certificate DN must be chosen in such a way that it meets one of the following two criteria:
 - The DN of the user certificate is the same as the user DN in Oracle Internet Directory
 - The DN of the user certificate contains the user nickname and, optionally, the name of the realm that the user belongs to
- The certificate of the client certificate issuer must be installed as a trusted certificate on the single sign-on server.
- The certificate of the server certificate issuer must be installed as a trusted certificate in the user's browser.

Configuring the Single Sign-On System for Certificates

Certificate-enabled single sign-on is not a default option in OracleAS, and it must be configured after installation. Before configuring certificate authentication, you must enable the single sign-on system for SSL. Perform the tasks in ["Enabling SSL" in Chapter 8](#); then return to this section and configure the following components for certificates:

- [Oracle HTTP Server](#)
- [Single Sign-On Server](#)
- [Oracle Internet Directory](#)

Oracle HTTP Server

Configuring the Oracle HTTP Server for certificates consists of adding parameters to the `ssl.conf` file and, optionally, choosing the certificate authority that issues server and user certificates.

Setting SSL Parameters

To set the required SSL parameters, complete the following steps:

1. Go to `ssl.conf`. The file is at `ORACLE_HOME/Apache/Apache/conf`.
2. In the `SSL Virtual Host Context` section of `ssl.conf`, add or edit the parameters listed in [Table 7-1](#). At the same time, verify that the `SSL Engine` parameter has been set to `on`. This should have been done as part of configuring the Oracle HTTP Server for SSL.

Table 7-1 HTTP Parameters for Certificate-Enabled Single Sign-On

Parameter	Description
<code>SSLWallet</code>	<p>The location, or path, of the server wallet. The default location is <code>ORACLE_HOME/Apache/Apache/conf/ssl.wlt/default</code>.</p> <p>Note:</p> <p>The actual location of the Oracle home must be substituted for the variable.</p> <p>If OracleAS Certificate Authority is installed in the same Oracle home as OracleAS Single Sign-On, and you want to use this CA to issue certificates, the wallet location is <code>ORACLE_HOME/oca/wallet/ssl</code>. See "Choosing a Certificate Authority" for details.</p>
<code>SSLWalletPassword</code>	Password for the server wallet
<code>SSLVerifyClient</code>	<p>The verification type for client certificates. These are the three types:</p> <ul style="list-style-type: none"> ■ <code>none</code>—SSL without certificates ■ <code>optional</code>—server certificate and optionally client certificate ■ <code>require</code>—server and client certificates <p>You must choose either <code>optional</code> or <code>require</code>.</p>

Choosing a Certificate Authority

If you have OracleAS Certificate Authority installed and want to use this CA to issue certificates, edit `ssl.conf` to point to the desired Oracle CA wallet. You can either use the Oracle CA wallet described in [Table 7-1](#) or have the Oracle CA issue a wallet that is specifically for the single sign-on server. If you choose the first option, copy the wallets that are in the wallet directory for the Oracle CA to the default wallet directory. If you choose the second option, see Chapter 7 in *Oracle Application Server Certificate Authority Administrator's Guide* for instructions. The relevant section is "Server/SubCA Certificates Tab." This is a subsection of "End-User Tabs and Processes." Once you obtain the wallet, edit `ssl.conf` to point to the wallet's location.

You may, of course, elect to use a third-party CA. In this case, too, you must edit `ssl.conf` to point to the wallet's location as explained in [Table 7-1](#).

Using OracleAS Single Sign-On in conjunction with OracleAS Certificate Authority simplifies the certificate provisioning process. You can configure the Oracle CA to broadcast the URL for its UI to single sign-on users. Users can then use this link to request a single sign-on certificate that is automatically linked to their entry in Oracle Internet Directory.

Single Sign-On Server

Configuring the single sign-on server to accept certificates consists of these tasks:

- [Configure policy.properties with the Default Authentication Plugin](#)
- [Modify the Configuration File for the Authentication Plugin \(Optional\)](#)

- [Customize the User Name Mapping Module \(Optional\)](#)
- [Restart the Single Sign-On Middle Tier](#)

Perform at least the first and the last step. Add the middle two if you want to customize the user name mapping module. The default module for user name mapping matches the distinguished name (DN) in the client certificate with a single sign-on user in Oracle Internet Directory. The default implementation assumes that the user's DN in the directory is the same as the certificate DN. A module that maps a field in the certificate DN to the user's name in Oracle Internet Directory is also available. If you want to substitute this module for the DN mapping module, modify the `CertificateMappingModule` parameter as prescribed in the third task.

Configure `policy.properties` with the Default Authentication Plugin

Update the `DefaultAuthLevel` section of the `policy.properties` file with the correct authentication level for certificate sign-on. This file is at `ORACLE_HOME/sso/conf`. Set the default authentication level to this value:

```
DefaultAuthLevel = MediumHighSecurity
```

Then, in the `Authentication plugins` section, pair this authentication level with the default authentication plugin:

```
MediumHighSecurity_AuthPlugin = oracle.security.sso.server.auth.SSOX509CertAuth
```

For your convenience, `policy.properties` is available in [Appendix C, "policy.properties"](#).

Modify the Configuration File for the Authentication Plugin (Optional)

The `X509CertAuth.properties` file contains the parameters that follow. The file is in the same directory as `policy.properties`.

Note: Omit this step if you are using the DN-based mapping module.

CertificateMappingModule This parameter is set to the class file that performs user name mapping. The parameter can have one of two default values:

```
oracle.security.sso.server.auth.SSOCertMapperDn
```

or

```
oracle.security.sso.server.auth.SSOCertMapperNickname
```

The first module assumes that the user's DN in the directory is the same as the certificate DN. This is the default, out-of-the-box setting. The second module assumes that the first attribute in the user DN in the certificate is `cn`. It also assumes that this attribute is the same as the user nickname in the default realm of Oracle Internet Directory. If, for example, the user DN in the certificate is `cn=john, cn=users, dc=acme, dc=com`, you can use the second module. If, on the other hand, the DN is `e=john.smith@acme.com, cn=john, cn=users, dc=acme, dc=com`, you cannot use the module. You can, however, write a mapping module that uses this DN. See ["Customize the User Name Mapping Module \(Optional\)"](#) for details. If you decide to write your own module, set `CertificateMappingModule` to the class file name for your implementation.

CheckUserCertificate This parameter indicates whether the user certificate must be verified in Oracle Internet Directory. The default value is `true`. If you deem the SSL protection provided by the Oracle HTTP Server to be sufficient, set this parameter to `false`.

CertificateAuthFailureUrl If certificate authentication fails, the user is redirected to this URL, which displays an error message.

CertificateAuthFallback Set this parameter to `true` if you want to make password authentication available to a user who tries to log in without a valid certificate. This fallback does not occur by default. You must enable it. If the parameter is set to `false` or is absent entirely, users see a message that tells them they must present a valid certificate. You may have to add `CertificateAuthFallback` to the file. Place it at the end this way:

```
#Allow authentication fallback
CertificateAuthFallback=true
```

Note: If `CertificateAuthFallback` is set to `true`, you cannot use multilevel authentication.

Customize the User Name Mapping Module (Optional)

To customize the user name mapping module, implement a mapping module based on `oracle.security.sso.ias904.toolkit.IPASUserMappingInterface`. Refer to the example mapping modules shipped with this release. Again, these modules are `SSOCertMapperDN.java` and `SSOCertMapperNickname.java`.

Note: Omit this step if you are not writing your own mapping module.

The example modules contain the following classes:

- Mapping module Interface

This interface contains the following methods:

```
public IPASUserInfo getUserInfo(
    javax.servlet.http.HttpServletRequest request)
    throws IPASException;
```

- User information class

This class contains user information such as the user nickname and user DN. The package name is `oracle.security.sso.ias904.toolkit.IPASUserInfo`. The constructor looks like this:

```
Public IPASUserInfo(
    String userNickName
    String realmNickname)

Public IPASUserInfo(
    String userNickName,
    String userDN,
    String userGUID,
    String realmNickname,
    String realmDN,
    String realmGUID)
```

- Exception class

A problem with user name mapping raises this exception. The class name is `oracle.security.sso.ias904.toolkit.IPASException`. The super class is `java.lang.Exception`. The constructor looks like this:

```
public IPASException()  
public IPASException(String Message)
```

1. Extract `ipassample.jar`, the file that contains the modules:

```
ORACLE_HOME/jdk/bin/jar -xvf ORACLE_HOME/sso/lib/ipassample.jar
```

2. Create a Java class that implements the following interface:

```
oracle.security.sso.ias904.toolkit.IPASUserMappingInterface
```

3. Compile your custom implementation:

```
ORACLE_HOME/jdk/bin/javac -classpath ORACLE_HOME/sso/lib/  
ipastoolkit.jar:ORACLE_HOME/lib/servlet.jar -d class_directory  
java_file_name
```

4. Jar your class file and place it into `ORACLE_HOME/sso/plugin`:

```
ORACLE_HOME/jdk/bin/jar -cvf ORACLE_HOME/sso/plugin/CertMapImpl.jar class_  
directory
```

This step assumes that you do not have individual class files in the plugin directory. If the directory contains the individual files, they may be duplicated.

5. Update `x509CertAuth.properties` with your implementation. See ["Modify the Configuration File for the Authentication Plugin \(Optional\)"](#).

Restart the Single Sign-On Middle Tier

After configuring the server, restart the middle tier:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server  
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Oracle Internet Directory

For certificate-based authentication to be successful, the user certificate must be present in Oracle Internet Directory. If the certificate is issued by OracleAS Certificate Authority, the certificate is published in the directory automatically. This may also be true if the CA is in-house. If the certificate issuer is a third-party CA, a self-service application can fulfill this function. Or the directory administrator can try to add the certificate to the directory as an LDIF file, using the command-line tool `ldapmodify`.

Note:

- The procedures in this section assume that the value of `CheckUserCertificate` is set to `true` in the `X509CertAuth.properties` file. See ["Modify the Configuration File for the Authentication Plugin \(Optional\)"](#).
 - You can search the directory for `usercertificate`, the binary attribute that stores certificates. See the appendix about certificate searching in *Oracle Internet Directory Administrator's Guide*.
-
-

If you use `ldapmodify` to publish the certificate, set the appropriate globalization support variable for your environment before running the tool. Here is an example:

- UNIX:


```
setenv NLS_LANG AMERICAN_AMERICA.UTF8
```
- Windows:


```
set NLS_LANG=AMERICAN_AMERICA.UTF8
```

In UNIX, you may have to use a different procedure to set this variable if you are using a shell other than `cs`h or `tc`sh.

Here is the syntax of `ldapmodify`:

```
ORACLE_HOME/bin/ldapmodify
-h directory_host
-p directory_ssl_port
-D "directory_administrator"
-w administrator_password
-f file_name.ldif
```

In the example LDIF file that follows, the certificate of user `jsmith` is represented as an attribute of his entry in the directory. The attribute type is `usercertificate`. The attribute value is the long string that follows the attribute type.

```
dn: cn=jsmith,cn=users,dc=realml,dc=oracle,dc=com
changetype: modify
replace: usercertificate
usercertificate: :MIIC3TCCAkYCAgP3MA0GCSqGSIb3DQEBAUAMIG8MQswCQ
NYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcml5pYTEXMBUGA1UEBxMOUmVkd29vZCBTaG9yZXMxGzAZBg
VBAoTEk9yYWNsZSBDb3Jwb3JhdG1vbJEFMB0GA1UECxMwV2ViIFNpbmVudmVudmVudmVudmVudmVudmVudm
1UEAxMwV2VydG1maWNhY0EhMmF4gontc4mxSKh/zAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAkwXoCLDRqm
KY9LQtIjLnCaIJKUZmS1Qj+bhu/IHeZLGHg4TJg3O2XVA5u/VxwjLeGBqLXy2z7o3Ru jNKx2CVx6p/0Hk
jnw4w6KVau2hcBgC9m4kzUGhHJ9b65v/zx7dIUkyJr4RF+1JhJg4/oYXxLrYHp5NAkHP4htT0gqCXiI=
```

Because it is a non-ASCII value, the certificate must be encoded in base 64 format, as shown here. Unlike other attributes, a base 64 attribute requires a double colon (`::`) as a delimiter. Note, too, that the use of a tab enables a base 64 attribute to be folded.

Maintaining a Certificate Revocation List

To ensure that users are unable to log in using invalid or expired certificates, the administrator must maintain an up-to-date certificate revocation list (CRL) on the Oracle HTTP Server. The CA that issued the certificate must provide this list. The `ca-bundle.crl` file can be used to maintain it. The path to the CRL file must be `ORACLE_HOME/Oracle/HTTPServer/conf`.

OracleAS users who use digital certificates to authenticate must not be able to update the `userCertificate` attribute in their directory entry. The reason is the potentially long lapse time between the revocation of a certificate and the update of the CRL. Once invalid users pass a CRL check, the only bar to login is the `userCertificate` setting. Fortunately, Oracle Internet Directory, by default, denies users access to `userCertificate`. The attribute must be modified only by trusted entities such as the single sign-on administrator, OracleAS Certificate Authority, or a third-party certificate authority.

For details about implementing and maintaining a CRL, see comments in the `SSL Virtual Host Context` section of `ssl.conf`.

Advanced Deployment Options

This chapter explores nondefault ways to use OracleAS Single Sign-On. It presents scenarios that you may encounter in a production environment. Some of these scenarios involve deploying and configuring the component to interact with other OracleAS components.

The chapter contains the following topics:

- [Enabling SSL](#)
- [Deployment Scenarios](#)
- [Replicating the Identity Management Database](#)
- [Deploying OracleAS Single Sign-On with a Proxy Server](#)
- [Setting Up Directory Synchronization for User Nickname Changes](#)

Enabling SSL

This section explains how to enable the single sign-on server for Secure Sockets Layer (SSL). In SSL, a secret session key is created, enabling the exchange of information over a secure channel. When the user logs in, the Web server sends the browser a digital certificate. The browser uses a public key sent by the Web server to encrypt a random number. This encrypted data is used in turn to create the secret key. Enabling the single sign-on server for SSL confers this form of protection on the server's partner applications. The process provides OracleAS with a high degree of security.

Out of the box, the single sign-on server uses the HTTP port of the Oracle HTTP Server. You can, however, configure SSL after installation. Complete the following tasks in the order listed:

- [Enable SSL on the Single Sign-On Middle Tier](#)
- [Reconfigure the Identity Management Infrastructure Database](#)
- [Protect Single Sign-On URLs](#)
- [Restart the Oracle HTTP Server and the Single Sign-On Middle Tier](#)
- [Reregister Partner Applications](#)

Note: If the Oracle HTTP Server is configured for SSL (topic 1) you must configure the single sign-on server for SSL as well (remaining topics); otherwise users will be unable to access single sign-on URLs. To skirt this restriction, disable SSL directives for URLs that you want to access over HTTP. You do this by editing `ORACLE_HOME/sso/conf/sso_apache.conf`.

Enable SSL on the Single Sign-On Middle Tier

The following steps involve configuring the Oracle HTTP Server. Perform them on the single sign-on middle tier. In doing so, keep the following in mind:

- You must configure SSL on the computer where the single sign-on middle tier is running.
- You are configuring one-way SSL.
- You may enable SSL for simple network encryption; PKI authentication is not required. Note though that you must use a valid wallet and server certificate. The default wallet location is `ORACLE_HOME/Oracle/HttpServer/conf/ssl.wlt/default`. If you want to use a different wallet, see the guidelines in the section "Oracle HTTP Server" in [Chapter 7](#). The chapter about managing wallets and certificates in *Oracle Application Server Administrator's Guide* is also helpful.

To quickly enable SSL on the Oracle HTTP Server, do the following:

1. Back up the `opmn.xml` file, found at `ORACLE_HOME/opmn/conf`.
2. In `opmn.xml`, change the value for the `start-mode` parameter to `ssl-enabled`. This parameter appears in boldface in the xml tag immediately following.

```
<ias-component id="HTTP_Server">
  <process-type id="HTTP_Server" module-id="OHS">
    <module-data>
      <category id="start-parameters">
        <data id="start-mode" value="ssl-enabled"/>
      </category>
    </module-data>
    <process-set id="HTTP_Server" numprocs="1"/>
  </process-type>
</ias-component>
```

3. Update the distributed cluster management database with the change:

```
ORACLE_HOME/dcm/bin/dcmctl updateconfig -ct opmn
```

4. Reload the modified `opmn` configuration file:

```
ORACLE_HOME/opmn/bin/opmnctl reload
```

5. Keep a non-SSL port active. The External Applications portlet communicates with the single sign-on server over a non-SSL port. The HTTP port is enabled by default. If you have not disabled the port, this step requires no action.

6. Apply the rule `mod_rewrite` to SSL configuration. This step involves modifying the `ssl.conf` file on the middle-tier computer. The file is at `ORACLE_HOME/Oracle/HttpServer/conf`. Back up the file before editing it.

Because the Oracle HTTP Server has to be available over both HTTP and HTTPS, the SSL host must be configured as a virtual host. Add the lines that follow to the

SSL Virtual Hosts section of `ssl.conf` if they are not already there. These lines ensure that the single sign-on login module in `OC4J_SECURITY` is invoked when a user logs in to the SSL host.

```
<VirtualHost ssl_host:port>
.
.
.
RewriteEngine on
RewriteOptions inherit
</VirtualHost>
```

Save and close the file.

7. Update the distributed cluster management database with the changes:

```
ORACLE_HOME/dcm/bin/dcmctl updateconfig -ct ohs
```

8. Restart the Oracle HTTP Server:

```
ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=HTTP_Server
```

9. Verify that you have enabled the single sign-on middle tier for SSL by trying to access the OracleAS welcome page, using the format `https://host:ssl_port`.

Note: If your installation has two or more middle tiers, make sure that you complete step 2 in "[Configure the Oracle HTTP servers on the single sign-on middle tiers](#)". This is a subsection of "Multiple Single Sign-On Middle Tiers, One Oracle Internet Directory," one of the deployment scenarios presented later in the chapter.

Reconfigure the Identity Management Infrastructure Database

Change all references of `http` in single sign-on URLs to `https` within the identity management infrastructure database. When you change single sign-on URLs in the database, you must also change these URLs in the `targets.xml` file on the single sign-on middle tier. `targets.xml` is the configuration file for the various "targets" that Oracle Enterprise Manager monitors. One of these targets is OracleAS Single Sign-On.

Change Single Sign-On URLs

Run the `ssocfg` script, taking care to enter the command on the computer where the single sign-on middle tier is located. Use the following syntax:

- UNIX:

```
$ORACLE_HOME/sso/bin/ssocfg.sh protocol host ssl_port
```

- Windows:

```
%ORACLE_HOME%\sso\bin\ssocfg.bat protocol host ssl_port
```

In this case, *protocol* is `https`. (To change back to HTTP, use `http`.) The parameter *host* is the host name, or server name, of the Oracle HTTP listener for the single sign-on server.

Here is an example:

```
ssocfg.sh https login.acme.com 4443
```

To determine the correct port number, examine the `ssl.conf` file. Port 4443 is the port number that the OracleAS installer assigns during installation.

If you run `ssocfg` successfully, the script returns a status 0. To confirm that you were successful, restart the OC4J_SECURITY instance:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Then try logging in to the single sign-on server at its SSL address:

```
https://host:ssl_port/pls/orasso/
```

Update targets.xml

After running `ssocfg`, update the `targets.xml` file on the single sign-on middle tier.

To update `targets.xml`:

1. Back up the file:

```
cp ORACLE_HOME/sysman/emd/targets.xml ORACLE_
HOME/sysman/emd/targets.xml.backup
```

2. Open the file and find the target type `oracle_sso_server`. Within this target type, locate and edit the three attributes that you passed to `ssocfg`:

- HTTPMachine—the server host name
- HTTPPort—the server port number
- HTTPProtocol—the server protocol

If, for example, you run `ssocfg` like this:

```
ORACLE_HOME/sso/bin/ssocfg.sh http sso.mydomain.com:4443
```

Update the three attributes this way:

```
<Property NAME="HTTPMachine" VALUE="sso.mydomain.com"/>
<Property NAME="HTTPPort" VALUE="4443"/>
<Property NAME="HTTPProtocol" VALUE="HTTPS"/>
```

3. Save and close the file.
4. Reload the OracleAS console:

```
ORACLE_HOME/bin/emctl reload
```

Protect Single Sign-On URLs

When the single sign-on server is enabled for SSL, you must specify that HTTP access be limited to those hosts that must access the server using this protocol. This is especially true in the case of those computers hosting the OracleAS installer and OracleAS Portal. In `ORACLE_HOME/sso/conf/sso_apache.conf`, locate and uncomment the three directives that follow; then provide a value for the `Allow from` parameter.

Invoke the first directive if you are installing a release 9.0.2 middle tier with a release 10.1.2 single sign-on server. This directive enables the installer to access the single sign-on server over HTTP. Replace `<your_domain_name>` with the fully qualified

host name of the middle tier on which the installer is located; then uncomment the line. If you plan to install from more than one location, enter just a domain name.

```
<Location "/pls/orasso/*[Ss][Ss][Oo][Pp][Ii][Nn][Gg]">
  Order deny,allow
  Deny from all
# Allow from <your_domain_name>
</Location>
```

OracleAS Portal must use HTTP to access the URL that provides a list of external applications. The directive that follows enables such access. Replace `<your_domain_name>` with the fully qualified Portal host name; then uncomment the directive. If you have more than one Portal database, enter just the domain name for these databases.

```
#<Location "/pls/orasso/*[Aa][Pp][Pp][Ss][Ll][Ii][Ss][Tt]">
# Order deny,allow
# Deny from all
# Allow from <your_domain_name>
#</Location>
```

Versions of OracleAS Portal that predate release 9.0.2 use HTTP to verify whether a user exists in the single sign-on database. This last directive enables verification. Again, replace `<your_domain_name>` with the fully qualified Portal host name; then uncomment the directive. If you have more than one Portal database, enter just the domain name for these databases.

```
#<Location "/pls/orasso/[Oo][Rr][Aa][Ss][Ss][Oo].wwsso_app_admin.validate_user*">
# Order deny,allow
# Deny from all
# Allow from <your_domain_name>
#</Location>
```

After editing `sso_apache.conf`, update the repository for Distributed Cluster Management:

```
ORACLE_HOME/dcm/bin/dcmctl updateConfig -v -d
```

Restart the Oracle HTTP Server and the Single Sign-On Middle Tier

Issue these two commands:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Reregister Partner Applications

Once you have enabled the single sign-on server for SSL, reregister `mod_osso` on the single sign-on middle tier and on the application middle tiers. This step configures `mod_osso` to use the effective single sign-on URL. See ["Registering mod_osso"](#) in [Chapter 4](#) for instructions. To reregister OracleAS Portal, an application integrated with the single sign-on SDK, use the `ptlconfig` tool. To learn how to use `ptlconfig`, see Appendix B in *Oracle Application Server Portal Configuration Guide*.

