

# **Oracle® Application Server**

Enterprise Deployment Guide

10g Release 2 (10.1.2)

**B13998-07**

March 2008

Oracle Application Server Enterprise Deployment Guide, 10g Release 2 (10.1.2)

B13998-07

Copyright © 2005, 2008, Oracle. All rights reserved.

Primary Author: Julia Pond

Contributing Authors: Janga Aliminati, Peter Lubbers, Greg Sowa, Tim Willard

Contributors: Senthil Arunagirinathan, Rachel Chan, Orlando Cordero, Eileen He, Pavana Jain, Pushkar Kapasi, Rajiv Maheshwari, Mark Nelson, Lei Oh, Ted Regan, Theresa Bandy, Malai Stalin, Yaqing Wang

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 USA, Inc., 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, JD Edwards, PeopleSoft, and Siebel are registered trademarks 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.

---

---

# Contents

## 1 Overview

1.1	What is an Enterprise Deployment? .....	1-1
1.2	Benefits of the Oracle Application Server Enterprise Deployment Configurations .....	1-2
1.2.1	Built-in Security .....	1-2
1.2.2	High Availability .....	1-2

## 2 Selecting a Deployment Architecture

2.1	Creating Solutions with Oracle Application Server.....	2-1
2.2	Enterprise Deployment Nomenclature.....	2-1
2.3	Understanding the Enterprise Deployment Architectures.....	2-2
2.3.1	myJ2EE .....	2-2
2.3.2	myPortal.....	2-5
2.3.3	myBIFCompany .....	2-8
2.4	Understanding Deployment Variants.....	2-10
2.4.1	Understanding Data Tier Variants.....	2-10
2.4.1.1	Using Multimaster Replication with Oracle Internet Directory .....	2-10
2.4.1.2	Using the Oracle Application Server Cold Failover Cluster (Identity Management) Solution .....	2-11
2.4.1.2.1	Implementing the OracleAS Cold Failover Cluster (Identity Management) Solution .....	2-11
2.4.2	Understanding Identity Management Tier Variants.....	2-12
2.4.2.1	Oracle Internet Directory: Data Tier or Identity Management Tier? .....	2-12
2.4.2.2	Oracle Internet Directory: AD/iPlanet Integration .....	2-12
2.4.2.3	Oracle Application Server Single Sign-On: Using Netegrity .....	2-12
2.4.2.4	Oracle Application Server Single Sign-On: Windows Authentication.....	2-13
2.4.3	Understanding Application Tier Variants .....	2-13
2.4.3.1	J2EE Applications: File Based or Database Repository? .....	2-13
2.4.4	Understanding Web Server Tier Variants.....	2-14
2.4.4.1	Oracle Application Server Web Cache Placement, Clustering and Deployment Considerations .....	2-14
2.4.4.2	Oracle HTTP Server: Forward and Reverse Proxies .....	2-15
2.4.4.3	Oracle HTTP Server as a Standalone Web Server.....	2-15
2.5	How to Use this Guide: The Enterprise Deployment Configuration Process.....	2-16
2.5.1	Installing and Configuring myJ2EE.....	2-16
2.5.2	Installing and Configuring myPortal.....	2-17
2.5.3	Installing and Configuring myBIF .....	2-18

2.6	Selecting a Deployment Architecture .....	2-19
<b>3</b>	<b>Before You Begin Installation</b>	
3.1	Best Practices for Installing and Configuring Enterprise Deployments .....	3-1
3.2	Hardware Sizing Guidelines .....	3-1
3.3	Managing Oracle Application Server Component Connections .....	3-2
<b>4</b>	<b>Installing and Configuring the Security Infrastructure</b>	
4.1	Installing the Oracle Application Server Metadata Repository for the Security Infrastructure .....	4-1
4.1.1	Installing the OracleAS Metadata Repository Creation Assistant.....	4-2
4.1.2	Installing the Metadata Repository in a Database Using Raw Devices.....	4-3
4.1.3	Installing the Metadata Repository in an Oracle Cluster File System (OCFS) .....	4-5
4.1.4	Updating the sqlnet.ora File for OracleAS Portal Communication.....	4-6
4.1.5	Configuring the Time out Value in the sqlnet.ora File.....	4-6
4.2	Installing the Oracle Internet Directory Instances in the Data Tier .....	4-7
4.2.1	Installing the First Oracle Internet Directory.....	4-7
4.2.2	Installing the Second Oracle Internet Directory .....	4-13
4.3	Configuring the Virtual Server to Use the Load Balancing Router .....	4-20
4.4	Testing the Data Tier Components.....	4-20
4.5	Installing and Configuring Authentication Services for myPortalCompany.com.....	4-21
<b>5</b>	<b>Installing and Configuring Authentication Services</b>	
5.1	Option 1: Using Oracle Application Server Single Sign-On .....	5-1
5.1.1	Installing the First Identity Management Configuration.....	5-1
5.1.2	Testing the Identity Management Components With Oracle Internet Directory .....	5-8
5.1.3	Installing the Second Identity Management Configuration.....	5-8
5.1.4	Testing the Identity Management Tier Components .....	5-15
5.2	Option 2: Using the Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider .....	5-15
5.2.1	Adding Administrative Users and Groups to Oracle Internet Directory for the OracleAS JAAS Provider .....	5-16
<b>6</b>	<b>Installing and Configuring the myJ2EECompany Application Infrastructure</b>	
6.1	Installing and Configuring the Security Infrastructure.....	6-1
6.2	Configuring the Load Balancing Router or Proxy Server .....	6-2
6.3	Installing and Configuring the Application Tier.....	6-2
6.3.1	Installing the First Application Tier Application Server Instance on APPHOST1 ....	6-2
6.3.2	Installing the Second Application Tier Application Server Instance on APPHOST2	6-6
6.3.3	Creating OC4J Instances on the Application Tier .....	6-10
6.3.4	Deploying J2EE Applications.....	6-11
6.3.5	Creating a DCM-Managed Oracle Application Server Cluster on the Application Tier .....	6-12
6.3.5.1	Creating the DCM-Managed OracleAS Cluster.....	6-12
6.3.5.2	Joining Application Server Instances to the DCM-Managed OracleAS Cluster .....	6-12