Deployment Scenarios

This section describes different ways that the single sign-on server may be deployed to improve availability. The section contains the following topics:

- [One Single Sign-On Middle Tier, One Oracle Internet Directory](#)
- [Multiple Single Sign-On Middle Tiers, One Oracle Internet Directory](#)

- [Multiple, Geographically Distributed Single Sign-On Instances](#)
- [Other High Availability Deployments](#)

Note: The IP addresses and host names presented in the scenarios that follow are examples only. These addresses and names may not work in an actual implementation. Substitute values that apply to your installation.

One Single Sign-On Middle Tier, One Oracle Internet Directory

The simplest and quickest way to deploy OracleAS Single Sign-On is to install OracleAS infrastructure components on the same computer. To do this, you choose the installation type "OracleAS Infrastructure" and the installation option "Identity Management and OracleAS Metadata Repository." When presented with the component list for this installation type, accept the default selected components.

Alternatively, you can install the single sign-on middle tier on a separate computer, choosing in succession "OracleAS Infrastructure," "Identity Management," and finally "Single Sign-On." This is the simplest distributed configuration.

Figure 8–1 shows the first type of installation. Figure 8–2 shows the second. The first is typical of a testing, staging, or development environment. The second is appropriate when you want to position a firewall between the single sign-on computer and the Oracle Internet Directory computer. Placing these servers on separate computers has the added benefit that it improves performance. In Figure 8–2, the single sign-on server might be situated within a DMZ, where it filters internet traffic. In this configuration, the directory and the database are available only to intranet users.

Figure 8–1 Default Single Sign-On Installation: One Computer

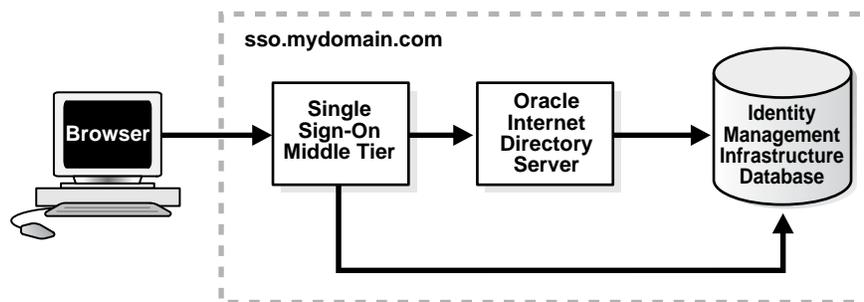
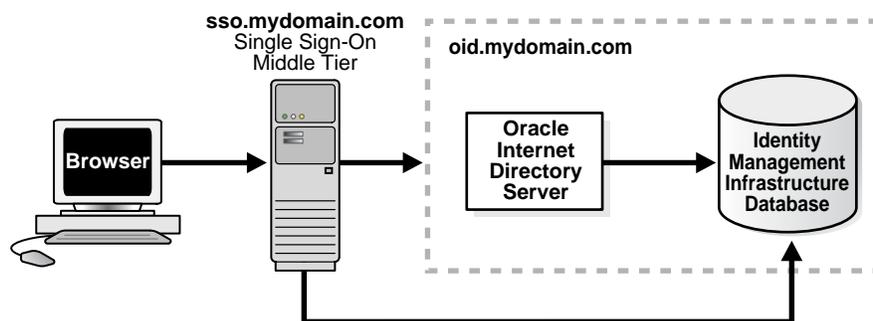


Figure 8–2 Single Sign-On Installation: Two Computers



Multiple Single Sign-On Middle Tiers, One Oracle Internet Directory

The simplest high availability scenario involves failover within the single sign-on instance itself, at the middle tier. Adding multiple middle tiers increases throughput and makes the single sign-on server more available.

In this configuration, a single HTTP load balancer is placed in front of two or more Oracle HTTP servers. At the backend is one directory server and one identity management infrastructure database. The purpose of the load balancer is to publish a single address to single sign-on partner applications while providing a farm of single sign-on middle tiers that actually service the application requests. The HTTP load balancer can detect when one of these Oracle HTTP Server instances has failed and can then fail over requests to another instance.

Usage Scenario

The usage scenario presented here assumes the following hypothetical configurations:

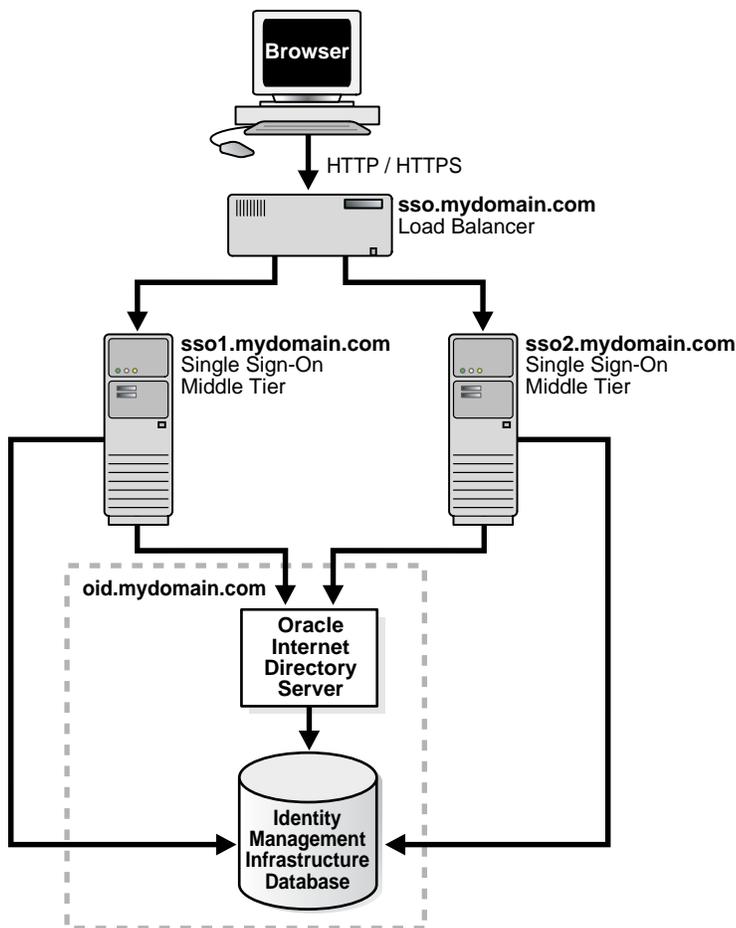
- The directory server and identity management infrastructure database are located at `oid.mydomain.com`.
- There are two single sign-on middle tiers. One is installed on host `sso1.mydomain.com`, IP address `138.1.34.172`. The other is installed on `sso2.mydomain.com`, IP address `138.1.34.173`. Both servers listen on non-SSL port `7777`. Both are configured to use the directory and identity management infrastructure database located at `oid.mydomain.com`.
- The effective URL of the single sign-on server that is published to partner applications is `sso.mydomain.com`, IP address `138.1.34.234`. The HTTP load balancer is configured to listen on `sso.mydomain.com`, port `80`. It load balances user requests between `sso1.mydomain.com` and `sso2.mydomain.com`.

Note:

- In this scenario, the load balancer is listening on port `80`, a non-SSL port number. If the load balancer is configured to use SSL to interact with the browser, a different port number must be selected. The default SSL port number is `4443`.
 - In this scenario and the one immediately following, two single sign-on middle tiers are used. There can, in fact, be any number of middle tiers.
-
-

[Figure 8–3](#) on page 8-8 shows two single sign-on middle tiers configured to use a single instance of Oracle Internet Directory.

Figure 8–3 Two Single Sign-On Middle Tiers, One Oracle Internet Directory



Configuration Steps

Setting up the single sign-on system presented in Figure 9–3 involves the following tasks:

- [Install the identity management infrastructure database, the directory server, and the single sign-on servers](#)
- [Configure the Oracle HTTP servers on the single sign-on middle tiers](#)
- [Configure the HTTP load balancer](#)
- [Configure the identity management infrastructure database](#)
- [Reregister mod_osso on the single sign-on middle tiers](#)

Install the identity management infrastructure database, the directory server, and the single sign-on servers

1. Choose a single sign-on server name that will be published to partner applications. This will also be the address of the load balancer. In the scenario presented here, the address is `sso.mydomain.com`.
2. Install the OracleAS infrastructure on `oid.mydomain.com`, choosing the option "Identity Management and OracleAS Metadata Repository." When presented with the component list for this installation type, choose Oracle Internet Directory only.

3. Install the OracleAS infrastructure on the middle tiers `sso1.mydomain.com` and `sso2.mydomain.com`, choosing the option "Identity Management." When presented with the component list for this installation type, choose OracleAS Single Sign-On. Because many tasks in OracleAS involve Oracle Delegated Administration Services, you may want to choose this component too. The configuration steps that follow apply to both components.

When the Oracle Universal Installer asks you to name the directory server associated with the two single sign-on instances, enter `oid.mydomain.com`.

Note: The OracleAS installer, by default, assigns port numbers from a range of numbers. If you want the installer to assign a different port number to a component, see the section about static port numbers in the requirements chapter of *Oracle Application Server Installation Guide*.

Configure the Oracle HTTP servers on the single sign-on middle tiers

When a load balancer is placed between the user and the Oracle HTTP Server, the effective URL of the single sign-on server changes. The Oracle HTTP configuration file `httpd.conf` on both single sign-on middle tiers must be modified to reflect this change. This file is found at `ORACLE_HOME/Apache/Apache/conf`.

1. Edit the following lines in `httpd.conf` on `sso1.mydomain.com` and `sso2mydomain.com`:

```
KeepAlive off
ServerName sso.mydomain.com
Port 80
```

Note:

- If multiple ports are listed in `httpd.conf`, the effective port must appear last.
 - Make sure that the IP check option is turned off. You can use the single sign-on administration UI to do this. See [Using the Edit SSO Server Page to Configure the Server](#) in [Chapter 2](#).
-

This step configures the Oracle HTTP servers at the single sign-on middle tiers to listen at the effective URL, which, in the scenario presented, is `sso.mydomain.com`.

2. If you configure SSL between the browser and the load balancer, and the SSL connection terminates at the load balancer, configure `mod_certheaders` on both `sso1.mydomain.com` and `sso2.mydomain.com`. This module enables the Oracle HTTP Server to treat requests that it receives over HTTP as SSL requests. Add the lines that follow to `httpd.conf`. You can place them at the end of the file. Where they appear is unimportant.

- a. Enter this line to load the module:

UNIX:

```
LoadModule certheaders_module libexec/mod_certheaders.so
```

Windows:

```
LoadModule certheaders_module modules/ApacheModuleCertHeaders.dll
```


Reregister mod_osso on the single sign-on middle tiers

On `sso1.mydomain.com`, reregister `mod_osso` as the partner application `sso.mydomain.com`; then copy the registration record to `sso2mydomain.com`.

To reregister `mod_osso` on `sso1.mydomain.com`:

1. Set the environment variable `ORACLE_HOME` to point to the Oracle home for `sso1.mydomain.com`. To learn how to do this on different platforms see "Command Example" in Chapter 4.
2. Run the registration script. For the URLs, be sure to substitute values appropriate for your installation. Following the example, the script creates a partner application called `sso.mydomain.com`.

- UNIX:

```
$ORACLE_HOME/sso/bin/ssoreg.sh
-oracle_home_path orcl_home_path
-site_name load_balancer_name
-config_mod_osso TRUE
_mod_osso_url http://load_balancer_host:load_balancer_port
```

- Windows:

```
%ORACLE_HOME%\sso\bin\ssoreg.bat
-oracle_home_path orcl_home_path
-site_name load_balancer_name
-config_mod_osso TRUE
_mod_osso_url http://load_balancer_host:load_balancer_port
```

3. Update the configuration for the Oracle HTTP Server in the distributed cluster management repository:

```
ORACLE_HOME/dcm/bin/dcmctl updateconfig
```

4. Restart the middle tier at `sso1.mydomain.com`:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

To update `mod_osso` on `sso2.mydomain.com`:

1. On the computer `sso2.mydomain.com`, log in to the single sign-on administration pages as the single sign-on administrator. Be sure to log in to the externally published address of the single sign-on server:

```
http://sso.mydomain.com/pls/orasso
```

2. Use the Administer Partner Applications page to delete the existing entry for the partner application `sso2.mydomain.com`.
3. Copy the `osso.conf` file from the computer `sso1.mydomain.com`. Make sure that you use binary mode if you FTP the file. Copy the file to `ORACLE_HOME/Apache/Apache/conf/osso`.
4. Synchronize the Distributed Cluster Management repository with the file copy. You do this by running this command on `sso2.mydomain.com`:

```
ORACLE_HOME/Apache/Apache/bin/ssotransfer ORACLE_HOME/Apache/Apache/conf/
osso/osso.conf
```

Note: The `ssotransfer` command should not be used to synchronize the Distributed Cluster Management repository with the `mod_osso` configuration file created for a virtual host. To learn how to register `mod_osso` for a virtual host, see "[Configuring mod_osso with Virtual Hosts](#)" in [Chapter 4](#).

5. Update the configuration for the Oracle HTTP Server in the distributed cluster management repository:

```
ORACLE_HOME/dcm/bin/dcmctl updateConfig
```

6. Restart the middle tier at `sso2.mydomain.com`:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

7. If Oracle Delegated Administration Services is installed, change its base URL, using Oracle Directory Manager:

- a. Start the tool:

- * UNIX:

```
ORACLE_HOME/bin/oidadmin
```

- * Windows:

Click the following in succession: **Start, Programs, OracleAS 10g Infrastructure, Integrated Management Tools, Oracle Directory Manager**

- b. Log in to Oracle Directory Manager as `cn=orcladmin`.

- c. Go to the entry that contains the `orcldasurlbase` attribute:

```
cn=OperationURLs,cn=DAS,cn=Products,cn=OracleContext,Entry Management
```

- d. Change the attribute to the following value:

```
http://sso.mydomain.com/
```

Make sure that you include the backslash after the host name.

Note: This step assumes that you are using the default HTTP port 80. If you are not using the default, include the port number:
`http://sso.mydomain.com:port.`

- e. Test the partner application `oiddas`:

```
http://sso.mydomain.com/oiddas
```

8. Test the single sign-on administration application:

```
http://sso.mydomain.com/pls/orasso
```

Multiple Single Sign-On Middle Tiers, Replicated Oracle Internet Directory

In local area networks that experience high traffic, it may be beneficial to supplement multiple single sign-on middle tiers with replicated instances of Oracle Internet Directory. This arrangement provides failover not only at the middle tier, but also at the directory server. It is useful for managing rolling upgrades because replica nodes can be removed for maintenance while other nodes continue to serve users.

To learn how to deploy an Oracle Identity Management system that uses multimaster replication, see the appendix about this topic in *Oracle Identity Management Concepts and Deployment Planning Guide*. The appendix provides detailed steps for configuring every component in the identity management infrastructure.

Multiple, Geographically Distributed Single Sign-On Instances

Server availability is critical for an enterprise whose operations are widely distributed geographically. If the enterprise uses a single server to authenticate remote users over a wide area network, the authentication time can be lengthy. To shorten network round-trips and speed access to applications, the enterprise can implement multiple, geographically distributed instances of the single sign-on server. This arrangement enables users to travel to remote locations and be authenticated by the nearest server, regardless of where applications are located.

In this scenario, single sign-on database tables are replicated over either a local area network or a wide area network. The DNS server located at each single sign-on middle tier site must be configured to resolve the effective address of the single sign-on server to the single sign-on instance that is nearest to the user.

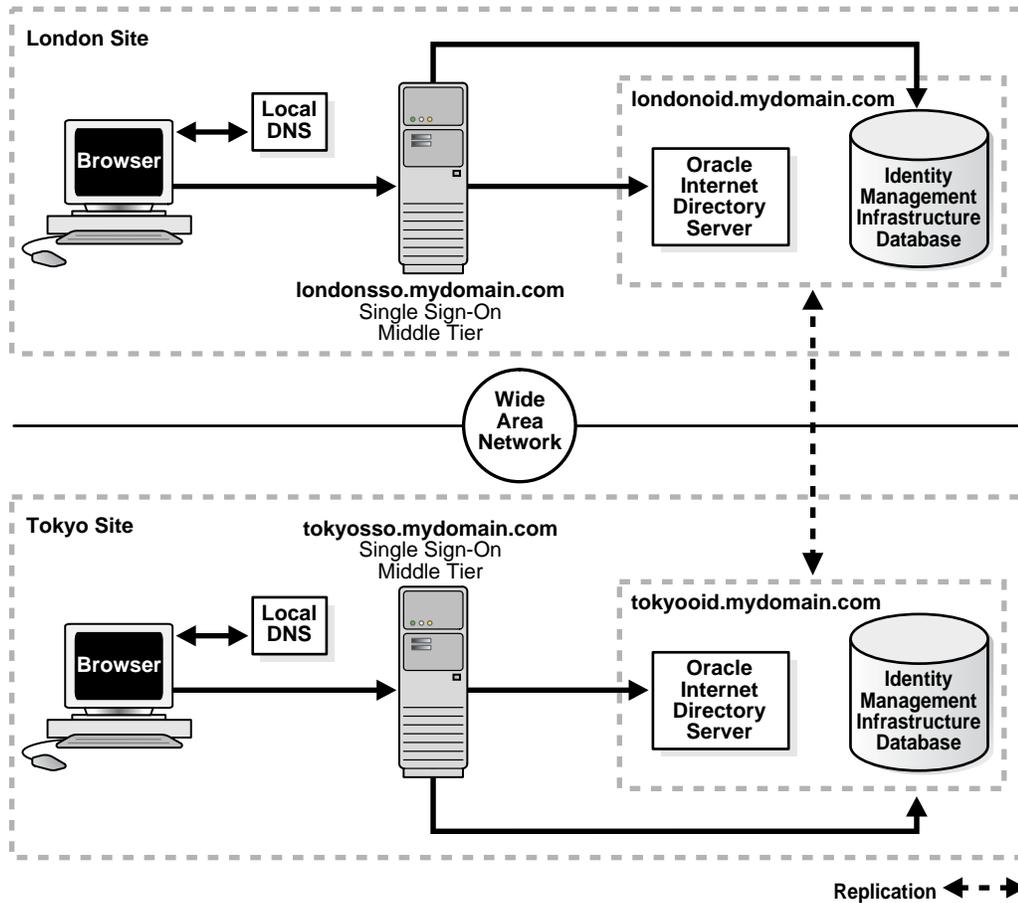
Usage Scenario

The usage scenario presented here assumes the following hypothetical configurations:

- There are two single sign-on middle tiers: `london.sso.mydomain.com` and `tokyo.sso.mydomain.com`. The effective address of the single sign-on server is `sso.mydomain.com`.
- There are two directory servers/identity management infrastructure databases associated with the two single sign-on middle tiers: `londonoid.mydomain.com` and `tokyooid.mydomain.com`.
- For replication purposes, `londonoid.mydomain.com` is the master definition site (MDS), the site from which the replication scripts are run and data is first replicated. `tokyooid.mydomain.com` is the remote master site (RMS), the site to which data is replicated.
- The single sign-on middle tiers and the identity management infrastructure databases are located on separate computers.

[Figure 8–4](#) on page 8-14 depicts what this geographically distributed system looks like once it is deployed.

Figure 8–4 A Highly Available, Geographically Distributed Single Sign-On System



Configuration Steps

The geographically dispersed single sign-on system shown in [Figure 8–5](#) incorporates steps presented in "[Multiple Single Sign-On Middle Tiers, One Oracle Internet Directory](#)" and "[Configuring the Identity Management Database for Replication](#)".

1. Install Oracle Internet Directory on the MDS, `londonoid.mydomain.com`, and on the RMS, `tokyooid.mydomain.com`; then set these servers up as a replication group. For instructions, see *Oracle Internet Directory Administrator's Guide*. These procedures cover both installation and replication. For replication concepts, see also *Oracle Internet Directory Administrator's Guide*.
2. Install the OracleAS infrastructure on the middle tier `londonosso.mydomain.com`, choosing the option "Identity Management." When presented with the component list for this installation type, choose "Single Sign-On." When the Oracle Universal Installer asks you to name the directory server associated with this single sign-on instance, enter `londonoid.mydomain.com`.
3. Repeat step 2, this time on middle tier `tokyosso.mydomain.com`. In this case, you must associate the single sign-on server with the directory server located at `tokyooid.mydomain.com`.
4. Synchronize single sign-on schema passwords between the MDS database and the RMS database. To do this, complete step 2 in "[Configuring the Identity Management Database for Replication](#)".

5. Although two single sign-on instances are now running at different locations, only one effective server URL is published to partner applications. Configure the single sign-on server to use this URL. In this scenario, we call the URL `sso.mydomain.com`. See "[Configure the Oracle HTTP servers on the single sign-on middle tiers](#)" for instructions.
6. Add a DNS alias, `sso.mydomain.com`, that points to the single sign-on middle tiers. Configure the DNS server to rout the user to the nearest middle tier when single sign-on authentication is required. When, for example, a London user is redirected to `http://sso.mydomain.com`, the DNS server should route the user to `http://london.sso.mydomain.com`. Similarly, a Tokyo user redirected to `http://sso.mydomain.com` should be routed to `http://tokyo.sso.mydomain.com`.

Note that some advanced DNS server products may be able to route users to the nearest server based on the geographic location.

Other High Availability Deployments

OracleAS supports cold failover clusters, disaster recovery, and backup and recovery for single sign-on as well as for other OracleAS components.

OracleAS Cold Failover Cluster (Infrastructure)

A cold failover cluster is a group of loosely coupled computers that together provide a single view of network services. Cluster software enables the logical IP address and processes of the primary node to be moved to a secondary node in the event that the primary fails. The node running the infrastructure is "hot." The node waiting to take over is "cold." Hence the term cold failover.

To learn more about cold failover clusters, see the chapter about infrastructure high availability in *Oracle Application Server High Availability Guide*.

Disaster Recovery

A disaster recovery deployment consists of two identically configured sites—one primary (production), the other secondary (standby). Both sites may be dispersed geographically and connected by a wide area network. When the primary site becomes unavailable due to a disaster, the secondary site can become operational within a reasonable amount of time. Client requests are always routed to the site playing the production role. After failover occurs, client requests are routed to the secondary site, which then assumes the production role. Both sites have identical middle tier servers, and these servers are also identical between the two sites. To learn more about disaster recovery, see the chapter devoted to this topic in *Oracle Application Server High Availability Guide*.

Backup and Recovery

Backup and recovery are terms used to describe strategies and procedures for preventing data loss and reconstructing lost data. To learn more about backup and recovery, see the chapter devoted to this topic in *Oracle Application Server Administrator's Guide*.

Replicating the Identity Management Database

This section describes how to replicate the identity management database between two or more instances. Note that OracleAS Single Sign-On and Oracle Internet

Directory share the scripts and procedures that replicate database tables. Before continuing with this section, become familiar with the following material:

- "Directory Replication Concepts" in *Oracle Internet Directory Administrator's Guide*
- "Oracle Directory Replication Administration" in *Oracle Internet Directory Administrator's Guide*
- "Replication-Management Command-Line Tools Syntax" also in *Oracle Internet Directory Administrator's Guide*

The section covers the following topics:

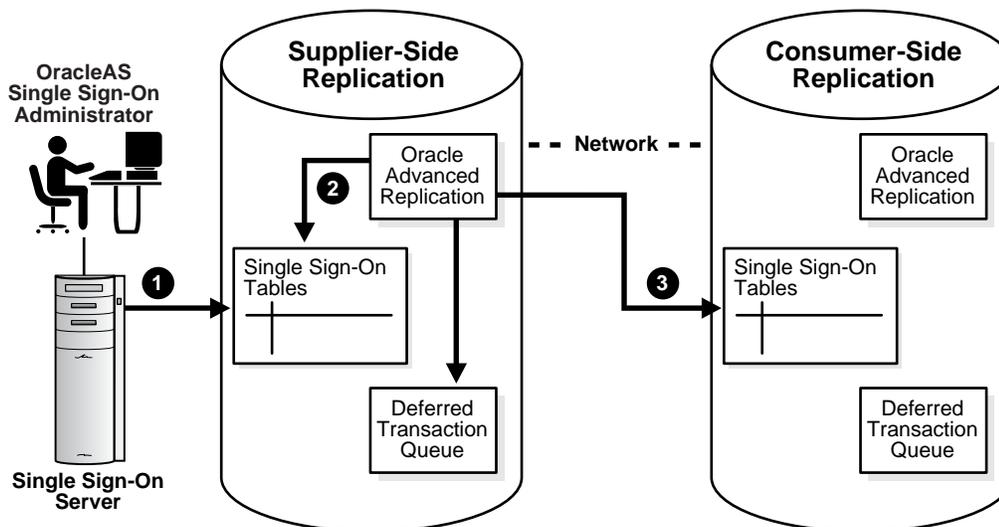
- [The Replication Mechanism](#)
- [Configuring the Identity Management Database for Replication](#)
- [Adding a Node to a Replication Group](#),
- [Deleting a Node from a Replication Group](#)

The Replication Mechanism

The identity management infrastructure uses Oracle9i Advanced Replication to replicate tables between two databases. This feature propagates data changes between databases asynchronously. In other words, suppliers write changes to single sign-on tables and periodically send batched changes to consumers, servers that replicate this data. All of the servers in a multiple, geographically distributed system can either propagate or receive data. This arrangement is called multimaster replication.

Figure 8-5 illustrates the process.

Figure 8-5 Multimaster Replication Architecture



1. The single sign-on administrator uses the single sign-on administration application to modify single sign-on partner applications or configuration data. This process modifies the corresponding table entry in the identity management infrastructure database.
2. Oracle9i Advanced Replication copies the change to a deferred transaction queue.

3. At a scheduled interval, Oracle9i Advanced Replication pushes transactions in the deferred transaction queue to the single sign-on table on the consumer side.

Configuring the Identity Management Database for Replication

Before proceeding with this section, become familiar with multimaster replication concepts in *Oracle Internet Directory Administrator's Guide*.

You may also want to familiarize yourself with the deployment scenario presented in "[Multiple, Geographically Distributed Single Sign-On Instances](#)". This section describes the circumstances under which single sign-on replication occurs.

The sequence for enabling the identity management database for replication is as follows:

1. Follow the instructions in *Oracle Internet Directory Administrator's Guide* to install and configure a multimaster replication group. Note that single sign-on tables are replicated as part of this process.
2. After running the replication scripts, the administrator must run scripts to synchronize schema passwords among replicated nodes and to establish a connection between the single sign-on server and the directory.

On the MDS, run the `ssoReplSetup.jar` tool to synchronize single sign-on schema passwords between the MDS database and the RMS database. This step must be repeated for each RMS. [Table 8-1](#) on page 8-18 defines the tool parameters.

To run the script:

- a. Go to `ORACLE_HOME/sso/lib`.
- b. Set the library path:

* UNIX (csh and tcsh):

```
setenv LD_LIBRARY_PATH $ORACLE_HOME/lib32:$LD_LIBRARY_PATH
```

* Windows:

```
set PATH=%ORACLE_HOME%\bin;%PATH%
```

- c. Issue this command:

```
ORACLE_HOME/jdk/bin/java -jar ssoReplSetup.jar
[-prompt]
mds_oid_host
mds_oid_port
mds_oid_admin
mds_oid_password
mds_ssl_enabled
rms_oid_host
rms_oid_port
rms_oid_admin
rms_oid_password
rms_ssl_enabled
rms_db_sys_password
[-help]
```

Table 8–1 Parameters for ssoReplSetup

Parameter	Description
<i>mds_oid_host</i>	Host name of the MDS directory server.
<i>mds_oid_port</i>	Port number of the MDS directory server.
<i>mds_oid_admin</i>	Bind DN—that is, the user authenticating to the MDS directory server.
<i>mds_oid_password</i>	Bind password of the MDS directory server.
<i>mds_ssl_enabled</i>	Indicates whether the MDS has SSL enabled. Can be either Y or N. Note that this parameter is case insensitive. This parameter is usually set to Y because the directory and the single sign-on server communicate over SSL by default.
<i>rms_oid_host</i>	Host name of the RMS directory server.
<i>rms_oid_port</i>	Port number of the RMS directory server.
<i>rms_oid_admin</i>	Bind DN—that is, the user authenticating to the RMS directory server.
<i>rms_oid_password</i>	Bind password of the RMS directory server.
<i>rms_ssl_enabled</i>	Indicates whether the RMS has SSL enabled. Can be either Y or N. Note that this parameter is case insensitive. This parameter is usually set to Y because the directory and the single sign-on server communicate over SSL by default.
<i>rms_db_sys_password</i>	SYS password of the RMS database.
-prompt	Prompts you for all values from the console.
-help	Displays usage notes.

Note: Repeat step 2 for each additional RMS node.

Adding a Node to a Replication Group,

If you want to add a node to an existing single sign-on replication group and have not replicated Oracle Internet Directory to this node, follow the instructions in *Oracle Internet Directory Administrator's Guide*. To configure this new node for single sign-on, install the single sign-on middle tier and repeat step 2 in "[Configuring the Identity Management Database for Replication](#)".

Deleting a Node from a Replication Group

To delete a node from the single sign-on replication group, follow the instructions in *Oracle Internet Directory Administrator's Guide*.

Deploying OracleAS Single Sign-On with a Proxy Server

OracleAS Single Sign-On can have reverse proxies deployed in front of it. Proxies fulfill various functions:

- They hide the host name of the single sign-on server.
- They terminate an SSL connection at the proxy instead of at the single sign-on server.

- They limit the number of ports exposed on a firewall

Whatever proxy you use in front of the single sign-on server, the configurations that follow apply. They assume that you have already installed OracleAS Single Sign-On and the proxy server. To install the proxy, use instructions provided by your proxy vendor.

Note: These instructions also apply to virtual hosts.

Turn Off IP Checking

In network configurations where a range of distinct proxy addresses "front" the single sign-on server, the single sign-on IP check feature must be turned off. IP check is turned off by default, but to verify this, go to the Edit SSO Server page. To learn how to access this page, see ["Accessing the Administration Pages"](#) in [Chapter 2](#). Once into the Edit SSO Server page, make sure that the box **Verify IP addresses for requests made to the SSO Server** is deselected.

Enable the Proxy Server

To enable a proxy server, do the following:

1. Run the `ssocfg` script on the single sign-on middle tier. This script changes the host name stored in the single sign-on server to the proxy host name. Use the following command syntax, entering values for the protocol, host name, and port of the proxy server:

- UNIX:

```
$ORACLE_HOME/sso/bin/ssocfg.sh http proxy_server_name proxy_port
```

- Windows:

```
%ORACLE_HOME%\sso\bin\ssocfg.bat http proxy_server_name proxy_port
```

If the server is configured for SSL, substitute `https` for `http` and an SSL port for a non-SSL port.

After running `ssocfg`, update the `targets.xml` file on the single sign-on middle tier. See ["Update targets.xml"](#) for instructions.

2. Add the lines that follow to the `httpd.conf` file on the single sign-on middle tier. The file is at `ORACLE_HOME/Apache/Apache/conf`.
 - a. These lines change the directive `ServerName` from the name of the actual server to the name of the proxy:

```
KeepAlive off
ServerName proxy_host_name
Port proxy_port
```

Note that if you are using SSL, the port must be an SSL port such as 4443.

- b. Enter the following lines to ensure that the single sign-on login module in `OC4J_SECURITY` is invoked when a user logs in to the proxy:

```
<VirtualHost proxy_host_name:proxy_port>
.
.
.
RewriteEngine On
```

```
RewriteOptions inherit
</VirtualHost>
```

- c. (SSL only) If you have configured SSL communication between just the browser and the proxy server, configure `mod_certheaders` on the middle tier. This module enables the Oracle HTTP Server to treat HTTP proxy requests that it receives as SSL requests. Add the lines that follow to `httpd.conf`. You can place them at the end of the file. Where they appear is unimportant.

- * Enter this line to load the module:

UNIX:

```
LoadModule certheaders_module libexec/mod_certheaders.so
```

Windows:

```
LoadModule certheaders_module modules/ApacheModuleCertHeaders.dll
```

- * If you are using OracleAS Web Cache as a proxy, enter this line:

```
AddCertHeader HTTPS
```

If you are using a proxy other than OracleAS Web Cache, enter this line:

```
SimulateHttps on
```

3. Reregister `mod_osso` on the single sign-on middle tier. This step configures `mod_osso` to use the proxy host name instead of the actual host name. To learn how to run the registration tool, see "[Registering mod_osso](#)" in [Chapter 4](#).
4. Update the Distributed Cluster Management schema:


```
ORACLE_HOME/dcm/bin/dcmctl updateconfig
```
5. Restart the single sign-on middle tier:


```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```
6. If you are deploying more than one single sign-on middle tier, repeat steps 2 through 4 on each additional middle tier.
7. Log in to the single sign-on server, using the single sign-on login URL:

```
http://proxy_host_name:proxy_port/pls/orasso/
```

This URL takes you to the single sign-on home page. If you are able to log in, you have configured the proxy correctly.

Setting Up Directory Synchronization for User Nickname Changes

The single sign-on database uses the user nickname to store and reference user data for external applications. In the event that the nickname attribute value changes in Oracle Internet Directory, users are forced to reenter their credentials when they log in with a new user ID. For their convenience, changes to their user names can be automatically synchronized between the directory and the single sign-on database. This synchronization mechanism, offered through Oracle Directory Integration and Provisioning, also deletes the external application data from the single sign-on database when a user's entry is deleted from the directory.

To synchronize nickname changes between the directory and the single sign-on database, follow these steps:

1. Start the Oracle Directory Integration and Provisioning server. For instructions, see the chapter about administration tools in *Oracle Identity Management Integration Guide*.

2. Load the synchronization package. First, navigate to `ORACLE_HOME/sso/admin/plsql/sso`; then connect to the single sign-on schema:

```
sqlplus orasso/password
```

See [Appendix B](#) to learn how to obtain the `orasso` password.

3. Run these packages in the order listed:

```
SQL> @ssodip.sql
SQL> @ssodip.pks
SQL> @ssodip.pkb
```

4. Register the single sign-on profile with Oracle Internet Directory. You do this by running the Provisioning Subscription Tool (`oidprovtool`):

```
ORACLE_HOME/bin/oidprovtool
operation=create
ldap_host=oid_host
ldap_port=oid_port
ldap_user_dn=cn=orcladmin
ldap_user_password=orcladmin_password
schedule=synchronization_interval_in_seconds
organization_dn=realm_DN
application_dn=orclApplicationCommonName=ORASSO_SSOSERVER,cn=SSO,
cn=Products,cn=OracleContext
interface_name=LDAP_NOTIFY interface_type=PLSQL
interface_connect_info=sso_database_host:sso_database_port:sso_
database_SID:orasso:orasso_schema_password
event_subscription=USER:user_search_base_for_realm:ADD(attribute_type)
event_subscription=USER:user_search_base_for_realm:MODIFY(attribute_type)
event_subscription=USER:user_search_base_for_realm:DELETE
```

If changes to the realm occur, reregister the profile. The user search base may change. Or the nickname attribute type. For example, the `uid` attribute may replace the `cn` attribute.

For help using `oidprovtool`, see the syntax appendix in *Oracle Internet Directory Administrator's Guide*.

5. Give the Oracle Directory Integration and Provisioning server privileges to proxy as `orasso`. This involves modifying the `orasso` entry in the directory.

First create an LDIF file:

```
dn: orclApplicationCommonName=ORASSO_SSOSERVER,cn=SSO,cn=Products,
cn=OracleContext
changetype: modify
add: orclaci
orclaci: access to entry by group="cn=odisgroup,cn=odi,cn=oracle internet
directory" (proxy)
```

6. Load the LDIF file into the directory as the super user `cn=orcladmin`.
7. Make sure that the Oracle Directory Integration and Provisioning server is running.

Depending upon how synchronization is scheduled, there may be a lapse between the time changes are made in the directory and the time they are synchronized with the single sign-on server. Because of this lapse, users whose user IDs have changed gain access to external applications only when synchronization finally occurs.

Enabling Support for Application Service Providers

This chapter explains how to enable the single sign-on server to support multiple realms within one instance of the Oracle Identity Management infrastructure.

The chapter contains the following topics:

- [Application Service Providers: Deciding to Deploy Multiple Realms](#)
- [Setting Up and Enabling Multiple Realms](#)
- [How the Single Sign-On Server Enables Authentication to Multiple Realms](#)
- [Configuring the Single Sign-On Server for Multiple Realms](#)
- [Granting Administrative Privileges for Multiple Realms](#)

Application Service Providers: Deciding to Deploy Multiple Realms

Application service providers are companies that install and maintain Oracle and non-Oracle applications and make them available to their customers, typically for a fee. These companies achieve economies of scale by serving multiple sets of users within the same application instance. The application service provider may, for example, use different realms, or namespaces, within one instance of the Oracle Identity Management infrastructure to set and store Oracle configuration information unique to different customers.

If user IDs are the only criterion for deciding whether to deploy multiple realms, and there are no ID conflicts, Oracle recommends maintaining users in a single, default realm. The application service provider may, for example, be one who has users log in with an email ID, which is unique. In situations where user IDs conflict, separate realms may be unavoidable. Note, too, that the decision to deploy multiple realms affects how Oracle 10g middle-tier components and customer applications are deployed.

Note: To gain a thorough understanding of Oracle Identity Management, see *Oracle Identity Management Concepts and Deployment Planning Guide*.

Setting Up and Enabling Multiple Realms

The work involved in setting up multiple realms may require resources and administrative overhead that exceed those of OracleAS Single Sign-On. Other

components are involved in the process. In fact, realm configuration is a three-part process that consists of the following:

- Creating realms in Oracle Internet Directory
- "Turning on" multiple realms in OracleAS Single Sign-On
- Making partner applications aware of identity management realms

The first process is discussed in *Oracle Internet Directory Administrator's Guide*. The second is the subject of this chapter. The third is discussed in product-relevant documentation.

How the Single Sign-On Server Enables Authentication to Multiple Realms

The authentication sequence for single sign-on to multiple realms is much the same as it is for single sign-on in a single, default realm. The only difference from the user's perspective is that, when the user affiliated with the first type of realm is presented with the login screen (see [Figure 9-1](#) on page 9-3), he or she must enter not only a user name and password but also a new credential: the realm nickname. The value entered can be case insensitive.

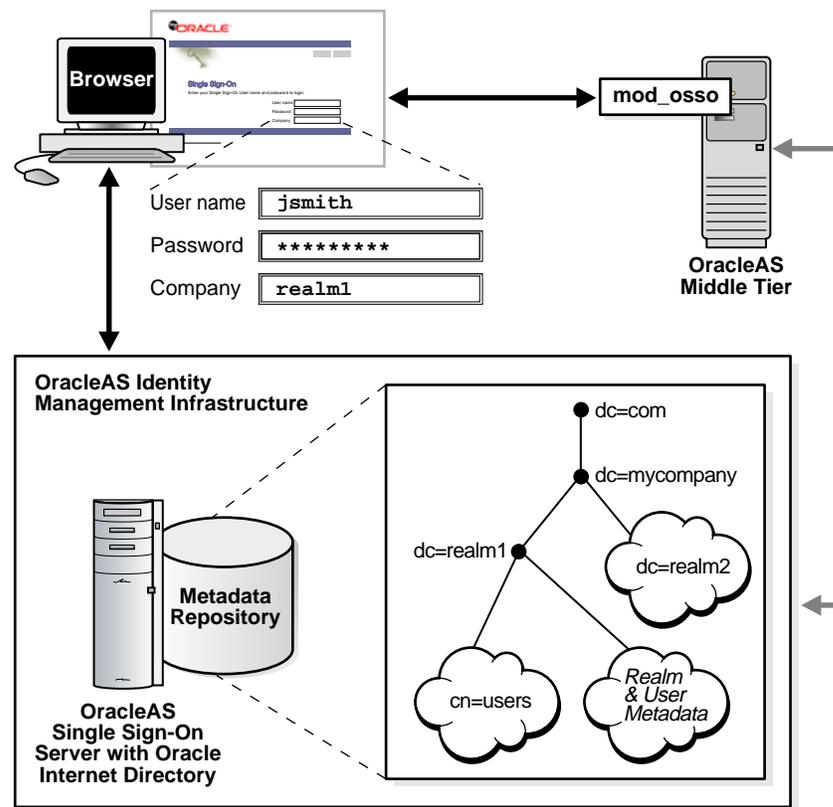
This section covers the following topics:

- [Locating Realms in Oracle Internet Directory](#)
- [Validating Realm-Affiliated Users to Partner Applications](#)

Locating Realms in Oracle Internet Directory

Once a user has entered his credentials, both his realm nickname and user name are mapped to entries in Oracle Internet Directory. More specifically, the single sign-on server uses directory metadata to find the realm entry in the directory. Once it finds this entry, the single sign-on server uses realm metadata to locate the user. Once the user's entry is found, his password, an attribute of his entry, is validated. And once his password is validated, he is authenticated.

Figure 9–1 The Big Picture: Single Sign-On in Multiple Realms



Validating Realm-Affiliated Users to Partner Applications

Presented with two users, both with the same nickname but affiliated with different realms, a partner application requires some mechanism for distinguishing between these users. The application requires such a mechanism because it must be able to adapt content—an OracleAS Portal page with stock news and stock listings, for instance—to match the needs of the realm requesting it. Accordingly, OracleAS release 9.0.4 adds the realm nickname, realm DN, and realm GUID as attributes passed to `mod_osso`. Recall that `mod_osso` sets a cookie, storing the retrieved attributes as HTTP headers. When deciding what content to offer up, the application may use function calls to retrieve any one of these attributes from `mod_osso` headers.

For detailed information about `mod_osso` headers and the methods used to access them, see the chapter about `mod_osso` in *Oracle Identity Management Application Developer's Guide*.

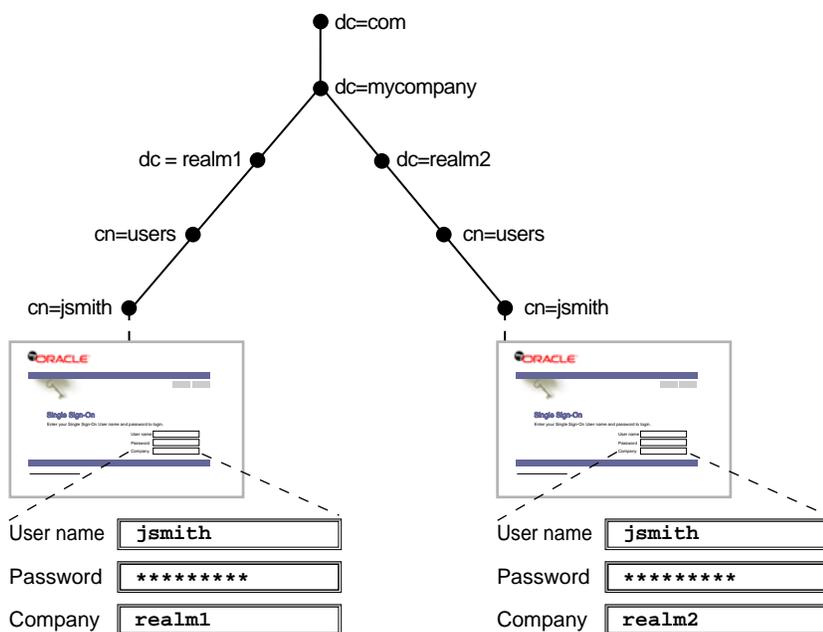
Figure 9–2 on page 9-4 shows how applications running in `mod_osso` see HTTP headers for two users with the same nickname who are affiliated with two different realms. The application uses the headers that appear in bold face to distinguish between the two users. The host, or default realm, in this case is `mycompany.com`.

Figure 9–2 *mod_ossso Headers for Users with the Same Name*

```

Realm1
REMOTE_USER = "jsmith"
HTTP_OSSO_USER_DN = "cn=jsmith,cn=users,dc=realm1,dc=mycompany,dc=com"
HTTP_OSSO_USER_GUID = "5D92F6E61F7A4CA7854BF59BA890EBFC"
HTTP_OSSO_SUBSCRIBER = "REALM1"
HTTP_OSSO_SUBSCRIBER_DN = "dc=realm1,dc=mycompany,dc=com"
HTTP_OSSO_SUBSCRIBER_GUID = "F76B7C1945AB4F8DB9391B45D3021334"

Realm2
REMOTE_USER = "jsmith"
HTTP_OSSO_USER_DN = "cn=jsmith,cn=users,dc=realm2,dc=mycompany,dc=com"
HTTP_OSSO_USER_GUID = "6786605E41604E18B74D5B90708F5CA4"
HTTP_OSSO_SUBSCRIBER = "REALM2"
HTTP_OSSO_SUBSCRIBER_DN = "dc=realm2,dc=mycompany,dc=com"
HTTP_OSSO_SUBSCRIBER_GUID = "D9D52D0DC8FF4B6FAF19A795B9B2EA23"
    