6.3.6	Modifying the Oracle Enterprise Manager 10g Application Server Control Console Welcome Page .....	6-13
6.4	Installing and Configuring the Web Tier .....	6-13
6.4.1	Installing the Oracle HTTP Servers on WEBHOST1 and WEBHOST2.....	6-13
6.5	Configuring the Manually Managed Oracle Application Server Cluster.....	6-15
6.6	Configuring the Oracle HTTP Server with the Load Balancing Router .....	6-16
6.7	Configuring OC4J Routing .....	6-17
6.8	Configuring Application Authentication and Authorization .....	6-18

## 7 Installing and Configuring the myPortalCompany Application Infrastructure

7.1	Installing the Metadata Repository for the Application Infrastructure.....	7-2
7.1.1	Installing the Metadata Repository in a Database Using Raw Devices.....	7-4
7.1.2	Installing the Metadata Repository in an Oracle Cluster File System (OCFS) .....	7-7
7.2	Configuring the Load Balancing Router or Proxy Server .....	7-11
7.3	Installing the Application Tier .....	7-13
7.3.1	Installing the First Application Server on APPHOST1 .....	7-14
7.3.2	Configuring Load Balancing and Monitoring .....	7-23
7.3.3	Configuring the First Application Server on APPHOST1 .....	7-24
7.3.3.1	Executing the SSL Configuration Tool on APPHOST1.....	7-26
7.3.3.2	Re-Setting the Oracle Enterprise Manager 10g Link .....	7-28
7.3.3.3	Configuring the Portal Tools Providers on APPHOST1.....	7-28
7.3.3.4	Creating an Indirect Password .....	7-32
7.3.3.5	Re-registering mod_osso on APPHOST1.....	7-34
7.3.3.6	Verifying Connectivity for Invalidation Messages from the Database to the OracleAS Web Cache on APPHOST1 through the Load Balancing Router	7-36
7.3.3.7	Enabling Monitoring of the Load Balancing Router's OracleAS Portal Host and Port Settings .....	7-36
7.3.3.8	Testing the Configuration on APPHOST1 .....	7-39
7.3.4	Installing the Second Application Server on APPHOST2 .....	7-40
7.3.5	Configuring the Second Application Server on APPHOST2 .....	7-48
7.3.5.1	Enabling Portal on APPHOST2 .....	7-49
7.3.5.2	Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST2 .....	7-50
7.3.5.3	Configuring the Parallel Page Engine Loop-Back with the Load Balancing Router on APPHOST2 .....	7-53
7.3.5.4	Modifying the Portal Dependency Settings (iasconfig.xml) File on APPHOST2 .....	7-55
7.3.5.5	Configuring the Portal Tools Providers on APPHOST2.....	7-56
7.3.5.6	Re-registering mod_osso on APPHOST2.....	7-57
7.3.6	Configuring OracleAS Web Cache Clusters .....	7-59
7.3.7	Configuring Load Balancing and Monitoring .....	7-63
7.3.8	Enabling Session Binding on OracleAS Web Cache Clusters .....	7-63
7.3.9	Modifying the Oracle Application Server Welcome Page.....	7-65
7.3.10	Registering Web Providers or Provider Groups Exposed over SSL (Optional) .....	7-66
7.3.11	Enabling the Federated Portal Adapter for SSL (Optional).....	7-67
7.3.12	Registering OracleAS Portal as an Oracle Ultra Search Content Source (Optional)	7-69
7.3.12.1	Enabling Oracle Ultra Search Access.....	7-69

7.3.12.2	Registering OracleAS Portal as an Oracle Ultra Search Content Source.....	7-70
7.4	Testing the Application Server Tier .....	7-73
7.5	Configuring Custom Java Portal Development Kit (JPDK) Providers .....	7-76
7.5.1	Deploying Custom JPDK Providers.....	7-77
7.5.2	Configuring Manually Managed Oracle Application Server Clusters for Session State Replication in OC4J_JPDK Applications .....	7-77
7.5.2.1	Configuring State Replication in the OC4J Instances.....	7-78
7.5.2.1.1	Configuring State Replication for Web Applications .....	7-78
7.5.2.2	Configure the J2EE Applications for Clustering.....	7-80
7.5.2.3	Configure the Oracle HTTP Server for Failover and Load Balancing .....	7-81
7.5.2.4	Disabling the JAZN Session Cache for UDDI Session Replication .....	7-83
7.6	Setting the OracleAS Single Sign-On Query Path URL for External Applications .....	7-85
7.6.1	Firewall Considerations for OracleAS Portal .....	7-87

## 8 Installing and Configuring the myBIFCompany Application Infrastructure

8.1	Installing the Metadata Repository for the Application Infrastructure.....	8-1
8.2	Configuring the Load Balancing Router or Proxy Server .....	8-1
8.3	Installing the Application Tier .....	8-2
8.3.1	Installing the First Application Server on APPHOST1 .....	8-2
8.3.2	Configuring the First Application Server on APPHOST1 .....	8-7
8.3.2.1	Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST1 .....	8-7
8.3.2.2	Re-registering mod_osso on APPHOST1.....	8-8
8.3.2.3	Configuring OracleAS Reports Services Server Targets in Oracle Enterprise Manager 10g Application Server Control Console .....	8-9
8.3.2.4	Testing the Configuration on APPHOST1 .....	8-9
8.3.3	Installing the Second Application Server on APPHOST2 .....	8-10
8.3.4	Configuring the Second Application Server on APPHOST2 .....	8-10
8.3.4.1	Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST2 .....	8-10
8.3.4.2	Re-registering mod_osso on APPHOST2.....	8-11
8.3.4.3	Configuring OracleAS Reports Services Server Targets in Oracle Enterprise Manager 10g Application Server Control Console .....	8-11
8.3.5	Configuring OracleAS Web Cache Clusters .....	8-12
8.3.6	Selecting the Secure Tunneling Protocol for Oracle Business Intelligence Discoverer Plus Deployment .....	8-12
8.3.7	Completing the Configuration.....	8-12
8.3.8	Managing Connection Availability for OracleAS Reports Services.....	8-13
8.3.9	Configuring Session State Replication in OC4J Instances.....	8-13
8.3.10	Modifying the Oracle Enterprise Manager 10g Application Server Control Console Welcome Page .....	8-13
8.3.11	Updating Host and Port Entries in OC4J_BI_Forms .....	8-13
8.4	Testing the Application Server Tier .....	8-14
8.5	Configuring OracleAS Portal in Business Intelligence and Forms.....	8-14

## 9 Implementing Architecture Variants

9.1	Configuring a Dedicated Intranet and Internet for OracleAS Portal .....	9-1
9.1.1	Installing the Infrastructure and External Middle Tier Instances .....	9-4