```



Configuring the Single Sign-On Server for Multiple Realms

Configuring the single sign-on server for multiple realms involves creating an entry for each realm in the single sign-on schema. Every realm that you create in Oracle Internet Directory must have a corresponding entry in the single sign-on schema.

Note:

- Create the realm in the directory before creating it in the single sign-on schema.
- The configuration scripts that follow work only on UNIX platforms. They cannot be run on Windows platforms.

To configure the single sign-on server for multiple realms, complete the steps that follow. Steps 1, 2, and 5 must be completed only once because these steps enable the

server for multiple realms. Steps 3 and 4 must be completed each time you add a realm.

1. Ensure that you have installed the OracleAS infrastructure and the single sign-on server.
2. Go to `ORACLE_HOME/sso/admin/plsql/wwhost`.

Run the `enblhstg.csh` script using the syntax that follows. See [Table 9-1](#) on page 9-6 for an explanation of script parameters:

```
enblhstg.csh -mode sso
              -sc sso_schema_connect_string
              -ss orasso
              -sw sso_schema_password
              -h oid_host_name
              -p oid_port
              -d "cn=orcladmin"
              -w oid_bind_password
```

Note: If the single sign-on server is part of a distributed deployment, make sure that you run the script on the computer that contains the metadata repository for OracleAS.

Here is an example:

```
enblhstg.csh -mode sso
              -sc webdbsvr2:1521:s901dev3
              -ss orasso
              -sw xyz
              -h dlsun670.us.oracle.com
              -p 389
              -d "cn=orcladmin"
              -w welcome123
```

3. Add realms to Oracle Internet Directory. To do this, follow the instructions in *Oracle Internet Directory Administrator's Guide*.
4. Create an entry for the realm in the single sign-on database. Use the script `ORACLE_HOME/sso/admin/plsql/wwhost/addsub.csh`. Again, if your single sign-on server is part of a distributed deployment, run the script on the computer that contains the metadata repository for OracleAS.

Use the following syntax to execute the script:

```
addsub.csh -name realm_nickname
            -id realm_ID
            -mode sso
            -sc sso_schema_connect_string
            -ss sso_schema_name
            -sw sso_schema_password
            -h oid_host_name
            -p oid_port
            -d oid_bind_dn
            -w oid_bind_dn_password
            -sp sys_schema_password
```

[Table 9-1](#) on page 9-6 defines parameters for both `enblhstg.csh` and `addsub.csh`.

Table 9–1 Parameters for *enblhstg.csh* and *addsub.csh*

Parameter	Description
-mode	The value here must be <i>sso</i> .
-sc	The connect string for the single sign-on schema. Use the format <i>host:port:sid</i> .
-ss	The name of the single sign-on schema. This parameter must be <i>orasso</i> .
-sw	The password for the single sign-on schema. See Appendix B to learn how to obtain it.
-h	The host name for the Oracle Internet Directory server.
-p	The port number for the Oracle Internet Directory server.
-d	The bind DN for the Oracle Internet Directory server. The value of this parameter is <i>cn=orcladmin</i> . This is the directory super user.
-w	The password for the Oracle Internet Directory super user, <i>cn=orcladmin</i> .
-name	The realm nickname. This is the value that you enter into the company field on the login page.
-id	The realm ID. Choose an integer greater than 1. The value 1 is reserved for the default realm. The single sign-on server uses realm IDs internally, as an index.
-sp	The <i>sys</i> schema password. This password is chosen during the installation of OracleAS.

Note:

- When the script asks you about the duplicated subscriber entry, choose the option to use the existing entry.
- If you are creating a one-level realm, include the parameters *-sd default_realm_id* and *-type db* in the script.

5. Update the sample login page with the multiple realm version of the page. You do this by editing *ORACLE_HOME/j2ee/OC4J_SECURITY/applications/sso/web/jsp/login.jsp*.

Note: In a distributed deployment, this file is located on the single sign-on middle tier.

After making a backup copy of the file, uncomment this section:

```
<!-- UNCOMMENT TO ENABLE MULTIPLE REALM SUPPORT
<tr>
<th id="c4" ALIGN="RIGHT"><label for="c4"><font class="OraFieldText">
<%=msgBundle.getString(ServerMsgID.COMPANY_LBL)%></font></label></th>
<td headers="c4"> <INPUT TYPE="text" SIZE="30" MAXLENGTH="50" ID="c4"
NAME="subscribername" value="<%=str_subscriber%>"></td>
</tr>
-->
```

6. Stop and then start the single sign-on middle tier:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server  
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Granting Administrative Privileges for Multiple Realms

Oracle Internet Directory propagates the DIT structure of the default realm across realms when it creates these realms. Note, however, that the users, groups, and privileges that exist in the DIT of the default realm are not propagated. The directory super user or realm administrator must assign, or reassign, privileges, using Oracle Directory Manager. To learn how to use the tool for this purpose, see "[Granting Administrative Privileges](#)" in [Chapter 2](#).

Monitoring the Single Sign-On Server

This chapter explains how to use Oracle Enterprise Manager, the Oracle system management console, to monitor the single sign-on server.

The chapter contains the following topics:

- [Accessing the Monitoring Pages](#)
- [Interpreting and Using the Home Page on the Standalone Console](#)
- [Interpreting and Using the Details of Login Failures Page](#)
- [Updating the Port Property for the Single Sign-On Monitoring Target](#)
- [Using the OracleAS Web Cache Instance to Monitor the Server](#)
- [Monitoring a Single Sign-On Server Enabled for SSL](#)

Accessing the Monitoring Pages

The single sign-on monitoring UI on the standalone console consists of two pages: the home page and the Details of Login Failures page. The first provides general information about server load and user activity. The second provides a login failure profile for a particular user.

To access the home page for single sign-on monitoring:

1. Go to the standalone console for the instance of Oracle Enterprise Manager that you want to administer. This is effected by entering the host name of the computer hosting the OracleAS instance and the port number of Oracle Enterprise Manager. The default port number is 1810, but it may be configured in increments of one, up to 1816. To pinpoint it, examine the file `ORACLE_HOME/install/portlist.ini`.
2. Log in using the credentials of an OracleAS administrator.
3. From the **Standalone Instances** section of the Farm page, choose the appropriate OracleAS instance.
4. From the **System Components** list of the Application Server page, choose the single sign-on server.

Interpreting and Using the Home Page on the Standalone Console

The home page, reproduced in [Figure 10-1](#) on page 10-2, displays the following metrics in the **General** section:

- Status
A green "up" arrow signifies that the single sign-on server is running. A red "down" arrow signifies that the server is not running.
- Start Time
The start time of the database serving the single sign-on schema.
- Database
SID/instance name of the database serving the single sign-on schema.
- Database Version
Version of the database serving the single sign-on schema.

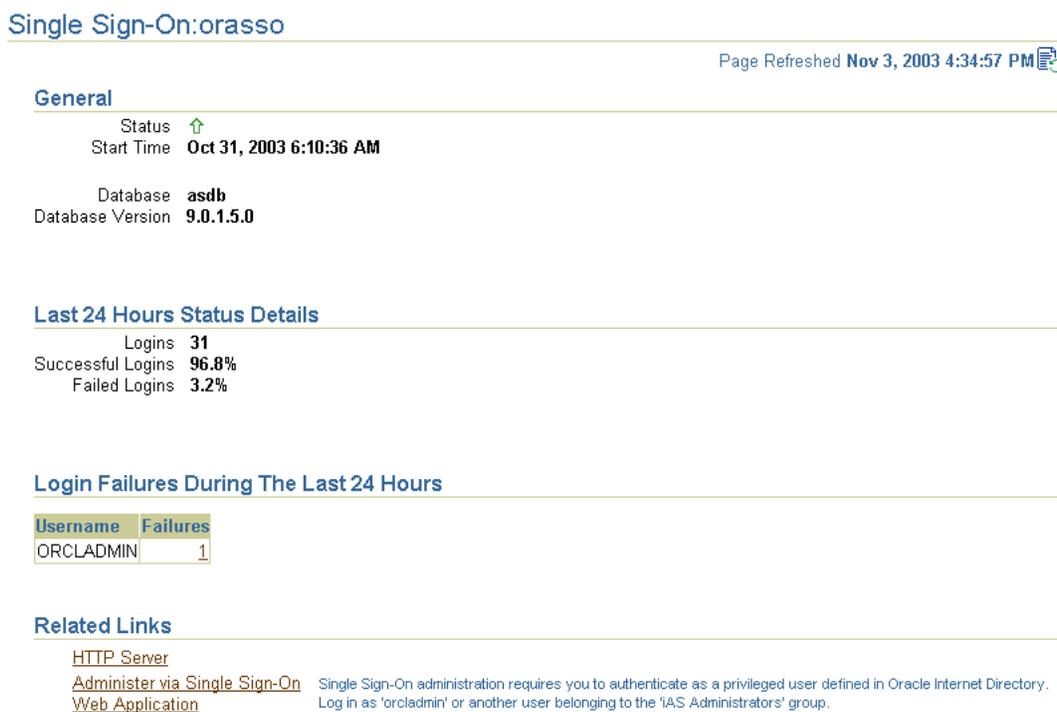
The **Last 24 Hours Status Details** section contains the following metrics:

- Logins
- Successful Logins
- Failed Logins

As the heading implies, the statistics displayed are for the previous 24 hours.

The **Login Failures During the Last 24 Hours** section enables you to determine the number of login failures that have occurred during the previous 24 hours. You choose a name from the Login Failures During the Last 24 Hours table. You then choose the associated link under the **Failures** heading. When populated, this link contains the number of login failures for the user. Clicking it takes you to the Details of Login Failures page.

Figure 10–1 Monitoring Home Page for OracleAS Single Sign-On



The **Related Links** section contains the following links:

- HTTP Server
Takes you to the monitoring home page for the Oracle HTTP Server
- Administer via Single Sign-On
Takes you to the home page for single sign-on administration

Interpreting and Using the Details of Login Failures Page

Clicking a link in the Login Failures During the Last 24 Hours table takes you to the Details of Login Failures page (Figure 10-2). This page contains a table that displays login failure times and associated IP addresses for a particular user.

Figure 10-2 Details of Login Failures Page

ORACLE Enterprise Manager 10g
Application Server Control

Logs Preferences Help

Farm > Application Server: sso123.isun6221.us.oracle.com > Single Sign-On: orasso > Details of Login Failures: ORCLADMIN

Details of Login Failures: ORCLADMIN

Page Refreshed Nov 3, 2003 5:59:39 PM

Details of Login Failures:

I.P. Address	Failure Login Time
144.25.174.159	Nov 3, 2003 2:26:51 PM

Logs | Preferences | Help

Copyright © 1996, 2003, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

Updating the Port Property for the Single Sign-On Monitoring Target

A change in the port number of the Oracle HTTP Server requires a change in the port property of the single sign-on monitoring target on that server. Perform these steps to effect the change:

1. Back up the `targets.xml` file:

```
cp ORACLE_HOME/sysman/emd/targets.xml ORACLE_HOME/sysman/emd/targets.xml.backup
```

This file is the configuration file for the various "targets" that Oracle Enterprise Manager monitors, one of which is OracleAS Single Sign-On.

2. In `targets.xml`, find the target type `oracle_sso_server`; then locate and edit the HTTP port value associated with this target type:

```
<Property NAME="HTTPPort" VALUE="7777" />
```

3. Save and close the file.
4. Reload the OracleAS Console:

```
ORACLE_HOME/bin/emctl reload
```

Note: For more information about port dependency changes, see the appendix about port numbers in *Oracle Application Server Administrator's Guide*.

Using the OracleAS Web Cache Instance to Monitor the Server

If you are using OracleAS Web Cache as a load balancer for multiple single sign-on instances, you can monitor the single sign-on server from the OracleAS Web Cache computer. At the same time, you can use the monitoring pages on any one of the single sign-on instances to monitor that instance.

Follow these steps to add a single sign-on target to the OracleAS Web Cache instance:

1. Back up the file `ORACLE_HOME/sysman/emd/targets.xml` located on the OracleAS Web Cache instance.
2. Copy the single sign-on target definition from a `targets.xml` file located on a single sign-on instance. Paste the definition to the end of the `targets.xml` file on the OracleAS Web Cache instance, inserting it just before the closing tag `</Targets>`.
3. In the `Target TYPE` tag, replace the name of the `oracle_ias` target for single sign-on instance with the name of the `oracle_ias` target for the OracleAS Web Cache instance.
4. Replace the single sign-on OracleHome value with the OracleAS Web Cache OracleHome value.
5. Change `HTTPMachine`, `HTTPPort`, and `HTTPProtocol` to values that correspond to the OracleAS Web Cache instance.
6. Run this command to incorporate the changes:

```
ORACLE_HOME/bin/emctl reload
```

Monitoring a Single Sign-On Server Enabled for SSL

If you are monitoring a single sign-on server enabled for SSL, the `HTTPProtocol` parameter in the file `ORACLE_HOME/sysman/emd/targets.xml` must be `https`. The `HTTPPort` parameter must match the SSL port of the Oracle HTTP Server for the Oracle Identity Management infrastructure. If, for example, you connect to an SSL-enabled single sign-on server with this URL:

```
http://myhost.us.oracle.com:4443/pls/orasso
```

enter 4443 for `HTTPProtocol`. Back up `targets.xml` before editing it. After editing the file, run this command to incorporate the changes:

```
ORACLE_HOME/bin/emctl reload
```

Another requirement of SSL monitoring is that the certificate configuration file for Oracle Enterprise Manager contain the certificate of the infrastructure server. To learn how to add the certificate to the configuration file, see the chapter about Oracle Enterprise Manager security in *Oracle Enterprise Manager Advanced Configuration*. See specifically the section about configuring Oracle Beacons to monitor Web applications over HTTPS.

Creating Deployment-Specific Pages

OracleAS Single Sign-On provides a framework for integrating deployment-specific login, change password, and single sign-off pages with the single sign-on server. This means that you can tailor these pages to your UI look and feel and globalization requirements, using any suitable Web technology. We recommend, however, that you use JavaServer (JSP) pages. The sample pages provided with the product have been integrated using the same framework.

This chapter contains the following topics:

- [How the Single Sign-On Server Uses Deployment-Specific Pages](#)
- [How to Write Deployment-Specific Pages](#)
- [Page Error Codes](#)
- [Adding Globalization Support](#)
- [Guidelines for Deployment-Specific Pages](#)
- [Installing Deployment-Specific Pages](#)
- [Examples of Deployment-Specific Pages](#)

How the Single Sign-On Server Uses Deployment-Specific Pages

The process that enables single sign-on pages can be summarized in a few steps:

1. The user requests a partner application and is redirected to the single sign-on server.
2. If the user is not authenticated, the single sign-on server either redirects the user to the sample login page or, if configured to use a deployment-specific page, redirects the user to this page. As part of the redirection, the server passes to the page the parameters contained in [Table 11-1](#) on page 11-3.
3. The user submits the login page, passing the parameters contained in [Table 11-2](#) on page 11-3 to the authentication URL:

```
http://sso_host:sso_port/pls/orasso/orasso.wssso_app_admin.ls_login
```

or

```
https://sso_host:sso_ssl_port/pls/orasso/orasso.wssso_app_admin.ls_login
```

At least two of these parameters, `ssusername` and `password`, appear on the page as modifiable fields.

4. If the user's password is not set to expire soon, and the single sign-on server successfully verifies the user name and password, the server redirects the user to

the success URL of the application. If authentication fails, the server redirects the user back to the login page and displays an error message.

5. If the user's password is set to expire soon, the single sign-on server presents the change password page instead of the login page. Again, if the server is configured to use a deployment-specific change password page, it redirects the user to the URL for this page, passing to the page the parameters contained in [Table 11-3](#) on page 11-4.

Note: In step 5, the same conditions apply if the directory administrator forces the user to change the password, password expiration notwithstanding.

The user submits the change password page, entering her old password, new password, and new password confirmation. The page passes the parameters contained in [Table 11-4](#) on page 11-5 to the change password URL:

```
http://sso_host:sso_port/sso/ChangePwdServlet
```

or

```
https://sso_host:sso_ssl_port/sso/ChangePwdServlet
```

If an error occurs, the single sign-on server redirects the user to the change password page and displays an error message. See "[Change Password Page Behavior](#)" in [Chapter 3](#) for a detailed discussion of conditions under which errors may occur.

If the password change is successful, the user is redirected to the partner application URL that triggered the authentication request.

6. To finish the single sign-on session, the user clicks **Logout** in the partner application he or she is working in. This act calls application logout URLs in parallel, logging the user out from all accessed applications and ending the single sign-on session.
7. The user is redirected to the single sign-on server, which presents the single sign-off page. If the server is configured to use a deployment-specific page, it redirects the user to the URL for this page, passing to the page the parameters contained in [Table 11-5](#) on page 11-5.
8. The user can click **Return** on the single sign-off page to return to the application from which logout was initiated.

Note: The change password page can be used to change a password only when the password is about to expire. The UI for Oracle Delegated Administration Services can be used for this purpose at any time. See "[Change Password Page Behavior](#)" in [Chapter 3](#) for more about this topic.

How to Write Deployment-Specific Pages

The URLs for login, change password, and single sign-off pages must accept the parameters described in the tables that follow if these pages are to function properly.

This section contains the following topics:

- [Login Page Parameters](#)

- [Forgot My Password](#)
- [Change Password Page Parameters](#)
- [Single Sign-Off Page Parameters](#)

Login Page Parameters

The URL for the login page must accept the parameters listed in [Table 11–1](#) on page 11-3.

Table 11–1 Login Page Parameters Submitted to the Page by the Single Sign-On Server

Parameter	Description
site2pstoretoken	Contains the authentication request token for login processing.
ssusername	Contains the username.
p_error_code	Contains the error code in the form of a string. Passed when an error occurs during authentication.
p_cancel_url	Contains the URL to redirect to if the user clicks Cancel —if such a button exists on the login page. This URL points to the home URL of the partner application from which login was initiated.
locale	User’s language preference (optional). Must be in ISO format. For example, French is <code>fr-fr</code> . For more about this parameter, see "Adding Globalization Support" .

The login page must pass the parameters listed in [Table 11–2](#) to the authentication URL:

```
http://sso_host:sso_port/pls/orasso/orasso.wvssso_app_admin.ls_login
```

Table 11–2 Login Page Parameters Submitted by the Page to the Single Sign-On Server

Parameter	Description
site2pstoretoken	Contains the redirect URL information for login processing.
ssusername	Contains the username. Must be UTF-8 encoded.
password	Contains the password entered by the user. Must be UTF-8 encoded.
subscribername	The subscriber nickname when realms are enabled. Must be UTF-8 encoded. Note: This field is required on the login page only when multiple realms are enabled in the single sign-on server.
locale	User’s language preference (optional). Must be in ISO format. For example, French is <code>fr-fr</code> . For more about this parameter, see "Adding Globalization Support" .
v	Contains the page version. Recommended but optional. If the parameter is passed, its value must be <code>v1.4</code> .

The login page must have at least two fields: a text field with the parameter name `ssusername` and a password field with the parameter name `password`. The values are submitted to the authentication URL. The login page must also include `site2pstoretoken` as a hidden parameter. It must submit this parameter to the login URL.

In addition to submitting these parameters, the login page is responsible for displaying appropriate error messages, as specified by `p_error_code`, redirecting to `p_cancel_url` if the user clicks **Cancel**.

Forgot My Password

When building your login page, you may want to configure it with a link that enables users to reset their passwords. This URL can go either to the home page for Oracle Delegated Administration Services or to the **Forgot My Password** link within Oracle Delegated Administration Services. Users who click the **Forgot My Password** link are challenged with a question. They must successfully answer this question before their password is reset.

Oracle Delegated Administration Services is generally available on the same computer as OracleAS Single Sign-On at a URL of the following form:

```
http://sso_host:sso_port/oiddas/
```

To learn how the **Forgot My Password** link is used to reset passwords, see the chapter about the Oracle Internet Directory Self-Service Console in *Oracle Identity Management Guide to Delegated Administration*.

Change Password Page Parameters

The URL for the change password page must accept the parameters listed in [Table 11-3](#).

Table 11-3 *Change Password Parameters Submitted to the Page*

Parameter	Description
<code>p_username</code>	Contains the user name to be displayed somewhere on the page.
<code>p_subscribername</code>	The subscriber nickname when hosting is enabled. Note: This field is required on the login page.
<code>p_error_code</code>	Contains the error code, in the form of a string, if an error occurred in the prior attempt to change the password.
<code>p_done_url</code>	Contains the URL of the appropriate page to return to after the password is saved.
<code>site2pstoretoken</code>	Contains the <code>site2pstoretoken</code> that is required by the <code>LS_LOGIN</code> routine if the password has expired or is about to expire.
<code>p_pwd_is_exp</code>	Contains the flag value indicating whether the password has expired or is about to expire. The value can be either <code>WARN</code> or <code>FORCE</code> . See Table 11-6 for the associated error codes.
<code>locale</code>	User's language preference (optional). Must be in ISO format. For example, French is <code>fr-fr</code> . For more about this parameter, see " Adding Globalization Support ".

The change password page must pass the parameters listed in [Table 11-4](#) to the change password URL:

```
http://sso_host:sso_port/sso/ChangePwdServlet
```

Table 11–4 Change Password Page Parameters Submitted by the Page

Parameter	Description
p_username	Contains the user name to be displayed somewhere on the page. Should be posted as a hidden field by the change password page. Must be UTF-8 encoded.
p_old_password	Contains the user's old password. Must be UTF-8 encoded.
p_new_password	Contains the user's new password. Must be UTF-8 encoded.
p_new_password_confirm	Contains the confirmation of the user's new password. Must be UTF-8 encoded.
p_done_url	Contains the URL of the appropriate page to return to after the password is saved.
p_pwd_is_exp	Contains the flag value indicating whether the password has expired or is about to expire. The value can be either <code>WARN</code> or <code>FORCE</code> . See Table 11–6 for the associated error codes.
site2pstoretoken	Contains the redirect URL information for login processing.
p_action	Commits changes. The values must be either <code>OK</code> (commit) or <code>CANCEL</code> (ignore).
p_subscribername	Contains the user name to be displayed somewhere on the page.
p_request	Protected URL requested by the user.
locale	User's language preference (optional). Must be in ISO format. Example: French is <code>fr-fr</code> . See " Adding Globalization Support ".

The change password page must have at least three password fields: `p_old_password`, `p_new_password`, and `p_new_password_confirm`. The page should submit these fields to the change password URL.

The page should also submit `p_done_url` as a hidden parameter to the change password URL. In addition, it should display error messages according to the value of `p_error_code`.

Single Sign-Off Page Parameters

The URL for the single sign-off page must accept the parameters listed in [Table 11–5](#).

Table 11–5 Parameters Submitted to the Single Sign-Off Page

Parameter	Description
p_app_name[1. . .n]	Contains the application name to be displayed on the page. The variable <code>n</code> stands for the number of partner applications participating in single sign-off.
p_app_logout_url[1. . .n]	Contains the application logout URL. The variable <code>n</code> stands for the number of partner applications participating in single sign-off.
p_done_url	Contains the return URL. This URL returns users to the application from which they initiated logout.
locale	User's language preference in ISO format. Sent only if the user does not pass the same value during login.

Page Error Codes

URLs for login and change password pages must accept the process errors described in the tables that follow if these pages are to function properly.

Login Page Error Codes

The login page must process the error codes listed in [Table 11–6](#).

Table 11–6 Login Page Error Codes

Value of <code>p_error_code</code>	Corresponding error
<code>acct_lock_err</code>	The user has committed too many login failures.
<code>pwd_expiry_warn_err</code>	The user's password is about to expire.
<code>pwd_exp_err</code>	The user's password has already expired.
<code>pwd_force_change_err</code>	The user must change his or her password.
<code>pwd_grace_login_err</code>	The user's password falls within the grace login period.
<code>null_uname_pwd_err</code>	The user did not type in a username.
<code>auth_fail_exception</code>	Authentication has failed.
<code>null_password_err</code>	The user did not type in a password.
<code>sso_cookie_expired_err</code>	The login cookie has expired. The user must log in again.
<code>unexpected_exception</code>	An unexpected error occurred during authentication.
<code>unexp_err</code>	Unexpected error.
<code>internal_server_err</code>	Internal server error.
<code>internal_server_try_again_err</code>	Internal server error. Try again.
<code>gito_err</code>	User's session has expired because of inactivity. User must log in again.
<code>par_login_err</code>	User must sign on again to gain access to the application.
<code>cert_auth_err</code>	Certificate sign-on has failed. User should check that the certificate is valid or should contact the administrator.
<code>userid_mismatch</code>	The user ID presented during a forced authentication does not match the user ID in the current single sign-on session.

Change Password Page Error Codes

The change password page must process the error codes listed in [Table 11–7](#).

Table 11–7 Change Password Page Error Codes

Value of <code>p_error_code</code>	Corresponding Error
<code>confirm_pwd_fail_txt</code>	The old and the new password do not match.
<code>pwd_expiry_warn_err</code>	The password is about to expire.
<code>pwd_force_change_err</code>	The password must be changed before the user can proceed.

Table 11-7 (Cont.) Change Password Page Error Codes

Value of <code>p_error_code</code>	Corresponding Error
<code>pwd_grace_login_err</code>	The password has expired, but a grace login is permitted.
<code>account_deactivated_err</code>	The user account is disabled.
<code>act_lock_err</code>	The user account is locked.
<code>pwd_illegal_value</code>	The password contains an illegal value.
<code>pwd_in_history</code>	The password is in the password history.
<code>pwd_min_length</code>	The password does not meet the minimum length requirement.
<code>pwd_numeric</code>	The password does not meet the numeric character requirement.

Adding Globalization Support

The OracleAS Single Sign-On framework enables you to globalize deployment-specific pages to fit the needs of your deployment. When deciding what language to display the page in, you can adopt different strategies. Two are presented here.

Deciding What Language to Display the Page In

This section explains how to use either the HTTP Accept-Language header or deployment page logic to choose a language to display.

Use the Accept-Language Header to Determine the Page

Browsers enable end users to decide the language (locale) they would like to view their Web content in. The browser sends the language that the user chooses to the server in the form of the HTTP Accept-Language header. The logic of the deployment-specific page must examine this header and render the page accordingly. When it receives this page, the single sign-on server takes note of the header value for Accept-Language and sends it to partner applications when it propagates the user's identity. Note that, although many partner applications enable users to override this header, the single sign-off page appears in the language established at sign-on. The net effect is a consistent session language for all partner applications.

The Accept-Language header is the preferred mechanism for determining the language preference. A major benefit of this approach is that end users have typically already set their language preference while browsing other Web sites. The result is browsing consistency between these pages and single sign-on pages.

Use Page Logic to Determine the Language

Although Oracle recommends the approach described in the preceding section, you may choose to implement globalization based on mechanisms that extend or override the language preference set in the browser. You may, for instance, do one of the following:

- Display a list of languages on the login page and allow the user to select from this list. As a convenience to the user, you can make this selection persistent by setting a persistent cookie.
- Render the page in one, fixed language. This method is appropriate when you know that the user population is monolingual.

- Obtain language preferences from a centralized application repository or a directory. A centralized store for user and system preferences and configuration data is ideal for storing language preferences.

If you use page logic to set language preferences, the page must propagate this information to the single sign-on server. The server must propagate this information to partner applications. The net result is a consistent globalization experience for the user. Your page must pass the language in ISO-639 format, using the `locale` parameter (Table 11–2) in the login form. A number of sites contain a full list of ISO-639 two-letter language codes. Here is one of them:

<http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt>

Here is a site that contains a full list of ISO-3166 two-letter country codes:

http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html

Note: In the event that the `locale` parameter is passed to the single sign-on server (Table 11–1), the parameter value is sent to `mod_osso`. `mod_osso` prefixes this value to the HTTP Accept-Language header before passing the header to partner applications.

Rendering the Page

Once it determines the end-user's locale, the deployment-specific page must use the corresponding translation strings to render the page. To learn how to store and retrieve these strings, see the chapter about locale awareness in *Oracle Application Server Globalization Guide*. You may also want to consult standard documents about Java development. Here are two links:

- Java Internationalization Guide:
<http://java.sun.com/j2se/1.4.2/docs/guide/intl/index.html>
- General link for Java documentation:
<http://java.sun.com/j2se/1.4.2/docs>

Guidelines for Deployment-Specific Pages

When implementing deployment-specific pages, observe the following guidelines:

- Oracle recommends that login and change password pages be protected by SSL.
- The login and change password pages must code against cross-site scripting attacks.
- The login and change password pages must have auto-fill and caching set to `off`. This prevents user credentials from being saved or cached in the browser. Here is an example of the `AutoComplete` tag:

```
<FORM NAME="foo" AutoComplete="off" METHOD="POST" ACTION="bar">
```

- Oracle recommends that you configure your login page to display a banner that warns against unauthorized access. You may, for example, want to use the following text or a variant thereof:

```
Unauthorized use of this site is prohibited and may subject you to civil and  
criminal prosecution.
```

- Deploy the login and change password pages on the computer that hosts the single sign-on server. This makes it easier to detect false versions of these pages.

Installing Deployment-Specific Pages

Use the `policy.properties` file to install deployment-specific login and change password pages. Use the `WSSO_LS_CONFIGURATION_INFO$` table to install the deployment-specific single sign-off page.

Using `policy.properties` to Install Login and Change Password Pages

To install your own login and change password pages, edit the following parameters in `ORACLE_HOME/sso/conf/policy.properties`:

```
#Custom login page link
loginPageUrl = login_page_URL

#Custom change password page link
chgPasswordPageUrl = change_password_page_URL
```

Finally, restart the single sign-on server:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Using `policy.properties` to Install Wireless Login and Change Password Pages

OracleAS Wireless has its own framework for integrating deployment-specific wireless login and change password pages. The procedure for installing these pages is similar to that used to install standard pages (section immediately preceding). First, go to `ORACLE_HOME/sso/conf/policy.properties`; then edit (add) the following parameters:

```
#Wireless login page link
wirelessLoginPageUrl = wireless_login_page_url
wirelessChgPasswordPageUrl = change_password_page_URL
```

Finally, restart the single sign-on server:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Using `WSSO_LS_CONFIGURATION$` to Install the Single Sign-Off Page

The `WSSO_LS_CONFIGURATION_INFO$` table in the single sign-on schema contains the `LOGIN_URL` column. Use this column to enable the single sign-off page.

`LOGIN_URL` contains three values separated by a space. The first two values are reserved for backward compatibility. Do not edit these values. The third value specifies the single sign-off page. If you are installing your own single sign-off page, you must modify this last value:

1. On the database where the single sign-on server is installed, log in to the single sign-on schema using SQL*Plus:

```
ORACLE_HOME/bin/sqlplus orasso/password
```

See [Appendix B](#) to learn how to obtain the password for the single sign-on schema.

2. Update LOGIN_URL. To replace the sample page with your own page, update the third value in the column. In the following example, `single_signoff.jsp` is the deployment-specific page.

```
UPDATE WSSO_LS_CONFIGURATION_INFO$  
SET LOGIN_URL='UNUSED UNUSED http://server.domain[:port]/single_signoff.jsp';  
COMMIT;
```

3. To revert to the Oracle page, simply restore the original value:

```
UPDATE WSSO_LS_CONFIGURATION_INFO$  
SET LOGIN_URL='UNUSED UNUSED UNUSED';  
COMMIT;
```

Note: The first two values must be UNUSED.

Examples of Deployment-Specific Pages

The `ipassample.jar` file contains the files `login-ex.jsp`, `password-ex.jsp`, and `signoff-ex.jsp`. You may customize these to suit your deployment. If you want to use these files, see ["Obtaining the Sample Files"](#) in [Chapter 2](#).

Integrating with Third-Party Access Management Systems

This chapter explains how to integrate OracleAS Single Sign-On with third-party access management products. It describes how third-party integration works; then it presents the integration APIs. Finally, it provides an example of how to integrate OracleAS Single Sign-On with a third-party access management system.

An enterprise that has a third-party system in place can gain access to the OracleAS suite by building an authentication adapter that enables the OracleAS Single Sign-On server to act as an authentication gateway between the third-party system and Oracle applications.

The chapter contains the following topics:

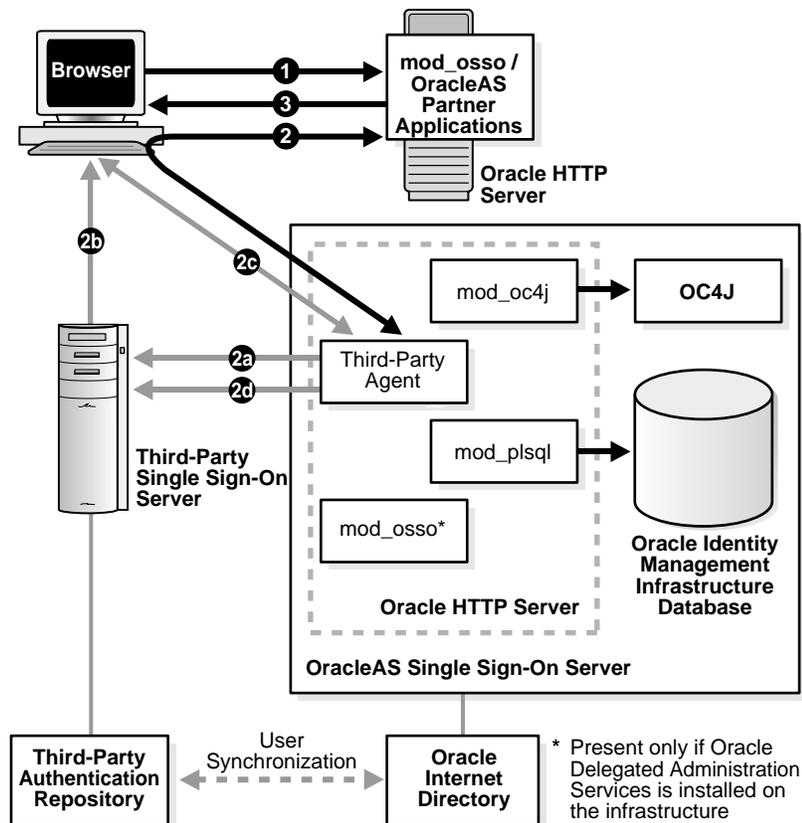
- [How Third-Party Access Management Works](#)
- [Synchronizing the Third-Party Repository with Oracle Internet Directory](#)
- [Third-Party Integration Modules](#)
- [Integration Case Study: SSOAcme](#)

How Third-Party Access Management Works

In third-party access management, the OracleAS Single Sign-On server, the third-party access management server, and the partner application form a chain of trust. The OracleAS Single Sign-On server delegates authentication to the third-party access management server, becoming a single sign-on-enabled application itself. Oracle applications continue to work only with the OracleAS Single Sign-On server and are unaware of the third-party access management server. Implicitly, however, they trust the third-party server.

For OracleAS Single Sign-On to issue users an authentication token under this arrangement, the third-party access management server must pass the OracleAS single sign-on server the user's identity by setting HTTP headers or by using some other mechanism. Once it obtains the user's identity, the OracleAS Single Sign-On server functions as before, authenticating and redirecting users to its partner applications. [Figure 12-1](#) on page 12-2 illustrates the process.

Figure 12–1 Accessing Oracle Partner Applications Using a Third-Party Server



The illustration captures two possible scenarios:

Scenario 1: The user has not yet authenticated to the third-party server

1. The unauthenticated user attempts to access an application protected by OracleAS Single Sign-On.
2. The application redirects the user to the OracleAS Single Sign-On server for authentication. As part of this redirection, the following occurs:
 - a. The OracleAS Single Sign-On server has the third-party agent, typically a module on the Oracle HTTP Server, authenticate the user. You install this module when you install the third-party adapter. Note that this module resides only on the single sign-on server. It does not reside on the application server.

Note: See ["Integration Case Study: SSOAcme"](#) for more about the third-party agent.

 - b. The third-party server sets a token in the user's browser.
 - c. The OracleAS Single Sign-On server retrieves the token from the browser.
 - d. The OracleAS Single Sign-On server verifies the token with the third-party server.

After token verification, the OracleAS Single Sign-On server returns the user—and the authenticated user information—to the requested application.

3. The application provides content to the user.

Scenario 2: The user has already authenticated to the third-party server

1. The authenticated user attempts to access an application protected by OracleAS Single Sign-On.
2. The application redirects the user to the OracleAS Single Sign-On server for authentication. As part of this redirection, the following occurs:
 - a. The OracleAS Single Sign-On server retrieves the token from the browser (step 2c in scenario 1).
 - b. The OracleAS Single Sign-On server verifies the token with the third-party server (step 2d in scenario 1).

After token verification, the OracleAS Single Sign-On server returns the user to the requested application.

3. The application provides content to the user.

Note: If the single sign-on systems involved are to be accessible to all authorized users, the user repository must be centralized in one place. This means that, before deployment, users must be synchronized between Oracle Internet Directory and the external repository if this repository is not also Oracle Internet Directory.

Synchronizing the Third-Party Repository with Oracle Internet Directory

The authentication scenario presented in the preceding steps assumes either that the user repository is Oracle Internet Directory or that the repository is a third-party directory or database. If the repository is the latter, the user name information must be synchronized with the user entry in Oracle Internet Directory. This synchronization enables the single sign-on server to retrieve the user attributes required by applications enabled for single sign-on.

Note: Third-party access management integration cannot proceed if the synchronization mechanism is not in place.

To synchronize the third-party repository with Oracle Internet Directory, use either Oracle Directory Integration and Provisioning or bulk load tools. For details, see the chapters about directory synchronization in *Oracle Identity Management Integration Guide*.

Third-Party Integration Modules

There are two ways to achieve third-party integration: you can use existing, vendor-supplied packages, or you can build your own third-party adapter, using interfaces provided by Oracle.

Using Vendor-Supplied Packages

Several third-party access management vendors provide authentication adapters for the OracleAS Single Sign-On server. These products enable you to integrate a third-party system with the Oracle system without having to write your own code. The link that follows provides information about these vendors' products. All of the vendors listed certify that their products work with OracleAS Single Sign-On. See the section **Single Sign-On** under the heading **Documentation**, which appears near the bottom of the page. Note that this material cannot be found under the **Services** menu.

http://www.oracle.com/technology/products/id_mgmt/partners/index.html

If you decide to use a third-party adapter, be aware that the instructions for implementing the adapter are contained not in this document, but in vendor documents. If you need help implementing the adapter, you must contact the vendor directly. The link provided includes contact information.

Building Your Own Package

You can use the Java tool kit `oracle.security.sso.ias904`, provided later in this chapter, to build your own third-party integration adapter. The tool kit consists of two interfaces, one for performing authentication, the other for setting deployment-specific cookies. The OracleAS single sign-on server invokes the first during authentication. It invokes the second, the cookie adapter, after it has successfully determined the user's identity.

This section contains the following topics:

- [Guidelines for Using the Interfaces](#)
- [The Interfaces](#)
- [Configuration Steps](#)

Guidelines for Using the Interfaces

The authentication interface is required only if you are building your own third-party integration adapter. The OracleAS single sign-on server uses the interface implementation to determine the user's identity during authentication. The cookie interface is provided for your convenience. Use it to set additional cookies once the user is authenticated. You may, for example, want to use this feature to set user preferences. You do not need the cookie interface to achieve third-party integration. If you choose to use it, you can include it in your own adapter or in one provided by a vendor.

The Interfaces

The two interfaces in the kit perform the following functions:

- [Authentication Using a Token](#)
- [Set External Cookies](#)

Authentication Using a Token The `IPASAuthInterface.java` package is invoked by the OracleAS Single Sign-On server during authentication. If authentication using a token is to be supported, the implementer of this interface must return the user name to the OracleAS Single Sign-On server by retrieving the user identity in a secure fashion—from a securely set HTTP header, for instance, or from a secure cookie. Here is the interface:

```

/**
 * returns IPASUserInfo
 * The returned object should contain user information
 * nicknames
 *
 * @param request The user's HTTP request object
 *
 * @throws IPASAuthException if the authentication fails for whatever
 * reasons.
 * The exception message will be propagated to the login page
 * directly.
 *
 * @throws IPASInsufficientCredException if all the required
 * credentials
 * @return IPASUserInfo authenticated user information
 */

public IPASUserInfo authenticate(HttpServletRequest request)
throws IPASAuthException, IPASInsufficientCredException;

```

Set External Cookies The `IPASCustomCookieInterface.java` package enables you to set optional, deployment-specific cookies. These cookies are set only if authentication is successful and the cookie adapter corresponds to the appropriate authentication level.

```

/**
 * A custom cookie can be implemented using this interface.
 * SSO server sends the cookie to the user browser.
 *
 * @param user user object that contains the authenticated user
 * information
 *
 * @param req HTTP user request object
 *
 * @return array of Cookie objects
 */

public Cookie[] getCustomCookie(IPASUserInfo user, HttpServletRequest req);

```

Configuration Steps

To create your own third-party integration adapter using the interfaces, follow these steps:

1. In the third-party access management system, create rules that protect these URIs:

```

/pls/orasso/orasso.wssso_app_admin.ls_login
/sso/auth/*

```

2. Register this logout URI with the third-party access management system:

```

/pls/orasso/orasso.wssso_app_admin.ls_logout

```

Note: In the remaining steps and in the sample package that follows, change all instances of `acme` and `SSOAcme` to your company name.

3. Create a Java file for your package. For help, see the sample file in "[Integration Case Study: SSOAcme](#)". The sample file is called `SSOAcmeAuth.java`. Before it is compiled, this package directive must be added to it:

```

package acme.security.ssoplugin;

```

4. Compile the file, including `ORACLE_HOME/sso/lib/ipastoolkit.jar` in the class path. The sample file `SSOAcmeAuth.java` is compiled this way:

```
ORACLE_HOME/jdk/bin/javac -classpath ORACLE_HOME/sso/lib/ipastoolkit.jar:  
ORACLE_HOME/lib/servlet.jar -d ORACLE_HOME/sso/plugin SSOAcmeAuth.java
```

This command creates `SSOAcmeAuth.class` and places it in the directory `ORACLE_HOME/sso/plugin/acme/security/ssoplugin`.

5. In the `policy.properties` file, replace the simple authentication plugin with the plugin that you created in step 3. The simple authentication plugin looks like this:

```
MediumSecurity_AuthPlugin = oracle.security.sso.server.auth.SSOAcmeAuth
```

The sample plugin looks like this:

```
MediumSecurity_AuthPlugin = acme.security.ssoplugin.SSOAcmeAuth
```

Note: When editing `policy.properties`, take care not to insert blank space at the end of a line.

6. If you decided to implement the custom cookie interface, add the class name of the implementation to `policy.properties`:

```
# Custom Cookie Provider Class name  
# -----  
# Sample custom cookie tester provider class
```

```
CustomCookie_ProviderPlugin = class_name
```

7. Designate the minimum authentication level at which the custom cookie is set if the user authenticates successfully:

```
# Custom Cookie auth level  
# -----  
CustomCookieAuthLevel = authentication_level
```

If you are not using multilevel authentication and are using default settings for authentication adapter levels, you can set this value to:

```
CustomCookieAuthLevel = MediumSecurity
```

8. Restart the single sign-on middle tier:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server  
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

9. Test the integrated system.

Logging Out of the Integrated System

Before proceeding with this section, make sure that you understand the concept of single sign-off and know how to implement it. Read the section "[Single Sign-Off](#)" in [Chapter 1](#).

Third-party logout takes two forms:

- The user initiates a logout request using the third-party access management system.

In this scenario, the user clicks a logout link that invokes a logout handler in the third-party system. The third-party logout flow cleans up its own session. After cleanup, The third-party system must invoke the OracleAS Single Sign-On logout handler. Invoking the OracleAS Single Sign-On logout handler ensures that the user is logged out of all applications protected by the OracleAS Single Sign-On server. To perform single sign-on logout, the third-party system must redirect the user to the following URL:

```
http://single_sign-on_host:single_sign-on_port/pls/orasso/orasso.wvso_app_admin.ls_logout?p_done_url=done_url
```

or, if SSL is enabled, to this URL:

```
https://single_sign-on_host:single_sign-on_ssl_port/pls/orasso/orasso.wvso_app_admin.ls_logout?p_done_url=done_url
```

done_url is the URL to which the user is redirected after logout.

- The user initiates a logout request using the OracleAS Single Sign-On system

In this scenario, the user clicks a logout link in an Oracle partner application. This invokes the OracleAS Single Sign-On logout handler. When logout is finished, the user should also be logged out from the third-party system. Concurrent logout is effected by registering the Oracle logout handler (*ls_logout* in the URL immediately preceding) with the third-party system. The third-party system cleans up the third-party session when it detects that the Oracle logout handler is being invoked.

If you have chosen to use an adapter from a third party, see that vendor's documentation to learn how logout is actually implemented.

Integration Case Study: SSOAcme

Consider the case of SSOAcme, a product that offers single sign-on authentication to protected resources. SSOAcme consists of two components: the SSOAcme policy server and the SSOAcme agent. The first provides users with a variety of services including user and session management, authentication, and authorization. The second is located on Web servers and Web application servers. It screens requests for resources and determines whether a resource is protected by SSOAcme.

Customers who have SSOAcme already installed may want to use it to gain access to OracleAS applications. They can achieve this access by using APIs that enable SSOAcme to talk to Oracle applications by way of OracleAS Single Sign-On.

Note: SSOAcme is a fictitious product, used for the purposes of illustration only.

This section contains the following topics:

- [Sample Integration Package](#)
- [Migrating the Release 9.0.2 Sample Implementation to Release 10.1.2](#)

Sample Integration Package

The `SSOAcme.java` package, presented here, can be used to integrate an existing SSOAcme implementation with OracleAS Single Sign-On.