9.1.2	Installing the First Internal Middle Tier on APPHOST3.....	9-4
9.1.3	Installing the Second Internal Middle Tier on APPHOST4.....	9-6
9.1.4	Configuring an OracleAS Web Cache Invalidation-only Cluster.....	9-6
9.1.4.1	Preparing the Network Environment for the OracleAS Web Cache Invalidation-only Cluster .....	9-6
9.1.4.2	Configuring the Caches .....	9-7
9.1.4.2.1	Adding Caches to the Invalidation-Only Cluster.....	9-8
9.1.4.3	Disabling External to Internal Communication Through the Firewall .....	9-9
9.1.5	Configuring the First Internal Middle Tier on APPHOST3 for Load Balancing Router Access .....	9-9
9.1.6	Configuring the Second Internal Middle Tier on APPHOST4 for Load Balancing Router Access .....	9-15
9.1.7	Registering the Internal Middle Tier as a Partner Application.....	9-20
9.1.8	Updating the Default JPKD Instance URL and Seeded Provider Group URLs ....	9-21
9.1.9	Configuring OracleAS Portal Invalidation Messages .....	9-21
9.1.9.1	Verifying the OracleAS Web Cache Invalidation Messages Configuration ....	9-22
9.1.10	Configuring the OracleAS Portal Schema in the OracleAS Metadata Repository ..	9-22
9.1.11	Modifying the Oracle Text Base Search URL.....	9-23
9.1.12	Enabling Session Binding on OracleAS Web Cache .....	9-23
9.1.13	Configuring the Oracle Drive WebDAV Client .....	9-24
9.1.14	Validating the Completed Configuration .....	9-25
9.2	Configuring a Reverse Proxy for OracleAS Portal and OracleAS Single Sign-On.....	9-25
9.2.1	Install and Configure the Proxy Server .....	9-27
9.2.1.1	Configuring OracleAS Web Cache as a Reverse Proxy .....	9-27
9.2.1.1.1	Installing OracleAS Web Cache on the Proxy Computer.....	9-28
9.2.1.1.2	Applying the OracleAS Web Cache Patch.....	9-28
9.2.1.1.3	Creating Wallets for the Reverse Proxy Servers .....	9-28
9.2.1.1.4	Configuring the OracleAS Web Cache Listen Port .....	9-30
9.2.1.1.5	Configuring Sites, the Origin Server, and Site-to-Server Mappings.....	9-31
9.2.1.2	Configuring the Oracle HTTP Server as a Reverse Proxy .....	9-32
9.2.1.2.1	Using the ProxyPass, ProxyPassReverse, and ProxyPreserveHost Directives .....	9-32
9.2.1.2.2	Using the RewriteRule Directive.....	9-34
9.2.1.2.3	Using the X-FORWARDED-HOST Value with Apache v. 1.3 mod_proxy	9-35
9.2.1.3	Configuring Internet Information Services as a Reverse Proxy .....	9-36
9.2.1.3.1	Installing the Oracle Application Server Proxy Plug-in .....	9-37
9.2.1.3.2	Configuring the Oracle Application Server Proxy Plug-in .....	9-37
9.2.1.3.3	Configuring the IIS Listener to Use the Oracle Application Server Proxy Plug-in .....	9-38
9.2.2	Testing the OracleAS Single Sign-On Connection .....	9-39
9.2.3	Configuring OracleAS Single Sign-On to Use a Reverse Proxy.....	9-39
9.2.3.1	Ensuring that IP Checking is Off.....	9-39
9.2.3.2	Executing the ssocfg Script.....	9-40
9.2.3.3	Updating the targets.xml File .....	9-40
9.2.3.4	Updating the httpd.conf File.....	9-40
9.2.3.5	Updating Oracle Internet Directory with the Operation URL.....	9-41
9.2.3.6	Registering mod_osso to Use the Proxy Host Name .....	9-41
9.2.3.7	Updating the Single Sign-On Configuration .....	9-43

9.2.4	Validating the OracleAS Single Sign-On Configuration.....	9-43
9.2.5	Testing the OracleAS Portal Connection.....	9-43
9.2.6	Configuring OracleAS Portal for a Reverse Proxy.....	9-44
9.2.6.1	Ensuring Validity of Self-Referential URLs Rendered on OracleAS Portal Pages .....	9-44
9.2.6.2	Configuring Loopback Communication to the Internal Server.....	9-45
9.2.6.3	Specifying the OracleAS Portal Published Address and Protocol .....	9-45
9.2.6.4	Configuring the Parallel Page Engine Loop-Back with the Load Balancing Router on APPHOST1 .....	9-47
9.2.6.5	Configuring OracleAS Web Cache with the Reverse Proxy Server on APPHOST1 .....	9-47
9.2.6.6	Configuring Seeded Providers and Locally Hosted Web Providers .....	9-49
9.2.6.7	Registering the Domain Name .....	9-50
9.2.6.8	Re-registering mod_osso on APPHOST1.....	9-50
9.2.6.9	Augmenting the Parallel Page Engine x509certfile for Web Providers Exposed Over SSL (Optional) .....	9-51
9.2.6.10	Registering Web Providers or Provider Groups Exposed over SSL (Optional) .....	9-51
9.2.6.11	Enabling the Federated Portal Adapter for SSL (Optional) .....	9-51
9.2.6.12	Registering OracleAS Portal as an Oracle Ultra Search Content Source (Optional) .....	9-51
9.2.6.13	Using Oracle HTTP Server 1.3 as a Reverse Proxy for OracleAS Portal .....	9-51
9.2.7	Validating the OracleAS Portal Configuration.....	9-52
9.3	Configuring J2EE and Web Cache on the Web Tier .....	9-52
9.3.1	Installing and Configuring the Security Infrastructure .....	9-52
9.3.2	Installing and Configuring the Application Tier .....	9-52
9.3.2.1	A Note About Port Assignments for the Oracle Application Server File-Based Farm .....	9-53
9.3.3	Installing and Configuring the Web Tier .....	9-54
9.3.3.1	Installing the Web Tier Application Servers on WEBHOST1 and WEBHOST2 .....	9-54