```
/**
 * returns IPASUserInfo
 **/

/* Copyright (c) 2002, 2003, Oracle Corporation. All rights reserved. */

/*
DESCRIPTION
    Sample class for SSOAcme integration with SSO Server

PRIVATE CLASSES

NOTES
    This class implements the SSOAuthInterface.
    To enable this integration, replace:
        oracle.security.sso.server.auth.SSOAuth
    with
        acme.security.sso.plugin.SSOAcmeAuth
    for the desired security level in policy.properties.
*/

import java.io.PrintWriter;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import oracle.security.sso.ias904.toolkit.IPASAuthInterface;
import oracle.security.sso.ias904.toolkit.IPASAuthException;
import oracle.security.sso.ias904.toolkit.IPASUserInfo;
import oracle.security.sso.ias904.toolkit.IPASInsufficientCredException;

public class SSOAcmeAuth implements IPASAuthInterface {

    private static String CLASS_NAME = "SSOAcmeAuth";
    private static String ACME_USER_HEADER = "ACME_USER";

    public SSOAcmeAuth() {

    }

    public IPASUserInfo authenticate(HttpServletRequest request)
        throws IPASAuthException, IPASInsufficientCredException {

        String AcmeUserName = null;

        try
        {
            AcmeUserName = request.getHeader(ACME_USER_HEADER);
        }
        catch (Exception e)
        {
            throw new IPASInsufficientCredException("No Acme Header");
        }

        if (AcmeUserName == null)
            throw new IPASInsufficientCredException("No Acme Header");

        IPASUserInfo authUser = new IPASUserInfo(AcmeUserName);

        return authUser;
    }
}
```

```

    }

    public String getUserCredentialPage(HttpServletRequest request,
        String msg) {

        // This function will never have been reached in the case of SSOAcme
        // because the SSOAcme Agent will intercept all requests
        return "http://error_url";

    }

}

```

Migrating the Release 9.0.2 Sample Implementation to Release 10.1.2

This section is provided for the benefit of those who used the release 9.0.2 external authentication package to perform third-party authentication. The release 9.0.2 package was written in PL/SQL. The release 10.1.2 package is written in Java. In the lines that follow, the pertinent sections of the two packages appear together.

New Authentication Interface

Release 10.1.2:

```

package acme.security.ssoplugin;

import java.io.PrintWriter;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import oracle.security.sso.server.util.SSODebug;
import oracle.security.sso.ias904.toolkit.IPASAuthInterface;
import oracle.security.sso.ias904.toolkit.IPASAuthException;
import oracle.security.sso.ias904.toolkit.IPASUserInfo;
import oracle.security.sso.ias904.toolkit.IPASInsufficientCredException;

public class SSOAcmeAuth implements IPASAuthInterface {

    private static String CLASS_NAME = "SSOAcmeAuth";
    private static String ACME_USER_HEADER = "ACME_USER";

    public SSOAcmeAuth() {
    }

    public IPASUserInfo authenticate(HttpServletRequest request)
    throws IPASAuthException, IPASInsufficientCredException {

```

Release 9.0.2:

```

FUNCTION authenticate_user
(
    p_user OUT VARCHAR2
)
return PLS_INTEGER
IS
    l_http_header varchar(1000);
    l_ssouser wwsec_person.user_name%type := NULL;
BEGIN

```

Get User Name from HTTP Header

Release 10.1.2:

```
String AcmeUserName = null;

try
{
    AcmeUserName = request.getHeader(ACME_USER_HEADER)
```

Release 9.0.2:

```
l_http_header := owa_util.get_cgi_env('HTTP_Acme_USER');
debug_print('Acme ID : ' || l_http_header);
```

Error Handling if User Name Not Present

Release 10.1.2:

```
}
catch (Exception e)
{
    DebugUtil.debug(SSODebug.ERROR, "exception: " + CLASS_NAME, e);
    throw new IPASInsufficientCredException("No Acme Header");
}

if (AcmeUserName == null)
throw new IPASInsufficientCredException("No Acme Header");
```

Release 9.0.2:

```
IF ( (l_ssouser IS NULL) or
    ( INSTR(l_ssouser, GLOBAL_SEPARATOR) != 0) ) THEN
    debug_print('malformed user id: '
        || l_ssouser
        || ' returned by wwsso_auth_external.authenticate_user');
    RAISE EXT_AUTH_FAILURE_EXCEPTION;
ELSE
```

Return User Name to Single Sign-On Server

Release 10.1.2:

```
IPASUserInfo authUser = new IPASUserInfo(AcmeUserName);

return authUser;

}
```

Release 9.0.2:

```
p_user := NLS_UPPER(l_ssouser);    -- p-user is the out parameter
return 0;                          -- SUCCESS error code
END IF;
```

Exporting and Importing Data

This chapter explains how to move data between two or more single sign-on servers. Various conditions dictate whether you export and import data. Perhaps you want to stage data on a test server before transferring it to a production server. Or maybe you want to consolidate multiple servers as one server. Or you may simply want to back up an existing server.

The chapter contains the following topics:

- [What's Exported and Imported?](#)
- [Export and Import Script: Syntax and Parameters](#)
- [Exporting Data from One Server to Another](#)
- [Consolidating Multiple Servers](#)
- [Verifying That Export and Import Succeeded](#)
- [Error Messages](#)

What's Exported and Imported?

The export and import script, `ssomig`, moves three categories of data:

- Definitions and user data for external applications
- Registration URLs and tokens for partner applications
- Connection information used by OracleAS Discoverer to access various data sources

If you need to move user accounts, use LDAP command-line scripts such as `ldapsearch` to extract data from the source directory. Use `ldapadd` or `ldapmodify` to load data into the target directory. To learn how to use these scripts, see the syntax chapter in *Oracle Identity Management Application Developer's Guide*.

Export and Import Script: Syntax and Parameters

The `ssomig` script uses Perl, Oracle SQL*Plus, and the database export and import tools `exp` and `imp` to move data between two release 10.1.2 servers. You must run the export and import modes separately. You can find `ssomig` at `ORACLE_HOME/sso/bin`.

Script Syntax

Use this syntax to run `ssomig`:

```

ssomig -s sso_schema
      -p sso_password
      -c net_service_name
      -log_d log_dir
      {
        -export [-prompt]
              [-noextappusrs]

        -import {-merge | -overwrite}
              [-discoforce | -disconoforce]
      }
      [-log_f log_file]
      [-d dump_file_name]
      [-help]

```

Script Parameters

Table 13–1 defines the parameters passed to `ssomig`.

Table 13–1 Parameters Passed to `ssomig`

Parameter	Description	Additional Information
-s	Database schema name for OracleAS Single Sign-On.	The default is ORASSO.
-p	Database schema password for OracleAS Single Sign-On.	The password is randomized during installation of the OracleAS infrastructure. To obtain the password, see Appendix B .
-c	Net service name for the OracleAS Single Sign-On database.	-
-log_d	Name of the log directory.	This directory must be writable. The log file, the export configuration file, and the dump file are written here. Use the absolute path for the directory when running the script. The default is <code>ORACLE_HOME/sso/log</code> .
-export	Extracts data from single sign-on tables and places it into a dump file.	-
-prompt	Exports partner and external applications selectively.	Use with <code>export</code> .
-noextappusrs	Specifies that external application users not be exported.	Use with <code>export</code> . Choose this mode if you are moving data from a staged server to a production server and do not want to move test users.
-import	Extracts data from a dump file and places it into single sign-on tables.	-
-merge	Imports only partner and external applications that do not already exist in the target server.	Choose this mode after you have imported the first of multiple servers. Use with <code>import</code> .
-overwrite	Imports all partner and external applications, regardless of whether some already exist in the target server.	Choose this mode when migrating the first of multiple servers. Use with <code>import</code> .
-discoforce	Imports OracleAS Discoverer information, replacing Discoverer information in the target server.	-

Table 13–1 (Cont.) Parameters Passed to *ssomig*

Parameter	Description	Additional Information
-disconoforce	Imports OracleAS Discoverer information only if the target server contains no Discoverer data.	-
-log_f	Log file name.	This file provides export results and the runtime status of tools such as SQL*Plus, exp, and imp. The default file name is <code>ssomig.log</code> .
-d	Dump file name.	The default is <code>ssomig.dmp</code> .
-help	Describes the syntax and parameters for <code>ssomig</code> .	-

Exporting Data from One Server to Another

The scenarios under which the export and import script is run fall into two categories: export from a single server and export from multiple servers. The choice of one category or the other dictates whether the script is run in overwrite mode or merge mode. It also dictates whether partner and external applications are exported selectively. This section examines single-server export and import. For multiple-server export and import, see "[Consolidating Multiple Servers](#)".

This section contains the following topics:

- [Export and Import Scenarios and Script Examples](#)
- [Running the Script](#)

Export and Import Scenarios and Script Examples

What follows are scenarios that you are likely to encounter when moving data from one single sign-on server to another. The command appropriate for each scenario is provided.

Note: The following examples are described with UNIX in mind, but they work with Windows as well. Simply substitute a backslash for the forward slash in the log directory path.

Export Scenarios

- Export all partner and external applications. Export OracleAS Discoverer data entirely. This command is appropriate when you want to back up a server:


```
ssomig -export -s orasso -p password -c net_service_name -log_d /tmp
```
- Selectively export partner and external applications. Export OracleAS Discoverer data entirely. Run this command when you want to move staged data to a production server:


```
ssomig -export -prompt -s orasso -p password -c net_service_name -log_d /tmp
```
- Selectively export partner applications. Selectively export definitions for external applications. Do not export user data for external applications. Export OracleAS Discoverer data entirely. Run this command when you want to move staged data to a production server, but do not want to move external application information for test users:

```
ssomig -export -prompt -noextappusrs -s orasso -p password -c net_service_name  
-log_d /tmp
```

Import Scenarios

- Import partner and external applications. Overwrite only entries that are the same as the entries that you are importing. Exclude OracleAS Discoverer data. This command is useful if you are not deploying Discoverer:

```
ssomig -import -overwrite -s orasso -p password -c net_service_name -log_d  
/tmp
```

- Import partner and external applications and OracleAS Discoverer data. Overwrite all entries, regardless of whether they are the same as the entries you are importing. Run this command if you need to refresh data in the target server:

```
ssomig -import -overwrite -s orasso -p password -c net_service_name -log_d  
/tmp -discoforce
```

- Import partner and external applications. Overwrite all entries, regardless of whether they are the same as the entries you are importing. Import OracleAS Discoverer information only if none is present in the target server:

```
ssomig -import -overwrite -s orasso -p password -c net_service_name -log_d  
/tmp -disconoforce
```

Running the Script

To export data:

1. Log in to the computer that you are exporting from.
2. Set the Oracle home environment variable, `ORACLE_HOME`, to point to the Oracle home of the release 10.1.2 single sign-on server.
3. Run the script. (See ["Export and Import Scenarios and Script Examples"](#).)

This action creates the dump file `ssomig.dmp`, the log file `ssoconf.log`, and the single sign-on configuration file `ssoconf.log`. All three are created in the log directory.

Note: When you run `ssomig` in export mode with the prompt option, the script asks you to identify applications that you do not want to export. At the same time, it asks you to press any key when you are finished making your selections. Press the **Return** or **Enter** key instead. The script ignores other keys.

To import data:

1. Log in to the computer that you are importing data to.
2. Set the environment variable `ORACLE_HOME` to point to the Oracle home for the release 10.1.2 single sign-on server.
3. Make sure that the `log_d` parameter points to the log directory where the log files for export are located. The script must reference the files `ssomig.dmp` and `ssoconf.log` when it runs in import mode. You may have to copy these files from the computer on which the export server is located.
4. Run the script, choosing `import` mode. (See ["Export and Import Scenarios and Script Examples"](#)).

Verifying That Export and Import Succeeded

After completing export and import operations, open `ssomig.log` and check for errors. To interpret the messages that you encounter in the file, see ["Error Messages"](#).

Consolidating Multiple Servers

This scenario is applicable if several departments in your enterprise maintain departmental single sign-on servers. You may want to consolidate these servers into a unified identity management service.

Use the following approach to export and import multiple servers:

1. Export data from all of the servers involved except the target server. To learn how to run the script, see ["Exporting Data from One Server to Another"](#).
2. Run the script in `import` mode, `overwrite` option, for the first single sign-on server that you migrate. For help, see the section ["Import Scenarios"](#).
3. For subsequent servers, run the script in `merge` mode. Import partner and external applications to the target server, importing the servers one at a time:

```
ssomig -import -merge -s orasso -p password -c net_service_name -log_d /tmp -d
ssomig.dmp
```

This command merges only partner and external applications.

Note: when importing multiple servers, you can run the script in `overwrite` mode to cancel the result of a previous run.

Error Messages

Any one of the following messages may appear during the course of export and import. [Table 13–2](#) defines these messages to aid problem resolution.

Table 13–2 Error Codes for Export and Import

Error	Cause	Action
SSO-80000: The operation was unsuccessful.	Import or export or both failed because of one or more errors.	Determine the error from the log file or from screen output.
SSO-80001: The environment variable <code>ORACLE_HOME</code> is not set.	The variable has not been set for the release 10.1.2 Oracle home.	Follow the instructions in "Running the Script" .
SSO-80002: Invalid <code>ORACLE_HOME</code> specified.	The directory represented by <code>ORACLE_HOME</code> does not exist or required scripts under it are unavailable.	Set the Oracle home to a valid Oracle instance.
SSO-80004: Invalid log directory. String is not writable.	You lack write permission for the log directory specified.	Specify a directory for which you have write permission.
SSO-80005: Invalid log directory. String is not directory.	The log directory specified does not exist.	Specify a valid directory.

Table 13–2 (Cont.) Error Codes for Export and Import

Error	Cause	Action
SSO-80008: Duplicate option string.	The command-line parameter string is repeated or both options that compose a set of complementary options are provided.	Avoid repeating the command-line parameter string. Avoid including both options that compose a set of complementary options— <code>export</code> and <code>import</code> , for instance.
SSO-80009: Mandatory parameter missing: string.	A mandatory command-line parameter string is missing	Specify the parameter string, including any relevant values.
SSO-80010: Invalid SSO Server version detected.	The script does not support the version of the source or destination server.	Make sure that you are using release 10.1.2 servers to perform export and import operations.
SSO-80011: Invalid option string.	The parameter string is not a recognized command-line parameter	Use the option <code>help</code> to obtain a list of valid parameters.
SSO-80012: Invalid SSO schema information.	The schema name, password, or net service name is invalid.	Reenter the command.
SSO-80014: Invalid log file. String is not writable.	You lack write permission for the log file that you specified.	Specify a log file for which you have write permission.
SSO-80015: Failed to drop temporary tables.	An expected script file was missing, or an operating system error or database error was encountered.	View the log files for details. Correct any errors that you find.
SSO-80050: Data export unsuccessful.	The export operation failed because of one or more errors.	Determine the error from the log file or from screen output.
SSO-80051: Copying data into the temporary tables failed.	A script file is missing or an operating system error or database error was encountered.	View the log file for details. Correct errors that you find.
SSO-80052: Invalid dump file. String not writable.	You lack write permission for the dump file specified.	Specify a dump file for which you have write permission.
SSO-80076: Cannot determine NLS information.	A script file is missing or an operating system error or database error was encountered.	View the log file for details. Correct errors that you find.
SSO-80077: The file string does not exist.	The file string has been deleted or renamed externally.	Ensure that the file string is not touched externally during execution of the script.
SSO-80078: Creating the table that represents the config file failed.	A script file is missing or an operating system error or database error was encountered.	View the log file for details. Correct errors that you find.
SSO-80100: Data import unsuccessful.	The import operation failed because of one or more errors.	Determine the error from the log file or from screen output. Correct errors that you find.
SSO-80101: Cannot read the import dump file: string.	You lack read permission for the dump file string.	Obtain read permission for the specified dump file.
SSO-80102: The dump file string is of size zero.	An error occurred during export.	View the log file. Correct errors that you find.

Table 13-2 (Cont.) Error Codes for Export and Import

Error	Cause	Action
SSO-80103: Config file not found: string.	This error appears if required configuration files such as dump and log are missing during import.	Ensure that the configuration files are present in the log directory.
SSO-80104: Corrupted or invalid config file.	The configuration file has been altered.	Ensure that the configuration file is not altered when transferred from the source to the destination.
SSO-80150: Package loading into the SSO schema failed.	A script file is missing or an operating system error or database error was encountered.	View the log file for details. Correct errors that you find.

Troubleshooting

This appendix contains the following topics:

- [Log Files](#)
- [Error Messages and Other Problems](#)
- [Increasing the Debug Level](#)
- [Enabling the Debug Option in the Single Sign-On Database](#)
- [Enabling LDAP Tracing for UI Operations](#)
- [Managing Single Sign-On Audit Records](#)
- [Refreshing the LDAP Connection Cache](#)
- [Restarting OC4J After Modifying Oracle Internet Directory](#)
- [Troubleshooting Replication](#)
- [A Word About Non-GET Authentication](#)
- [Need More Help?](#)

Log Files

These OracleAS log files record data about single sign-on operations.

- General log file for the single sign-on server:

`ORACLE_HOME/sso/log/ssoServer.log`

Usage Notes:

The single sign-on server writes all errors to this file. You can change the default location by editing `ORACLE_HOME/sso/conf/policy.properties`. You can also use this file to change the logging level.

- Startup error log for single sign-on server:

`ORACLE_HOME/opmn/logs/OC4J~OC4J_SECURITY~default_island-1`

Usage Notes:

This OC4J-generated file reports any errors that occur when the single sign-on server is started. Check the file for error messages if the `opmnctl` command hangs or if it reports errors on the command line when the `OC4J_SECURITY` instance is started.

- Web application log:

`ORACLE_HOME/j2ee/OC4J_SECURITY/application-deployments/sso/OC4J_SECURITY_default_island_1/application.log`

Usage Notes:

This file reports run-time errors for OC4J applications.

- OC4J servlet access log:

`ORACLE_HOME/j2ee/OC4J_SECURITY/log/OC4J_SECURITY_default_island_1/default-web-access.log`

Usage Notes:

Another OC4J-generated file. The file contains the servlet access logs for single sign-on. Check the file to determine whether the authentication request was received by the authentication servlet.

- Error log for Oracle HTTP Server:

`ORACLE_HOME/Apache/Apache/logs/error_log`

Usage Notes:

If the Oracle HTTP Server is configured to rotate its log files, it appends a timestamp to these files. Use this timestamp to find the latest file.

- Access log for Oracle HTTP Server:

`ORACLE_HOME/Apache/Apache/logs/access_log`

Usage Notes:

If the Oracle HTTP Server is configured to rotate its log files, it appends a timestamp to these files. Use this timestamp to find the latest file.

Error Messages and Other Problems

This section explains how to address error messages and other problems. It devotes sections to the following:

- [Basic Error Messages and Problems](#)
- [Certificate Authentication](#)
- [Password Policies](#)
- [Type 41400 Errors](#)

Basic Error Messages and Problems

Internal Server Error. Please contact administrator.

Cause: This error message usually appears when the single sign-on server is started incorrectly.

Action: Use the following sequence to solve the problem:

1. Verify that the single sign server was started correctly. To do this, examine the startup log file for errors.
2. If the file reports errors for the database or for Oracle Internet Directory, make sure that both are up and running before starting the single sign-on server. If

you see the message `SSOLoginServlet.init: SSO server started`, the server has been started correctly.

3. Next, check `ssoServer.log`, the log file for the single sign-on server.
4. If the log file contains the error message `NumberFormatException` or a specific configuration parameter not found, check `policy.properties` for blank spaces. Remove spaces that occur at the end of the line containing the questionable configuration; then restart the OC4J_SECURITY instance:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

5. If the file `ORACLE_HOME/opmn/logs/OC4J~OC4J_SECURITY~default_island~1` reports the error message `Orion Launcher SSO Server initialization failed`, do the following:

- * Make sure that the database is available; then restart the single sign-on server:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

- * If the database is available, the problem may be the directory connection. Check the `opmn` log. If you see the error message that follows, run `ssooconf.sql` to ensure that directory access is properly configured in the single sign-on database.