## A Sample Configurations for Load Balancers

A.1	Test Network Configuration .....	A-1
A.1.1	Network Subnets in the Test Configuration .....	A-2
A.1.2	Hardware in the Test Configuration.....	A-3
A.1.3	Configuration of Load Balancers and Firewalls for Oracle Application Server Component High Availability .....	A-3
A.1.3.1	OracleAS Portal Communication.....	A-3
A.2	F5 Big IP Application Switch (Software Version 4.5 PTF.5) .....	A-4
A.2.1	Subnets for the Big IP Configuration .....	A-4
A.2.2	Servers/Nodes for the Big IP Configuration .....	A-5
A.2.3	Pools for the Big IP Configuration .....	A-5
A.2.4	Virtual Servers (VIPs) for the Big IP Configuration .....	A-5
A.2.5	Load Balancing Method for the Big IP Configuration.....	A-6
A.2.6	Health Monitors for the Big IP Configuration.....	A-6
A.2.6.1	OracleAS Single Sign-On.....	A-6
A.2.6.2	Middle Tier Components .....	A-6
A.2.6.3	OracleAS Web Cache Invalidation.....	A-6



A.2.6.4	Oracle Internet Directory LDAP.....	A-6
A.2.6.5	SSL Configuration .....	A-6
A.2.7	OracleAS Portal Configuration Notes for Big IP.....	A-7
A.2.8	OracleAS Wireless Configuration Notes for Big IP .....	A-7
A.2.9	OracleAS Web Cache Configuration Notes for Big IP .....	A-8
A.3	Cisco CSM 3.1(2) .....	A-8
A.3.1	Subnets for the Cisco CSM 3.1(2) Configuration .....	A-8
A.3.2	Servers/Nodes for the Cisco CSM 3.1(2) Configuration.....	A-8
A.3.3	VLANs for the Cisco CSM 3.1(2) Configuration .....	A-8
A.3.4	Server Farms for the Cisco CSM 3.1(2) Configuration .....	A-8
A.3.5	Virtual Servers (VIPs) for the Cisco CSM 3.1(2) Configuration .....	A-9
A.3.5.1	Virtual Servers for Outside Traffic Access to Server Farms.....	A-9
A.3.5.2	Sticky Configuration .....	A-10
A.3.5.3	Virtual Servers for HTTP Request Forwarding From the SSL Accelerator.....	A-10
A.3.5.4	Virtual Servers for Traffic from VLAN for Parallel Page Engine Requests .....	A-10
A.3.6	Test Configuration: Cisco CSM 3.1(2) .....	A-11
A.4	Foundry Server Iron v08.1.00cT24.....	A-16
A.4.1	Subnets for the Foundry Server Iron v08.1.00cT24 Configuration .....	A-16
A.4.2	Servers/Nodes for the Foundry Server Iron v08.1.00cT24 Configuration.....	A-17
A.4.3	Real Servers for the Foundry Server Iron v08.1.00cT24 Configuration .....	A-17
A.4.4	OracleAS Portal Configuration Notes for Foundry Server Iron v08.1.00cT24.....	A-18
A.4.5	OracleAS Wireless Configuration Notes for Foundry Server Iron v08.1.00cT24 ....	A-18
A.4.6	Test Configuration: Foundry Server Iron v08.1.00cT24 .....	A-18
A.5	Nortel Alteon 2424 SSL (Software Version 20.2.2.1) .....	A-21
A.5.1	Subnets for the Nortel Alteon 2424 SSL (Software Version 20.2.2.1) Configuration .....	A-21
A.5.2	Servers/Nodes for the Nortel Alteon 2424 SSL (Software Version 20.2.2.1) Configuration .....	A-21
A.5.3	Real Servers for the Nortel Alteon 2424 SSL (Software Version 20.2.2.1) Configuration .....	A-21
A.5.4	Groups for the Nortel Alteon 2424 SSL (Software Version 20.2.2.1) Configuration .....	A-22
A.5.5	Virtual IP Addresses for Nortel Alteon 2424 SSL (Software Version 20.2.2.1) .....	A-22
A.5.6	Additional Server Configuration for Nortel Alteon 2424 SSL (Software Version 20.2.2.1) .....	A-22
A.5.7	OracleAS Portal Configuration Notes for Nortel Alteon 2424 SSL (Software Version 20.2.2.1) .....	A-23
A.5.8	OracleAS Wireless Configuration Notes for Nortel Alteon 2424 SSL (Software Version 20.2.2.1) .....	A-23
A.5.9	Test Configuration: Nortel Alteon 2424 SSL (Software Version 20.2.2.1).....	A-23
A.6	Radware Web Server Director NP with SynApps 7.50.05 .....	A-31
A.6.1	Subnets for the Radware Web Server Director NP Configuration .....	A-31
A.6.2	Servers/Nodes for the Radware Web Server Director NP Configuration.....	A-31
A.6.3	Farms for the Radware Web Server Director NP Configuration.....	A-31
A.6.4	Servers for the Radware Web Server Director NP Configuration.....	A-31
A.6.5	Additional Server Configuration for the Radware Web Server Director NP .....	A-32
A.6.6	Super Farms for the Radware Web Server Director NP Configuration .....	A-32
A.6.7	Load Balancing Method for the Radware Web Server Director NP Configuration	A-32

A.6.8	OracleAS Portal Configuration Notes for Radware Web Server Director NP.....	A-33
A.6.9	OracleAS Wireless Configuration Notes for Radware Web Server Director NP ....	A-33
A.6.10	Test Configuration: Radware Web Server Director NP .....	A-33

## **B Sample Files and Values**

B.1	Metadata Repository Tablespaces .....	B-1
B.2	Tablespace Mapping to Raw Devices Sample File .....	B-1
B.3	Using the Static Ports Feature with Oracle Universal Installer .....	B-2
B.4	dads.conf File .....	B-3

## **Index**

---

---

# Preface

This preface describes the audience, contents and conventions used in the *Oracle Application Server Enterprise Deployment Guide*.

## Intended Audience

This guide is intended for system administrators who are responsible for installing and configuring Oracle Application Server.

## 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. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

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

### Accessibility of Code Examples in Documentation

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

### Accessibility of Links to External Web Sites in Documentation

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

### TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

## Related Documents

The following manuals in the Oracle Application Server documentation library provide additional information on the process of installing and configuring the Enterprise Deployment architectures:

- *Oracle Application Server Concepts*
- *Oracle Application Server Administrator's Guide*
- *Oracle Application Server Installation Guide*
- *Oracle Internet Directory Administrator's Guide*
- *Oracle Application Server Single Sign-On Administrator's Guide*
- *Oracle Application Server Portal Configuration Guide*
- *Oracle Application Server Web Cache Administrator's Guide*
- *Oracle Business Intelligence Discoverer Configuration Guide*

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.
/	A forward slash is used as a directory separator in paths, regardless of platform.

This chapter introduces Enterprise Deployment concepts, and summarizes the benefits provided by the Oracle Application Server Enterprise Deployment configurations described in other chapters of this guide. It contains the following topics:

[Section 1.1, "What is an Enterprise Deployment?"](#) on page 1-1

[Section 1.2, "Benefits of the Oracle Application Server Enterprise Deployment Configurations"](#) on page 1-2

## 1.1 What is an Enterprise Deployment?

An enterprise deployment is one of the Oracle Application Server configurations described in this guide, designed to support large-scale, mission-critical business software applications. The hardware and software in an Enterprise Deployment configuration delivers:

### **High quality service**

- The system workload is managed and balanced effectively
- Applications continue to operate when resources are added or removed
- System maintenance and unexpected failures cause zero downtime

### **Built-in Security**

- All incoming network traffic is received by the load balancing router on a single, secure port and directed to internal IP addresses within the firewall; inside the firewall, functional components are grouped within DMZs
- User accounts are provisioned and managed centrally
- Delegation of administration is performed consistently
- Security systems are integrated

### **Efficient software provisioning and management**

- Application distribution is simple
- Systems are managed and monitored as one logical unit in a central console
- Death detection and restart mechanisms ensure availability

## 1.2 Benefits of the Oracle Application Server Enterprise Deployment Configurations

The Oracle Application Server configurations shown in this guide are designed to ensure security of all transactions, maximize hardware resources, and provide a reliable, standards-compliant system for enterprise computing with a variety of applications. This section describes in detail the security and high availability benefits of the Oracle Application Server configurations and how they are achieved.

### 1.2.1 Built-in Security

The Enterprise Deployment architectures are secure because every functional group of software components is isolated in its own DMZ, and all traffic is restricted by protocol and port. The following characteristics ensure security at all needed levels, as well as a high level of compliance with standards:

- All external communication received on port 80 is redirected to port 443.
- Communication from external clients does not go beyond the Load Balancing Router level.
- No direct communication from the Load Balancing Router to the Data tier DMZ is allowed.
- Components are separated between DMZs on the Web Tier, Application Tier, and the Data Tier.
- Direct communication between two firewalls at any one time is prohibited.
- If a communication begins in one firewall zone, it must end in the next firewall zone.
- Oracle Internet Directory is isolated in the Data tier DMZ.
- Identity Management components such as Oracle HTTP Server, OracleAS Single Sign-On, and Oracle Delegated Administration Services are in the DMZ.
- All communication between components across DMZs is restricted by port and protocol, according to firewall rules.

### 1.2.2 High Availability

The Enterprise Deployment architectures are highly available, because each component or functional group of software components is replicated on a different computer, and configured for component-level high availability.

In the Web Server and Application Tiers, communication between components proceeds as follows:

1. The external Load Balancing Router does the following:
  - Receives end user requests on port 443, at the URL `portal.mycompany.com`, and balances the requests to one of the OracleAS Web Cache listeners on port 7777.
  - Receives invalidation messages from the Application Metadata Repository on port 9401, and balances the requests to one of the OracleAS Web Cache listeners on port 9401. In these cases, the Load Balancing Router functions as a proxy to receive internal requests on port 9401, but this port is not visible to external traffic.

- Receives web provider design time messages (and the init\_session call, for session based providers) on port 7777 from the Application Metadata Repository, and balances the requests to one of the OracleAS Web Cache listeners on port 7777. In these cases, the Load Balancing Router functions as a proxy to receive internal requests on port 7777, but this port is not visible to external traffic.

---

**Note:** Although OracleAS Web Cache is clustered, invalidation and web provider messages cannot be sent directly to a OracleAS Web Cache server, because if that particular OracleAS Web Cache is not functioning, then there is no way for the Metadata Repository to communicate with the other OracleAS Web Cache instance. The Load Balancing Router's management of the invalidation and web provider messages provides component level high availability.

---

2. The Load Balancing Router balances the requests to one of the two OracleAS Web Cache servers on port 7777.
3. Each OracleAS Web Cache server receives the requests from the Load Balancing Router and passes them to one of the two OracleAS Portal Oracle HTTP Servers on port 7778.

---

**Note:** Since all of the OracleAS Portal sessions are stateless, these requests can be routed from any OracleAS Web Cache server to any OracleAS Portal Oracle HTTP Server, and vice versa. However, if you are using Web Clipping functionality in OracleAS Portal, then you will need to enable stateful binding (as described in [Section 7.3.8, "Enabling Session Binding on OracleAS Web Cache Clusters"](#) on page 7-63.)

---

4. The OracleAS Portal Parallel Page Engine also loops back to the Load Balancing Router (through the internal Network Address Translation port) to reach Portal Services to get the metadata information to construct the page. The Load Balancing Router is configured to handle Parallel Page Engine loop back calls, and load balances them to one of the Webcache listeners on port 7777.

---

**Note:** The Parallel Page Engine constructs OracleAS Portal pages based on metadata in the Metadata Repository. To read the metadata, it loops back to Portal Services through the local OracleAS Web Cache instance. However, if that OracleAS Web Cache instance is down, there is no way for the Parallel Page Engine to reach Portal Services or the other OracleAS Web Cache instance. If Parallel Page Engine loops back to the Load Balancing Router, the Load Balancing Router can balance requests to Portal Services through the surviving OracleAS Web Cache instance, which can still balance the requests to the Oracle HTTP Server on the first middle tier. This exemplifies component level high availability and intelligent routing for efficient resource utilization.

---

5. When the request goes to portal.mycompany.com, OracleAS Portal determines whether the request is authenticated with OracleAS Single Sign-On; if not, it will redirect the request to the OracleAS Single Sign-On URL, login.mycompany.com.

In the Identity Management Tier, communication proceeds as follows:

1. OracleAS Single Sign-On receives a user request at login.mycompany.com.
2. OracleAS Single Sign-On authenticates the credentials with one of two Oracle Internet Directory instances, through the internal Load Balancing Router that is configured to manage Oracle Internet Directory traffic.

---

**Note:** Two Oracle Internet Directory instances on different computers are using the same Metadata Repository. If the Identity Management components (OracleAS Single Sign-On, Oracle Delegated Administration Services) directly communicate with an Oracle Internet Directory instance, and that instance stops working, then there would be no way for the Identity Management components to redirect the traffic to the surviving Oracle Internet Directory instance. Thus the Load Balancing Router ensures high availability in re-routing traffic from a failed Oracle Internet Directory instance to a surviving instance.