```
java.lang.NumberFormatException: null
  at java.lang.Integer.parseInt(Integer.java:442)
  at java.lang.Integer.parseInt(Integer.java:524)
  at oracle.security.sso.server.conf.DatabaseConfigReader.
    setSSOServerConfig(DatabaseConfigReader.java:322)
```

6. To learn how to run `ssooconf.sql`, see ["Changing Single Sign-On Server Settings for Directory Access"](#) in [Chapter 3](#).

Internal Server Error. Please try the operation later.

Cause: This error message appears when either the infrastructure database or Oracle Internet Directory is unavailable or is down.

Action: Check `ssoServer.log` for a detailed description of the message; then try restarting the database or the directory, depending upon what you find.

Unexpected Error. Please Contact Administrator.

Cause: This message may indicate a server-side error. The `policy.properties` file may be misconfigured, or Java classes may not be loaded. Another cause may be that the partner application is registered incorrectly.

Action: In the first case, check `ssoServer.log` for the actual error message. If this file does not contain the message, check the Oracle HTTP Server error log. In the second case, try to log in to the administration pages:

```
http://single_sign-on_host:single_sign-on_port/pls/orasso
```

Be sure to log in as `orcladmin`, not as `cn=orcladmin`. If you are able to log in, the problem is not with the server, but with the partner application registration or with the application itself.

File not found.

Cause: This message may appear when you try to access the single sign-on server.

Action: Perform two checks:

1. Check the Oracle HTTP Server error log.

If you find the message `file not found`, Apache is not delegating the authentication request to OC4J.

Check `mod_oc4j.conf` for single sign-on application mappings. The mount configuration `Oc4jMount/sso OC4J_SECURITY` should be present.

2. Check `default-web-access.log` to determine whether the authentication request was received by the servlet.

Forbidden. You don't have permission to access /pls/orasso/orasso.home on this server.

Cause: This message may appear when you try to access the single sign-on administration URL. Perhaps the password for the ORASSO schema was changed in the database, but not in the `dads.conf` file.

Action: Perform these steps:

1. Update `ORACLE_HOME/Apache/modplsql/conf/dads.conf`.
2. Restart the Oracle HTTP Server:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

3. If the schema password is correct to begin with, check the Oracle HTTP Server error log for error messages.

Audit log insertion exception: ORA-00018: maximum number of sessions exceeded.

Cause: This message appears when the load on the single sign-on server is heavy. The number of database sessions required has exceeded the number specified in the `init.ora` file.

Action: Change the properties of the identity management infrastructure database. Specifically, increase the `processes` and `sessions` parameters to match anticipated load. Use a database-specific configuration file such as `init.ora` to make the change. `init.ora` is found at `ORACLE_HOME/dbs`.

Connection limit exceeded.

Cause: This is a variation of the message immediately preceding.

Action: The end user should retry the operation. Or the administrator can increase the connection limit.

Single Sign-On Administration UI is not working. The administrator sees a white page when clicking Login

Cause: This problem has three possible explanations:

- Case 1: The PUBLIC user entry is missing from Oracle Internet Directory. Either that or the user nickname attribute was changed in the directory, but the new attribute was not added to the PUBLIC entry.
- Case 2: The single sign-on server is configured with the wrong information for the directory.
- Case 3: There may be installation problems, namely, a missing Enabler entry or faulty SSL registration.
- Case 4: The directory DIT has changed and the single sign-on server has not been updated with the changes.

Action:

- Case 1: Add the PUBLIC user entry under the user search base in the directory. If, instead, the user nickname attribute was changed, add the attribute to the PUBLIC user entry.
- Case 2: Run `ssooconf .sql` to configure the single sign-on server with the correct directory information. To learn how to run the script, see ["Changing Single Sign-On Server Settings for Directory Access"](#) in [Chapter 3](#).
- Case 3: Run `ssooconf .sql` to update the single sign-on server with the enabler entry or to modify single sign-on URLs for SSL.
- Case 4: Run `ssoreoid .sql` to update the single sign-on server with directory DIT changes.

Authentication Failed.

Cause: The user's password is incorrect, or the server does not have the permissions necessary to authenticate the user.

Action:

1. Try binding to the directory as the user, making sure that the user DN corresponds to the appropriate realm:

```
ORACLE_HOME/bin/ldapbind
-h directory_server
-p directory_ssl_port
-D user_dn
-w user_password
-u 1
```

If the bind fails, the user's password is incorrect. Reset the password. If the bind succeeds, proceed to step 2.

2. Try binding to the directory as the single sign-on server:

```
ORACLE_HOME/bin/ldapbind
-h directory_server
-p directory_ssl_port
-D orclApplicationCommonName=ORASSO_
  SSOSERVER,cn=SSO,cn=Products,cn=OracleContext
-w single_sign-on_server_password
```

If the bind fails, the server password that you are trying to bind with may be incorrect. To set the correct password, run `ssooconf .sql` as explained in ["Changing Single Sign-On Server Settings for Directory Access"](#) in [Chapter 3](#). If the bind succeeds, proceed to step 3.

3. Check whether the single sign-on application is a member of the SecurityAdmins group. If it is not a member of this group, it cannot authenticate the user:

```
ORACLE_HOME/bin/ldapcompare
-h directory_host
-p directory_ssl_port
-D orclApplicationCommonName=ORASSO_
  SSOSERVER,cn=SSO,cn=Products,cn=OracleContext
-w orasso_password
-b "cn=user_dn,cn=users,realm_dn"
-a userpassword
-v user_password
```

If the application is not a member, add it to the SecurityAdmins group (cn=OracleUserSecurityAdmins, cn=Groups, cn=OracleContext) and have the user reauthenticate. If the application is a member, the problem may be directory based.

Administrator does not see administration pages when logging in to /pls/orasso.

Cause: The administrator is not a member of the iASAdmins group:

```
cn=iASAdmin, cn=Groups, cn=OracleContext, realm_dn
```

Action: Check the `uniquemember` attribute of the iASAdmins entry in the directory:

```
ldapsearch -h directory_host
            -p directory_port
            -D orclApplicationCommonName=ORASSO_
              SSOSERVER, cn=SSO, cn=Products, cn=OracleContext
            -w orasso_password
            -b "cn=iasadmins, cn=groups, cn=oraclecontext, realm_dn"
              "uniquemember=cn=user, cn=users, realm_dn"
```

If the `user` in the command is not a unique member of iASAdmins, follow the instructions in "[Granting Administrative Privileges](#)" in [Chapter 2](#).

The user name submitted for authentication does not match the user name present in the existing single sign-on session.

Cause: Users encounter this error only during a forced authentication request. They see the error because they fail to enter the same user ID and, optionally, realm that they entered when they first authenticated.

Action: The user ID and, optionally, realm entered during forced authentication must match the user ID and realm in the current single sign-on session. Users who want to use different credentials to log in must log out of the current session first.

The SSO Server Administration link is missing from the administration page

Cause: Only administrators see this link. The user who is missing the link is logged in as an end user.

Action: Make sure that the user is a member of the iASAdmins group:

```
cn=iASAdmins, cn=Groups, cn=OracleContext, dc=default_identity_management_realm
```

If you have changed the SSO administration group, make sure that the user is a member of this group.

Certificate Authentication

This section explains how to debug certificate authentication; then it explains how to interpret error messages.

Debugging Certificate Sign-On

1. Set the debug level in `policy.properties` to `DEBUG`; then restart the single sign-on middle tier:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl startproc process-type=OC4J_SECURITY
```

2. To view browser certificate information while debugging, extract the file `certinfo.jsp` file from `ORACLE_HOME/sso/lib/ipassample.jar`.

3. Place the file into `ORACLE_HOME/j2ee/applications/sso/web/jsp`.
4. View the file at this URL:
`https://host:port/sso/certinfo.jsp`

Error Messages

Network Error: Connection Refused.

Cause: This message appears when the user tries to access a partner application over SSL. The parameter `SSL Engine on` may be missing from `httpd.conf` or may not have been entered correctly.

Action: Add the missing parameter as specified in "Setting SSL Parameters" in [Chapter 7](#). If the parameter is present and is entered correctly, the Oracle HTTP Server log file may identify the problem.

The single sign-on server fails to prompt the user for a certificate.

Cause: The optional parameter `SSLVerifyClient` is missing from `httpd.conf` or has not been entered correctly.

Action: Add the missing parameter as specified in "Setting SSL Parameters" in [Chapter 7](#). If the parameter is present and is entered correctly, the Oracle HTTP Server log file may identify the problem.

Certificate authentication fails to work, and the user is presented with the login page.

Cause: The user's certificate is missing from the directory or has been entered incorrectly. Check `ssoServer.log` for details.

Action: Reenter the user's certificate in the directory. See the instructions in "Oracle Internet Directory" in [Chapter 7](#).

User's browser certificate not found.

Cause: This problem has two possible causes:

- Case 1: The user's certificate is not in the browser.
- Case 2: The SSL wallet on the Oracle HTTP Server may not contain the trusted certificate of the CA that issued the client certificate.

Action:

- Case 1: Install a valid certificate.
- Case 2: Use Oracle Wallet Manager to determine whether the SSL wallet contains the trusted certificate. To learn how to use the tool, see the chapter about managing wallets and certificates in *Oracle Application Server Administrator's Guide*.

Mapping Module class name not found.

Cause: The class name for the mapping module is missing from `x509CertAuth.properties` or is incorrect.

Action: Make sure that a value is assigned to the parameter `CertificateMappingModule`. If it is assigned, check that this value is correct.

Mapping module instance creation failed.

Cause: The customized mapping module has been incorrectly implemented.

Action: Ensure that the custom module has a default constructor.

Cannot create the mapping module object.

Cause: The customized mapping module has been incorrectly implemented.

Action: Ensure that the customized module implements the interface prescribed in "Customize the User Name Mapping Module (Optional)" in Chapter 7.

Exception in creating mapping module.

Cause: The customized mapping module has been incorrectly implemented.

Action: Ensure that the customized module implements the interface prescribed in "Customize the User Name Mapping Module (Optional)" in Chapter 7.

Certificate match failed.

Cause: The user's certificate is missing from the directory or has been entered incorrectly. Check `ssoserver.log` for details.

Action: Reenter the user's certificate in the directory. See the instructions in "Oracle Internet Directory" in Chapter 7.

Password Policies

The administrator disabled a user using the `orclIsEnabled` attribute in Oracle Internet Directory, but the user can still log in.

Cause: The `orclIsEnabled` attribute is incorrect.

Action: Execute `ldapbind` from the command line as the user. If this act invokes an "account disabled error," reenter the attribute value.

The administrator disabled a user using the `orclIsEnabled` attribute in Oracle Internet Directory, but the user receives an "authentication failed error" instead of an "account disabled" error.

Cause: This is expected behavior. If the user's account is disabled, she receives an "authentication failed error."

Action: None.

The user receives a password expiry message when trying to log in.

Cause: The user's password has expired.

Action: The administrator has to reset the password. He or she can enable password expiry warnings in the directory. These warnings prompt users to change their passwords before they expire.

The user logs in to the single sign-on server and is told that her password is about to expire and is prompted to change it. When, however, she does a command-line bind, the message does not appear, and the bind succeeds.

Cause: Certain extended directory messages are not visible through the command-line tools. These messages are visible only through the LDAP client-side APIs.

Action: None.

Type 41400 Errors

When a user sees a WWC-41400 error, it usually means that the single sign-on server is configured incorrectly. The most common errors involve `mod_osso`-protected sites that have been reconfigured. Either the `site2pstoretoken` parameter is invalid or the `site_id` parameter is missing from the `ORASSO.WWSSO_PAPP_CONFIGURATION_INFO$` table. Use the following steps to check these parameters:

1. Try to log in to a protected application. Make a note of any URLs that you use.
2. Display and save the HTTP page source of the single sign-on login page.
3. In the page source, search for the text `site2pstoretoken`. If the parameter is present, it should consist of three elements separated by the tilde character. The middle element in `site2pstoretoken` is the site ID. Here are two examples:

- Release 9.0.2:

```
"v1.2~1321~C4C41209C8E4F0E3E8D....."
```

- Release 9.0.4 or 10.1.2

```
"v1.4~2F02C369~121CBEE9920CDB....."
```

If any one of these elements is missing, `site2pstoretoken` is invalid.

4. If `site2pstoretoken` is valid, determine whether the site ID is missing from single sign-on configuration tables. Log in to the single sign-on schema as `orasso`; then use `SQL*Plus` to search for `site_id` in the `ORASSO.WWSSO_PAPP_CONFIGURATION_INFO$` table. See [Appendix B](#) to obtain the schema password.

WWC-41400 errors may be generated under the following conditions:

The `site2pstoretoken` value is missing.

Cause: The deployment-specific login page does not accept `site2pstoretoken`. Users see a WWC-41400 error after presenting their credentials.

Action: Rewrite the login page to accept `site2pstoretoken` and pass it back to the single sign-on server.

The `site2pstoretoken` value is blank ("").

Cause: The user has accessed the single sign-on login URL directly from the browser.

Action: The user must first access an application protected by `mod_osso` or an application integrated with the now-deprecated single sign-on SDK.

Login parameters are lost during redirection to a third-party server.

Cause: The user has used the POST method to access the single sign-on server. During login a third-party URL is invoked. In release 9.0.2, POST parameters are lost during redirection to the third-party server. This problem does not arise in release 9.0.4 and release 10.1.2 because `mod_osso` uses the GET method.

Action: Use an extra page that uses GET to pass login parameters to the single sign-on server.

The `site2pstoretoken` has an incorrect site ID.

Cause: The partner application has been deleted from single sign-on configuration tables.

Action: Reregister the partner application. See "[Registering mod_osso](#)" in [Chapter 4](#).

The site ID is obsolete.

Cause: The registration that generated the ID may have been removed because it was obsolete.

Action: Determine whether the correct version of the `osso.conf` file is referenced in the `OssosConfigFile` directive of the `mod_osso.conf` file. This file is found at `ORACLE_HOME/Apache/Apache/conf`. It may have been

overwritten accidentally. If you determine that the `osso.conf` file is incorrect, reregister the partner application. See "Registering `mod_osso`" in [Chapter 4](#).

The `site2pstoretoken` has a correct site ID.

Cause: In this case an error is thrown because an incorrectly configured virtual host is being used to access the single sign-on server. The `VirtualHost` container of the `httpd.conf` file—or the `ssl.conf` file if an HTTPS virtual host is defined—is missing the directives `RewriteEngine On` and `RewriteOptions inherit`. Invalid directives may be present as well.

Action: Add the missing directives to `httpd.conf` or `ssl.conf`. Check for invalid ones. Both files are found at `ORACLE_HOME/Apache/Apache/conf`.

Increasing the Debug Level

OracleAS Single Sign-On provides four levels of debugging. They are listed here in ascending order of detail provided.

- `ERROR`—log errors only
- `WARN`—log both errors and warning messages
- `INFO`—log informational messages—current date and time, for instance—as well as errors and warnings
- `DEBUG`—log details about program execution as well as errors, warnings, and informational messages

In the course of debugging, you may have to increase the level of debugging to, say, `DEBUG`. You do this by modifying the file `ORACLE_HOME/ss/conf/policy.properties`.

After changing the debug level, restart the `OC4J_SECURITY` instance:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Enabling the Debug Option in the Single Sign-On Database

Occasionally you may need to debug the `mod_plsql` code used to access external applications. This task requires that you enable debugging on the single sign-on database and then view detail logs. Note that this procedure does not apply to the debugging of partner applications. Debugging information for these applications is stored only in `ORACLE_HOME/ss/log/ssServer.log`.

To turn on `mod_plsql` debugging, log in to the `ORASSO` schema and run the `ssolsdbg.sql` script. See [Appendix B](#) to obtain the schema password. Be sure to uncomment the commented lines in the script before running it. A copy of the script is located at `ORACLE_HOME/ss/admin/plsql/ss`.

Here is the script:

```
set scan off;
set feedback ON;
set verify ON;
set pages 50000;
set serveroutput ON;

CREATE OR replace PROCEDURE debug_print (str VARCHAR2) AS
BEGIN
```

```

INSERT INTO wwsso_log$ VALUES (wwsso_log_pk_seq.nextval,
    substr(str, 1, 1000),
    sysdate, dbms_session.unique_session_id);

commit;

END debug_print;
/

show errors;

```

To query the debug logs, issue this command:

```
SELECT * FROM WWSO_LOG$ ORDER BY ID;
```

To turn off debugging, log in to the ORASSO schema and create the PL/SQL script that follows. Be sure to include this step when you finish debugging. If you skip it, superfluous records are created in the database table. See [Appendix B](#) to obtain the schema password.

```

set scan off;
set feedback ON;
set verify ON;
set pages 50000;
set serveroutput ON;

CREATE OR replace PROCEDURE debug_print (str VARCHAR2) AS
-- PRAGMA autonomous_transaction;
BEGIN

    null;

END debug_print;
/

show errors;

```

Enabling LDAP Tracing for UI Operations

The administration pages for single sign-on use the DBMS_LDAP package to perform directory operations. You can obtain details about these operations in the debug logs for the single sign-on database. To pinpoint the error though, you must enable client-side LDAP tracing. In trying, for example, to determine why an administrator cannot see administration links on the single sign-on home page, you can determine the exact point at which an error is being returned by the LDAP client-side APIs. You can then find the trace results in the RDBMS trace files.

Follow these steps to perform client-side tracing:

1. Enable tracing by loading the `debugonldap.sql` package into the ORASSO schema:

```
SQL> connect orasso/password
```

See [Appendix B](#) to obtain the schema password.

2. Run the script:

```
SQL> @debugonldap.sql
```

`debugonldap.sql` looks like this:

```

set scan off;
set feedback ON;
set verify ON;
set pages 50000;
set serveroutput ON;

CREATE OR replace PROCEDURE debug_print (str VARCHAR2) AS
BEGIN

    dbms_ldap.set_trace_level(65535);

    INSERT INTO wwssso_log$ VALUES
        (wwssso_log_pk_seq.nextval, substr(str, 1, 1000), sysdate,
        dbms_session.unique_session_id);

    commit;

END debug_print;
/

show errors;
```

3. Perform a single sign-on operation that raised an error or that requires debugging—logging in to the home page as an administrator, for instance.
4. Examine the LDAP client logs in the RDBMS trace directory. You can determine the location of this directory by connecting as `SYS` to the identity management infrastructure database and then issuing this command:

```
SQL> show parameter user_dump_dest
```

The value returned is the directory where trace files are written. Once in the directory, examine the file timestamps to find the relevant file.

If the client-side trace files fail to reveal the problem, enable server-side tracing, but perform client-side tracing first. To enable server-side tracing, see the chapter about logging, auditing, and monitoring in *Oracle Internet Directory Administrator's Guide*.

To disable tracing, load and run this package:

```

set scan OFF;
set feedback ON;
set verify ON;
set pages 50000;
set serveroutput ON;

CREATE OR replace PROCEDURE debug_print (str VARCHAR2) AS BEGIN
null;
END debug_print;
/
show errors;
```

Managing Single Sign-On Audit Records

The single sign-on server records authentication failures and successes in the Oracle Identity Management database. In time, the audit table, `ORASSO.WWSSO_AUDIT_`

LOG_TABLE_T, runs out of space. When this happens, this error message appears in database alert logs:

```
ORA-1654: unable to extend index ORASSO.AUDIT_INDEX1 by 128 in tablespace IAS_META
```

In addition, further authentication requests fail.

Be sure to monitor `ORASSO.WSSO_AUDIT_LOG_TABLE_T` regularly. When it becomes full, either back up the table and free up space or add space. Note that this is an internal, product-specific table. This means that you can use SQL*Plus to clean up the table, but you cannot use this tool or any other tool to build reporting or monitoring scripts based on the table.

Refreshing the LDAP Connection Cache

For performance reasons, the single sign-on server caches connections to Oracle Internet Directory. If the directory server has a scheduled or unscheduled outage, the single sign-on server is left holding bad directory connections, and users may encounter directory setup errors when they try to access external applications. If the LDAP connection cache is invalid, the Oracle HTTP Server must be restarted:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

Use the following steps to determine whether the LDAP connection cache must be refreshed:

1. Connect to the single sign-on schema. See [Appendix B](#) to obtain the schema password.
2. Issue the following command:


```
SELECT * FROM WSSO_LOG$
```
3. Restart the HTTP server if you see the following error in the log:


```
'INVALID LDAP CONNECTION CACHE: RESTART ORACLE HTTP SERVER'
```
4. Delete the error message from `WSSO_LOG$`.

Restarting OC4J After Modifying Oracle Internet Directory

If you change values in Oracle Internet Directory, you must update the single sign-on server with the changes. If, for example, you change a user, subscriber, or group search base in the directory and fail to "notify" the single sign-on server, users under the modified container are unable to log in. The `ssoreoid.sql` script updates the single sign-on server with directory changes. To learn how to run the script, see ["Updating the Single Sign-On Server with Directory Changes"](#) in [Chapter 3](#).

After running the script, make sure that you restart the single sign-on server:

```
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_SECURITY
```

Troubleshooting Replication

Deploying geographically distributed single sign-on instances requires, among other things, that you replicate the identity management infrastructure database. Each time you replicate the database, you should validate the replication process on each replicated node and correct errors that you find. Use the Replication Environment Management Tool (`remtool`) to complete both tasks.

`remtool` is run on the master definition site. You can find the tool at `ORACLE_HOME/ldap/bin`. It can be executed in two modes as the sections that follow explain.

Verifying Oracle9i Advanced Replication Configuration

To verify that the directory replication group has been successfully configured, issue the following command:

```
remtool -asrverify
```

The command option `-asverify` instructs the tool to report on the verification as it progresses, but not to rectify problems.

Verifying and Rectifying Oracle9i Advanced Replication Configuration

To verify that a directory replication group has been successfully configured and to rectify problems, execute the following command:

```
remtool -asrrectify -v -connect repadmin/repadmin_password@net_service_name
```

The command option `-asrectify` instructs the tool to report on the verification as it progresses and to rectify problems. [Table A-1](#) defines the two other command parameters.

Table A-1 Parameters for the Replication Environment Management Tool

Parameter	Description
<code>-v</code>	Verbose mode. Specifying this option not only shows the progress of <code>remtool</code> , it also logs all actions of the tool in <code>remtool.log</code> . This file is found at <code>ORACLE_HOME/ldap/log</code> .
<code>-connect</code>	Connect string of the database for the remote master site. As the command syntax shows, this string has three components: <ul style="list-style-type: none"> ▪ <code>repadmin</code>—Name of the replication administrator ▪ <code>repadmin_password</code>—Password of the replication administrator ▪ <code>net_service_name</code>—Network service name of the RMS database.

For an in-depth look at the Replication Environment Management Tool, see the syntax chapter in *Oracle Internet Directory Administrator's Guide*.

A Word About Non-GET Authentication

The first page of a `mod_osso`-protected application must be a URL that uses the GET authentication method. If the POST method is used, the data that the user provides when logging in is lost during redirection to the single sign-on server. When deciding whether to enable the global user inactivity timeout, note that users are redirected after timing out and logging in again.

Need More Help?

You can find more solutions on Oracle *MetaLink*, <http://metalink.oracle.com>. If you do not find a solution for your problem, log a service request.

Obtaining the Single Sign-On Schema Password

The single sign-on schema password is randomized when the Oracle Application Server infrastructure is installed. You can use either the command-line tool `ldapsearch` or Oracle Directory Manager to obtain the password.

Using the Command Line

Use this syntax to obtain the schema password with `ldapsearch`:

```
ldapsearch -h directory_host_name
           -p directory_ssl_port
           -D directory_bind_dn
           -w directory_bind_dn_password
           -b "orclReferenceName=infrastructure_database"
              "orclresourcename=ORASSO"
              orclpasswordattribute
           -u 1
```

The table that follows defines the parameters passed to `ldapsearch`.

Parameter	Description
<i>directory_host_name</i>	Host name of the directory server.
<i>directory_ssl_port</i>	Port number of the directory server.
<i>directory_bind_dn</i>	Distinguished name of the user authenticating to the directory.
<i>directory_bind_dn_password</i>	Password of the user authenticating to the directory.
<i>infrastructure_database</i>	Distinguished name of the directory entry under which the password attribute (<code>orclpasswordattribute</code>) is located.
<code>-u</code>	Changes the directory port to an SSL port globally.