---



## Selecting a Deployment Architecture

This chapter introduces Oracle Application Server installation types and architectures and the nomenclature used in this guide to describe the Enterprise Deployment architectures.

### 2.1 Creating Solutions with Oracle Application Server

Oracle Application Server 10g Release 2 (10.1.2) is a complete, fully integrated product that delivers a wide range of solutions to business and technology challenges. This guide presents a subset of these, in the form of recommendations based on deployments by Oracle customers.

This guide provides installation and configuration steps for the following installation types and selections:

1. Oracle Identity Management
2. J2EE
3. OracleAS Portal
4. Business Intelligence and Forms

For complete descriptions of the components comprising these installation types and selections, see the *Oracle Application Server Concepts* guide.

### 2.2 Enterprise Deployment Nomenclature

The naming convention for the components and computers is established in the architecture diagrams and is used throughout this guide. Server names and their related URLs and IP addresses are provided in [Table 2-1](#). The external load balancer nomenclature is provided in [Table 2-2](#).

**Table 2-1 Server Name, URL and IP Address Reference**

Description	Name	URL	IP Address
Servers with 2-node Real Application Clusters database for Security Metadata Repository	INFRADBHOST1	infradbhost1.mycompany.com	xxx.xxx.xxx.225
	INFRADBHOST2	infradbhost2.mycompany.com	xxx.xxx.xxx.226
Servers with 2-node Real Application Clusters database for Application Metadata Repository	APPDBHOST1	appdbhost1.mycompany.com	xxx.xxx.xxx.227
	APPDBHOST2	appdbhost2.mycompany.com	xxx.xxx.xxx.228
Oracle Internet Directory servers	OIDHOST1	oidhost1.mycompany.com	xxx.xxx.xxx.229
	OIDHOST2	oidhost2.mycompany.com	xxx.xxx.xxx.230

**Table 2–1 (Cont.) Server Name, URL and IP Address Reference**

Description	Name	URL	IP Address
Identity Management servers	IDMHOST1	idmhost1.mycompany.com	xxx.xxx.xxx.231
	IDMHOST2	idmhost2.mycompany.com	xxx.xxx.xxx.232
Application middle tier servers	APPHOST1	apphost1.mycompany.com	xxx.xxx.xxx.233
	APPHOST2	apphost2.mycompany.com	xxx.xxx.xxx.234
Web tier servers (myJ2EECompany)	WEBHOST1	webhost1.mycompany.com	xxx.xxx.xxx.235
	WEBHOST2	webhost2.mycompany.com	xxx.xxx.xxx.236

**Table 2–2 External Load Balancer Name, URL and IP Address Reference**

Description	URL	IP Address
Virtual IP Addresses	portal.mycompany.com:443	xxx.yyy.zzz.220
	login.mycompany.com:443	xxx.yyy.zzz.220
		xxx.yyy.zzz.222
		xxx.yyy.zzz.222
Virtual IP Address (myJ2EECompany)	myapp.mycompany.com:443	xxx.yyy.zzz.220
Failover Virtual IP Addresses	portal.mycompany.com:443	xxx.yyy.zzz.221
	login.mycompany.com:443	xxx.yyy.zzz.223
Internal Load Balancer for LDAP traffic	oid.mycompany.com:389/636	xxx.yyy.zzz.12
Failover Virtual IP Addresses (VIPs)	oid.mycompany.com:389/636	xxx.yyy.zzz.13
Internal Ports: Source Network Address Translation (SNAT) for VIP1	portal.mycompany.com:7777	xxx.yyy.zzz.14
	portal.mycompany.com:4001	xxx.yyy.zzz.15

## 2.3 Understanding the Enterprise Deployment Architectures

This section briefly describes the Enterprise Deployment architectures in this guide, including minimum hardware requirements and a diagram of the architecture.

### 2.3.1 myJ2EE

Figure 2–1 shows the enterprise deployment architecture for any J2EE application that uses JAZN LDAP for user authentication. If you need to use the Single Sign-On Server for authentication for J2EE applications, you should use the Standard Enterprise Deployment for Portal Applications: myPortalCompany.com described in Section 2.3.2, "myPortal".

For certain types of J2EE applications, such as JMS-based or EJB-based applications, there may be additional variants to these architectures. Refer to the *Oracle Application Server Containers for J2EE Services Guide* and *Oracle Application Server Containers for J2EE Enterprise JavaBeans Developer's Guide* for more information on these variants.

The servers in the myJ2EECompany system are grouped into tiers as follows:

- **Web Tier** — WEBHOST1 and WEBHOST2, with Oracle HTTP Server installed.
- **Application Tier** — APPHOST1 and APPHOST2, with Oracle Application Server Containers for J2EE installed, and multiple OC4J instances with applications deployed.

- **Data Tier** — OIDHOST1 and OIDHOST2, with Oracle Internet Directory installed, and INFRADBHOST1 and INFRADBHOST2, the two-node Real Application Clusters database.

Table 2–3, Table 2–4 and Table 2–5 identify the basic, minimum hardware requirements for the servers in the myJ2EECompany system on Windows, Linux and Solaris operating systems, respectively. The memory figures represent the memory required to install and run Oracle Application Server; however, for most production sites, you should configure at least 1 GB of physical memory.

For detailed requirements, or for requirements for a platform other than these, see the *Oracle Application Server Installation Guide* for the platform you are using.

**Table 2–3 myJ2EECompany Hardware Requirements (Windows)**

Server	Processor	Disk	Memory	TMP Directory	Swap
WEBHOST and APPHOST	300 MHz or higher Intel Pentium processor recommended	400 MB	512 MB	55 MB to run the installer; 256 MB needed for some installation types	512 MB
OIDHOST and INFRADBHOST	300 MHz or higher Intel Pentium processor recommended	2.5 GB	1 GB	55 MB to run the installer; 256 MB needed for some installation types	1 GB

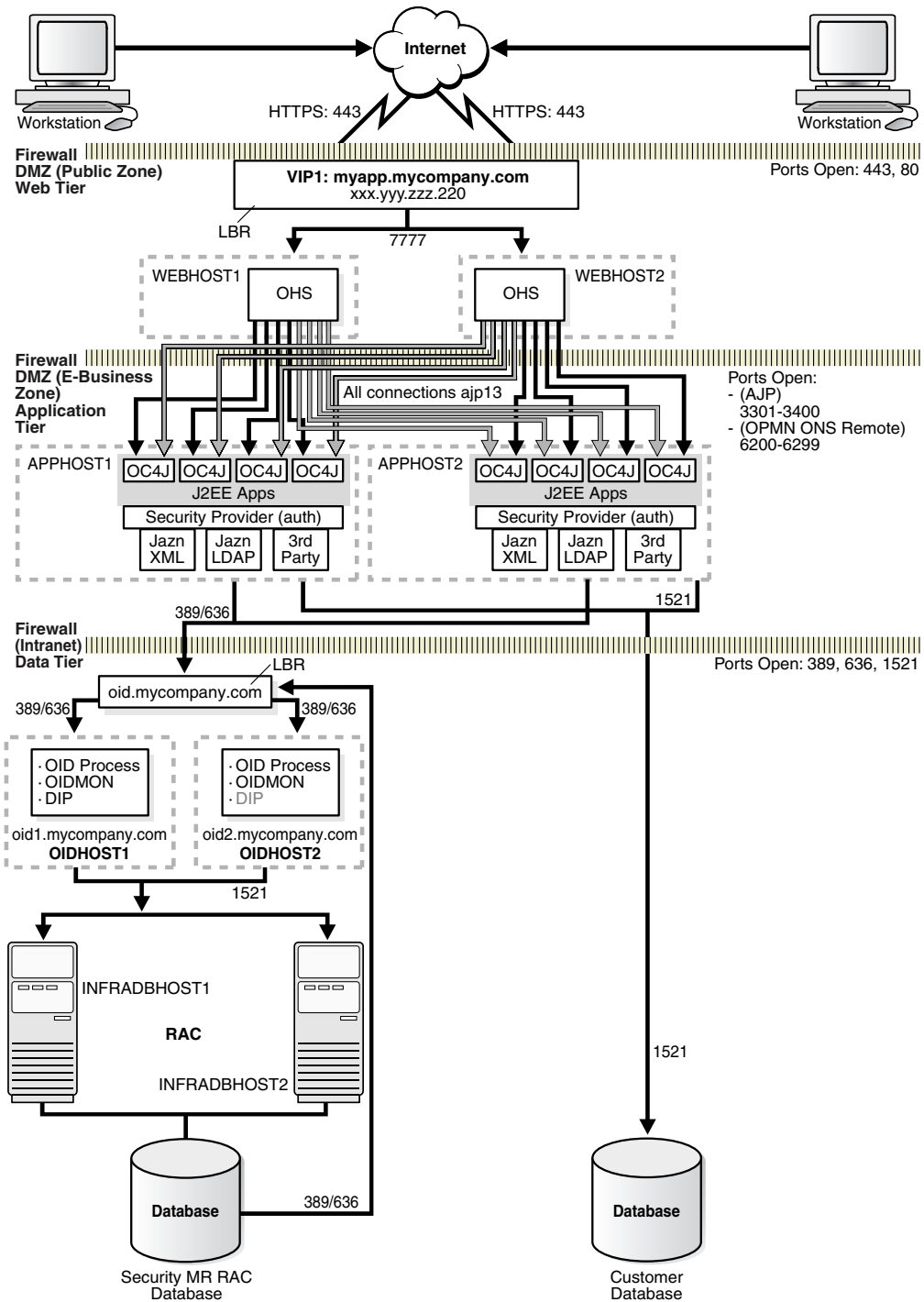
**Table 2–4 myJ2EECompany Hardware Requirements (Linux)**

Server	Processor	Disk	Memory	TMP Directory	Swap
WEBHOST and APPHOST	Pentium (32-bit), 450 MHz or greater	520 MB	512 MB	400 MB	1.5 GB
OIDHOST and INFRADBHOST	Pentium (32-bit), 450 MHz or greater	2.5 GB	1 GB	400 MB	1.5 GB

**Table 2–5 myJ2EECompany Hardware Requirements (Solaris)**

Server	Processor	Disk	Memory	TMP Directory	Swap
WEBHOST and APPHOST	450 MHz or greater; Oracle recommends a multiple CPU computer	750 MB	512 MB	250 MB	1.5 GB
OIDHOST	450 MHz or greater; Oracle recommends a multiple CPU computer	1.54 GB	1 GB	250 MB	1.5 GB
INFRADBHOST	450 MHz or greater; Oracle recommends a multiple CPU computer	3.93 GB	1 GB	250 MB	1.5 GB

**Figure 2–1 Enterprise Deployment Architecture for myJ2EECompany.com**



## 2.3.2 myPortal

Figure 2–2 shows the enterprise deployment architecture for OracleAS Portal applications.

The servers in the myPortalCompany system are grouped into tiers as follows:

- **Application Tier** — APPHOST1 and APPHOST2
- **Identity Management Tier** — IDMHOST1 and IDMHOST2
- **Data Tier** — OIDHOST1 and OIDHOST2, with Oracle Internet Directory installed, and INFRADBHOST1 and INFRADBHOST2, the two-node Real Application Clusters database. APPDBHOST1 and APPDBHOST2 contain the OracleAS Portal application metadata repository.

Table 2–6, Table 2–7 and Table 2–8 identify the basic, minimum hardware requirements for the servers in the myPortalCompany system on Windows, Linux and Solaris operating systems, respectively. The memory figures represent the memory required to install and run Oracle Application Server; however, for most production sites, you should configure at least 1 GB of physical memory. Table 2–9 describes the servers used in the Oracle test environment for myPortalCompany.

For detailed requirements, or for requirements for a platform other than these, see the *Oracle Application Server Installation Guide* for the platform you are using.

**Table 2–6 myPortalCompany Hardware Requirements (Windows)**

Server	Processor	Disk	Memory	TMP Directory	Swap
APPHOST, IDMHOST, OIDHOST, INFRADBHOST, and APPDBHOST	300 MHz or higher Intel Pentium processor recommended	2.5 GB	1 GB	55 MB to run the installer; 256 MB needed for some installation types	1 GB