Here is an example:

```
ldapsearch -h oid.acme.com
           -p 636
           -D "cn=orcladmin"
           -w welcome1
           -b "orclReferenceName=disco.us.acme.com,cn=IAS Infrastructure"
```

```
Databases,cn=IAS,cn=Products,cn=oraclecontext"  
"orclresource=ORASSO"  
orclpasswordattribute  
-u 1
```

Using Oracle Directory Manager

Follow these steps to obtain the schema password with Oracle Directory Manager:

1. Launch the tool:

```
ORACLE_HOME/bin/oidadmin
```

2. In the **System Objects** frame, expand in succession the following entries:

- Entry Management
- cn=OracleContext
- cn=Products
- cn=IAS
- cn=IAS Infrastructure Databases
- orclReferenceName=*database_service_name_for_infrastructure_database*
- OrclResourceName=ORASSO

The **orclpasswordattribute** text box on the **OrclResourceName=ORASSO** tab contains the schema password.

policy.properties

The `policy.properties` file, provided here, is a multipurpose configuration file that contains basic parameters required by the single sign-on server. The file is also used to implement advanced features such as multilevel authentication.

```
# SSO Server policy configurations

#####
# Authentication Levels
# -----
# Set the auth levels from lower value to higher value.
# 10 being the lowest authentication level
# The auth level names (on the right hand side) can be changed to
# some other names if desired as long as the change is consistent
# in other places of the file.

NoSecurity = 10
LowSecurity = 20
LowMediumSecurity = 30
MediumSecurity = 40
MediumHighSecurity = 50
HighSecurity = 60

# DefaultAuthLevel
# -----
# DefaultAuthLevel entry must have a value assigned. This is a mandatory
# requirement if any of the partner app URLs are not listed with the
# auth level mapping.
# If partner app url does not specify the auth level, then the DefaultAuthLevel
# will be used.

DefaultAuthLevel = MediumSecurity

#####
# Protected URL configurations
# -----
# Assign a auth level to each protected (partner) application that is
# participating in SSO. If any of the partner apps are not listed with
# a specific auth level, then the DefaultAuthLevel will be used.
#
# Protected application URL configuration format:
# "Partner Application Root URL" = "AuthenticationLevel"
# host.company.com\:port = AuthLevelName
# NOTE: The required backslash(escape character) before the
# colon (:) character immediately preceding.
# There should be a corresponding auth plugin configured for the
```

```

# "AuthenticationLevel" used.
#
# Examples:
# The following example configures a SSO partner application hosted
# on host1.company.com:7777 machine using LowSecurity authentication level.
# This configuration will secure all URLs hosted on this host/port.
# host1.company.com\:7777 = LowSecurity
#
# The following example configures a SSO partner application hosted
# on host2.company.com:7777 machine using MediumSecurity authentication level
# This configuration will secure all URLs hosted on this host/port.
# host2.company.com\:7777 = MediumSecurity

#####
# Authentication plugins
# -----
# Assign a class name that implements SSOServerAuthInterface for each auth
# level defined
#
# Note: also see the WeakAuthLevel attribute which must be set to
# the same auth level corresponding to the weak auth mechanism
#
# The Authentication level name must be appended with "_AuthPlugin"
# keyword.
LowSecurity_AuthPlugin = oracle.security.sso.server.auth.SSOServerWeakAuth
MediumSecurity_AuthPlugin = oracle.security.sso.server.auth.SSOServerAuth

#####
# Custom Cookie Provider Class name
# -----
# Sample custom cookie tester provider class
# CustomCookie_ProviderPlugin = oracle.security.sso.server.auth.CustomCookieTester

# Custom Cookie auth level
# -----
# This is a mandatory attribute. If custom cookies are not needed it should
# be set to a higher value than any of the authentication levels used.

CustomCookieAuthLevel = HighSecurity

#####
#SSO Server specific configurations

# set the cache size in kbytes
#default is 1000
#cacheSize = 1000

#set the minimum number of connections in the connection pool
#default is 5
minConnectionsInPool = 5

#set the maximum number of connections in the connection pool
#default is 150
maxConnectionsInPool = 150

#Debug level {ERROR, WARN, INFO, DEBUG}
# default debug level is set to ERROR

```

```
debugLevel = ERROR

#Debug file location
#This is a mandatory property that needs to be passed
#the SSO server. A valid file location should be specified here
debugFile = %ORACLE_HOME%/sso/log/ssoServer.log

#Custom login page link
loginPageUrl = /sso/jsp/login.jsp

#Custom weak authentication login page link
weakAuthLoginPageUrl = /sso/jsp/ssoWeakAuthLogin.jsp

#Custom change password page link
chgPasswordPageUrl = /sso/jsp/password.jsp

#Wireless login page link
wirelessLoginPageUrl = /wireless/sso/wirelesslogin.jsp
wirelessChgPasswordPageUrl = /wireless/sso/wirelesscpwd.jsp
```

Glossary

account lockout

Occurs when a single sign-on user submits an account and password combination from any number of workstations more times than is permitted by Oracle Internet Directory. The default lockout period is 24 hours.

application service provider

Company that installs and maintains Web applications and makes them available to its customers, typically for a fee.

authentication level

Parameter that enables you to specify a particular authentication behavior for an application. You can link this parameter with a specific authentication plugin.

authentication plugin

An implementation of a specific authentication method. OracleAS Single Sign-On has Java plugins for password authentication, digital certificates, Windows native authentication, and third-party access management.

basic authentication

An authentication method whereby login credentials are submitted in the application URL, which is protected by HTTP basic authentication.

certificate revocation list

A list of users whose X.509 certificates have been revoked. An application uses this list to determine who gains access to the application.

dads.conf

The file on the Oracle HTTP Server that is used to configure a database access descriptor (DAD).

database access descriptor (DAD)

Database connection information for a particular OracleAS component such as the single sign-on schema.

digital certificate

In asymmetric encryption, a data structure that vouches for the identity of a public key owner. A certificate is issued by a trusted third party called a certificate authority. As such it provides assurance that the public key may be safely used to encrypt messages to the key owner.

directory information tree (DIT)

The hierarchical collection of entries that constitute an LDAP directory.

distinguished name

A name that identifies the location of an entry in an LDAP-compliant directory. Also known as a DN. The distinguished name of the user in the example that follows consists of his name and parent entries in ascending order, from left to right.

```
cn=jsmith,cn=users,cn=defaultsubscribers,cn=acme,cn=com
```

external application

Applications that do not delegate authentication to the single sign-on server. Instead, they display HTML login forms that ask for application user names and passwords. At the first login, users can choose to have the single sign-on server retrieve these credentials for them. Thereafter, they are logged in to these applications transparently.

forced authentication

The act of forcing a user to reauthenticate if he or she has been idle for a preconfigured amount of time. OracleAS Single Sign-On enables you to specify a global user inactivity timeout. This feature is intended for installations that have sensitive applications.

GET

An authentication method whereby login credentials are submitted as part of the login URL.

global user inactivity timeout

An optional feature that forces single sign-on users to reauthenticate if they have been idle for a preconfigured amount of time. The global user inactivity timeout is much shorter than the single sign-out session timeout.

globalization support

Multilanguage support for graphical user interfaces. OracleAS Single Sign-On supports 29 languages.

globally unique user ID

A numeric string that uniquely identifies a user. A person may change or add user names, passwords, and distinguished names, but her globally unique user ID always remains the same.

httpd.conf

The file used to configure the Oracle HTTP Server.

identity management realm

Discrete namespace, or DIT, within a single instance of the Oracle Identity Management infrastructure.

iASAdmins

The administrative group responsible for user and group management functions in OracleAS. The single sign-on administrator is a member of the group iASAdmins.

identity management infrastructure database

The database that contains OracleAS Single Sign-On and Oracle Internet Directory.

infrastructure

The OracleAS components responsible for identity management. These components are OracleAS Single Sign-On, Oracle Delegated Administration Services, and Oracle Internet Directory.

LDAP connection cache

To improve throughput, the single sign-on server caches and then reuses connections to Oracle Internet Directory.

legacy application

Older application that cannot be modified to delegate authentication to the single sign-on server. Also known as an external application.

load balancer

Hardware devices and software that balance connection requests between two or more single sign-on servers, either because of heavy load or as failover. BigIP, Alteon, or Local Director are all popular hardware devices. OracleAS Web Cache is an example of load balancing software.

middle tier

That portion of a single sign-on instance that consists of the Oracle HTTP Server and OC4J. The single sign-on middle tier is situated between the identity management infrastructure database and the client.

mod_oss1

The SSL module on the Oracle HTTP Server.

mod_osso

A module on the Oracle HTTP Server that enables applications protected by OracleAS Single Sign-On to accept HTTP headers in lieu of a user name and password once the user has logged into the single sign-on server. The values for these headers are stored in the mod_osso cookie.

mod_osso cookie

User data stored on the HTTP server. The cookie is created when a user authenticates. When the same user requests another application, the Web server uses the information in the mod_osso cookie to log the user in to the application. This feature speeds server response time.

mod_proxy

A module on the Oracle HTTP Server that makes it possible to use mod_osso to enable legacy, or external, applications.

OC4J (Oracle Containers for J2EE)

A lightweight, scalable container for Java2 Enterprise Edition.

Oracle Delegated Administration Services

A Web service of Oracle Internet Directory that performs user and group management functions.

Oracle Directory Integration and Provisioning

A feature of Oracle Internet Directory that enables an enterprise to use an external user repository to authenticate to Oracle products.

Oracle Directory Manager

A Java-based GUI for managing most functions in Oracle Internet Directory. It is used to create members of the group iASAdmins. It is also used to manage password policies.

Oracle Enterprise Manager

The GUI that monitors server load and user activity on the single sign-on server. Oracle Enterprise Manager monitors other OracleAS components as well.

Oracle HTTP Server

Software that processes Web transactions that use the Hypertext Transfer Protocol (HTTP). Oracle uses HTTP software developed by the Apache Group.

OracleAS Portal

A single sign-on partner application that provides a mechanism for integrating files, images, applications, and Web sites. The External Applications portlet provides access to external applications.

partner application

An OracleAS application or non-Oracle application that delegates the authentication function to the single sign-on server. This type of application spares you from reauthenticating by accepting mod_osso headers.

policy.properties

Multipurpose configuration file for OracleAS Single Sign-On. Contains basic parameters required by the single sign-on server. Also used to configure advanced features such as multilevel authentication.

POST

An authentication method whereby login credentials are submitted within the body of the login form.

proxy server

A server that proxies for the real server, or host. In OracleAS Single Sign-On, proxies are used for load balancing and as an extra layer of security. See load balancer.

SSL (Secure Sockets Layer)

A widely used security protocol that uses public-key cryptography to secure communications between a client and server. The client uses a public key provided by the server to conduct a secret key exchange.

single sign-on SDK

The APIs that enable partner applications for single sign-on. The SDK consists of PL/SQL and Java APIs as well as sample code that demonstrates how these APIs are implemented. Now deprecated.

single sign-on server

Program logic that enables users to log in securely to single sign-on applications such as expense reports, mail, and benefits.

single sign-off

The process by which you terminate a single sign-on session and log out of all active partner applications simultaneously. You can do this by logging out of the application that you are working in.

success URL

The URL to the routine responsible for establishing the session and session cookies for an application.

third-party access management system

Non-Oracle single sign-on system that can be modified to use OracleAS Single Sign-On to gain access to OracleAS applications.

URLC token

The code that passes authenticated user information to the partner application. The partner application uses this information to construct the session cookie.

user name mapping module

A Java module that maps a user certificate to the user's nickname. The nickname is then passed to an authentication module, which uses this nickname to retrieve the user's certificate from the directory.

virtual host

A server that proxies for the real server or servers. In the case of OracleAS Single Sign-On, virtual hosts are used for load balancing between two or more single sign-on servers. They also provide an extra layer of security.

Index

A

account lockout, 3-3
addsub.csh script, 9-5
Administer External Applications page, 5-1 to 5-4
administration pages
 accessing, 2-7
 debugging, A-11, A-12
 external applications, 5-1
application service providers, 9-1
authentication adapters. See authentication plugins
authentication dynamics
 certificate-enabled sign-on, 7-1, 7-2
 identity management realms, 9-3, 9-4
 third-party access management, 12-1, 12-3
authentication levels, 6-2, 6-3
authentication plugins, 6-3

B

backup and recovery, 8-15
basic authentication method, 5-2, 5-6
browser settings
 standard, 2-6

C

certificate revocation lists, 7-7
certificate-enabled sign-on
 authentication dynamics, 7-1, 7-2
 configuring
 Oracle HTTP Server, 7-2, 7-3
 Oracle Internet Directory, 7-6, 7-7
 single sign-on server, 7-3 to 7-6
 user name mapping module, 7-4, 7-6
 CRL maintenance, 7-7
 error messages, A-6, A-8
 sample files, 2-11
change password page
 behavior, 3-2
 error messages, 11-6
 installing, 11-9
 overview, 1-6
 parameters, 11-5
configuration files
 httpd.conf, 4-6, 8-9, A-7

osso.conf, 4-7, 8-11
policy.properties, 6-2 to 6-4, 7-4, A-6, A-10, C-1
ssl.conf, 7-2, 7-3
targets.xml, 10-3
x509CertAuth.properties, 7-4, 7-6

D

debugging
 administration pages, A-11, A-12
 PL/SQL pages, A-10, A-11
deployment scenarios
 geographically distributed instances, 8-13
 multilevel authentication, 6-3, 6-4
 multiple middle tiers, 8-7
 partner applications, 4-4
deployment-specific pages
 examples, 11-10
 globalization support, 11-7
 guidelines, 11-8
 installing, 11-9, 11-10
 sample files, 2-11
 support for OracleAS Wireless, 11-9
Details of Login Failures page, 10-3
directory access
 configuring, 3-5
 scripts, 3-5
directory entries, for OracleAS Single Sign-On, 3-3, 3-4
disaster recovery, 8-15
Distributed Cluster Management, 4-7, 8-10

E

Edit SSO Server page, 2-8
enblhstg.csh script, 9-5
error messages
 basic, A-2, A-6
 certificate-enabled sign-on, A-6, A-8
 export and import, 13-5 to 13-7
 password policies, A-8
 type 41400, A-8, A-10
export and import
 error messages, 13-5 to 13-7
 scenarios, 13-3
 scripts, 13-4

- external applications
 - access using mod_osso/mod_proxy, 5-5 to 5-7
 - adding, 5-1 to 5-4
 - administration pages, 5-1
 - authentication dynamics, 1-4, 1-5
 - authentication methods
 - basic, 5-2, 5-6
 - GET, 5-2
 - POST, 5-2
 - editing, 5-4
 - login, 5-4
 - overview, 1-2
- External Applications portlet, 5-4

F

- force change password feature, 3-3

G

- GET authentication method, 5-2, A-14
- global user inactivity timeout, A-14
 - configuring, 2-9, 2-11
 - overview, 1-6
 - scripts, 2-9
- globalization support
 - deployment-specific pages, 11-7
 - standard pages, 2-8
- grace login, 3-3

H

- high server availability
 - configuring, 8-15
 - deployment options
 - geographically distributed instances, 8-13 to 8-15
 - multiple middle tiers, 8-7 to 8-12
 - replicated directory, 8-13
- httpd.conf file, 4-6, 4-9, 8-9, A-7

I

- iASAdmins administration group, 2-1
- identity management infrastructure database
 - configuring for SSL, 8-3
 - replicating, 8-15 to 8-18
 - support for multiple realms, 9-1
- identity management realms
 - administrative privileges, 9-7
 - authentication dynamics, 9-3, 9-4
 - benefits, 9-1
 - configuring, 9-4 to 9-6
 - overhead, 9-1
 - overview, 9-1
 - support for partner applications, 9-3
- IP checking, 2-8

L

- LDAP command-line tools, 3-2

- LDAP connection cache, A-13
- load balancers
 - OracleAS Web Cache, 8-10
 - with multiple partner applications, 4-4, 4-7
 - with multiple single sign-on middle tiers, 8-7, 8-8, 8-10
- log files, A-2
- login page
 - error messages, 11-6
 - installing, 11-9
 - parameters, 11-3
 - password reset feature, 11-4
- login scenarios
 - third-party access, 12-2

M

- mod_osso
 - compared with single sign-on SDK, 1-2
 - overview, 1-2
 - registering, 4-1
 - reregistering, 4-7, 4-8, 8-11, 8-20
- mod_osso.conf file, 2-10, 4-9
- monitoring home page, 10-1
- monitoring pages
 - accessing, 10-1
 - ports, 10-3
- multilevel authentication
 - authentication levels, 6-2, 6-3
 - configuring, 6-3, 6-4
 - flow, 6-1
 - plugins, 6-3
- multimaster replication, 8-16

O

- oidprovtool, 8-21
- Oracle Delegated Administration Services, 1-6, 3-1
- Oracle Directory Manager, 2-3, 3-1
- Oracle HTTP Server
 - configuring
 - certificate-enabled sign-on, 7-2, 7-3
 - partner application middle tier, 4-6
 - single sign-on middle tier, 8-9, 8-10
 - SSL configuration, 8-3
 - starting and stopping, 2-5
- Oracle Internet Directory
 - configuring for certificate-enabled sign-on, 7-6, 7-7
 - configuring for SSL, 8-5
 - role in third-party access management, 12-3
 - synchronization with Microsoft Active Directory, 12-3
- OracleAS Certificate Authority, 7-3
- OracleAS Cold Failover Cluster, 8-15
- OracleAS Discoverer, 13-1, 13-3, 13-4
- OracleAS Portal
 - External Applications portlet, 5-4
 - registering, 4-1

- OracleAS Single Sign-On
 - administrative pages, 1-3
 - administrators, 2-1 to 2-3
 - benefits, 1-1
 - browser preferences, 2-6, 2-7
 - configuring directory access, 3-5
 - directory information tree, 3-3, 3-4
 - external applications, 5-1 to 5-4
 - globalization support, 2-8, 11-7
 - home page, 1-3
 - nondefault configuration, 8-1
 - password policies, 3-2, 3-3
 - passwords, 1-6
 - sample files, 2-11
 - schema, 1-3
 - scripts
 - addsub.csh, 9-5
 - enblhstg.csh, 9-5
 - ssocfg, 8-10, 8-19
 - ssogito.sql, 2-9
 - ssomig, 13-1
 - ssooconf.sql, 3-5, A-3, A-5
 - ssoreoid.sql, 3-6, A-5, A-13
 - timeouts, 1-6
 - user accounts, 3-1
 - user attributes, 1-2
- OracleAS Web Cache, 4-7, 8-10
- OracleAS Wireless, 1-6
- ossoca.jar tool, 2-8, 2-9
- osso.conf file, 4-7, 8-11

P

- partner applications
 - configuring for high availability, 4-4
 - deploying, 4-4
 - examples of, 1-2
 - overview, 1-2
 - registering, 4-1
 - reregistering, 4-7, 4-8, 8-5, 8-11
- password policies, 3-2, 3-3
- passwords
 - changing, 1-6, 3-2
 - configuring, 3-3
 - expiry, 3-2
 - external applications, 1-2
 - force change password feature, 3-3
 - management, 3-2
 - resetting, 1-6, 3-2, 11-4
 - rules, 3-2
 - schema, 3-6, B-1
- policy.properties file
 - in certificate-enabled sign-on, 7-4
 - in debugging, A-6, A-10
 - in multilevel authentication, 6-2 to 6-4
 - in third-party access management, 12-8
 - purpose, 2-4
 - sample, C-1
- POST authentication method, 5-2, A-14
- proxy authentication, 5-5 to 5-7

- proxy server
 - configuring, 8-19, 8-20
 - function, 8-18

R

- refresh script, 3-6
- Remember My Login Information For This
 - Application check box, 5-4
- Replication Environment Management Tool, A-13, A-14
- reverse proxy, 8-18 to 8-20

S

- sample files
 - certificate-enabled sign-on, 2-11
 - deployment-specific pages, 2-11
- scripts
 - ssogito.sql, 2-9
 - ssomig, 13-1, 13-2
 - ssooconf.sql, 3-5
 - ssoreg, 4-1, 4-2, 4-3, 4-4, 8-11, 8-20
 - ssoreoid.sql, 3-6
- server cache, 3-6
- single sign-off page
 - installing, 11-9
 - parameters, 11-5
- single sign-on administrators
 - assigning privileges to, 2-1
 - creating a new group for, 2-3, 2-4
 - duties, 2-1
- single sign-on server
 - accessing, 1-3
 - cache, 3-6
 - configuring directory access, 3-5
 - deployment options
 - geographically distributed instances, 8-13 to 8-15
 - multiple middle tiers, 8-7, 8-13
 - replicated directory, 8-13
 - LDAP connection cache, A-13
 - log files, A-2
 - overview, 1-1
 - role in third-party access management, 12-1
 - starting and stopping, 2-5
 - with reverse proxy, 8-18
- single sign-on session timeout, 2-8
- SSL (Secure Sockets Layer), 8-1 to 8-5
- ssl.conf file, 4-8, 4-9, 7-2, 7-3
- SSO Server Administration page, 2-7
- ssocfg script, 8-10, 8-19
- ssogito.sql script, 2-9
- ssomig script
 - executing, 13-4
 - parameters, 13-2, 13-3
 - syntax, 13-1
- ssomig.log file, 13-5
- ssooconf.sql script, 3-5, A-3, A-5

- ssoreg script
 - example, 4-4
 - purpose, 4-1
 - syntax, 4-2
 - usage, 8-11, 8-20
- ssoreoid.sql script, 3-6, A-5, A-13
- ssoReplSetup.jar tool, 8-17
- synchronization
 - between directory and single sign-on server, 8-20, 8-22
 - between third-party directory and Oracle Internet Directory, 12-3

T

- targets.xml file, 10-3
- third-party access management
 - authentication dynamics, 12-1, 12-3
 - code example, 12-7, 12-9
 - logout, 12-6, 12-7
 - migration, 12-9 to 12-10
- timeouts
 - global user inactivity timeout, 1-6, 2-9, 2-11
 - single sign-on session timeout, 2-8

U

- URLs, configuring for SSL, 8-5
- URLs, protecting, 8-3 to 8-5, 8-10
- user accounts
 - lockout, 3-3
 - management, 3-1
- user management tools, 3-1
- user name mapping module, 7-4
 - custom implementation, 7-5, 7-6
 - default implementation, 7-4

V

- virtual hosts, 4-8, 4-9, 8-19

X

- X509CertAuth.properties file, 7-4, 7-6