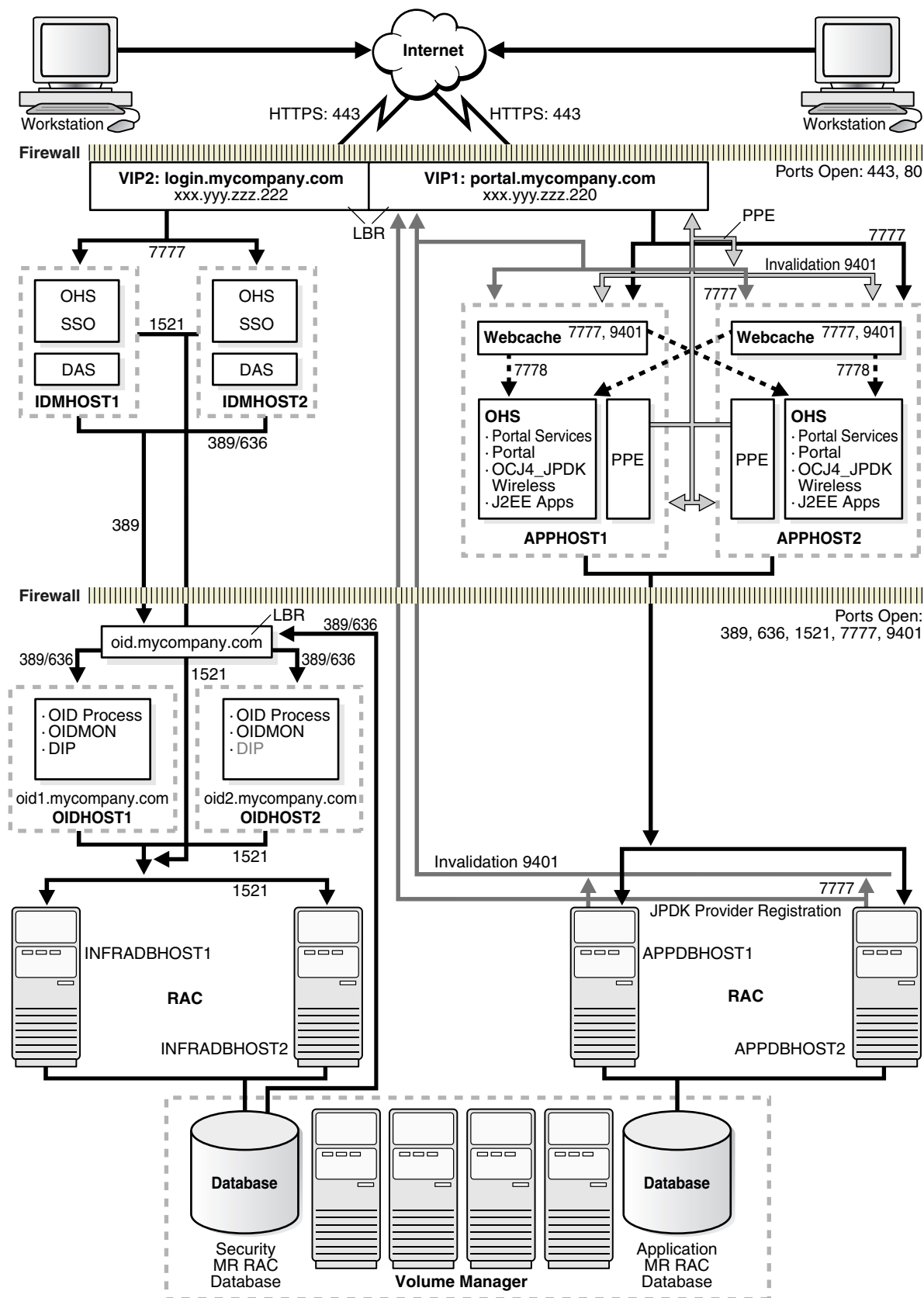
**Table 2–7 myPortalCompany Hardware Requirements (Linux)**

Server	Processor	Disk	Memory	TMP Directory	Swap
APPHOST, IDMHOST, OIDHOST, INFRADBHOST, and APPDBHOST	Pentium (32-bit), 450 MHz or greater	2.5 GB	1 GB	400 MB	1.5 GB

**Table 2–8 myPortalCompany Hardware Requirements (Solaris)**

Server	Processor	Disk	Memory	TMP Directory	Swap
APPHOST, IDMHOST, OIDHOST, INFRADBHOST, and APPDBHOST	450 MHz or greater; Oracle recommends a multiple CPU computer	750 MB	512 MB	250 MB	1.5 GB

**Figure 2–2 Enterprise Deployment Architecture for myPortalCompany.com**



**Table 2–9** *myPortalCompany Servers in Oracle Test Environment*

Server	Platform	Virtual Memory	TMP	RAM	CPU
INFRADBHOST1 and INFRADBHOST2	Windows 2000	2 GB	Not applicable	2 GB	3 GHz
OIDHOST1	Windows 2000	3 GB	Not applicable	2 GB	3 GHz
OIDHOST2	Windows 2000	2GB	Not applicable	2 GB	3 GHz
IDMHOST1 and IDMHOST2	Windows 2000	2 GB	Not applicable	3.75 GB	3 GHz
APPDBHOST1 and APPHOST2	Red Hat Linux 2.1	2 GB	2.5 GB	6 GB	4 CPU, 3 GHz
APPHOST1 and APPHOST2	Windows 2000	1.6 GB	Not applicable	3.75 GB	3 GHz

### 2.3.3 myBIFCompany

Figure 2–3 shows the enterprise deployment architecture for Oracle Business Intelligence and Forms applications.

The servers in the myBIFCompany system are grouped into tiers as follows:

- **Application Tier** — APPHOST1 and APPHOST2
- **Identity Management Tier** — IDMHOST1 and IDMHOST2
- **Data Tier** — OIDHOST1 and OIDHOST2, with Oracle Internet Directory installed, and INFRADBHOST1 and INFRADBHOST2, the two-node Real Application Clusters database. APPDBHOST1 and APPDBHOST2 contain the OracleAS Portal application metadata repository.

Table 2–10, Table 2–11 and Table 2–12 identify the basic, minimum hardware requirements for the servers in the myBIFCompany system on Windows, Linux and Solaris operating systems, respectively. The memory figures represent the memory required to install and run Oracle Application Server; however, for most production sites, you should configure at least 1 GB of physical memory. Table 2–13 describes the servers used in the Oracle test environment for myBIFCompany.

For detailed requirements, or for requirements for a platform other than these, see the *Oracle Application Server Installation Guide* for the platform you are using.

**Table 2–10 myBIFCompany Hardware Requirements (Windows)**

Server	Processor	Disk	Memory	TMP Directory	Swap
APPHOST, IDMHOST, OIDHOST, INFRADBHOST, and APPDBHOST	300 MHz or higher Intel Pentium processor recommended	2.5 GB	1 GB	55 MB to run the installer; 256 MB needed for some installation types	1 GB

**Table 2–11 myBIFCompany Hardware Requirements (Linux)**

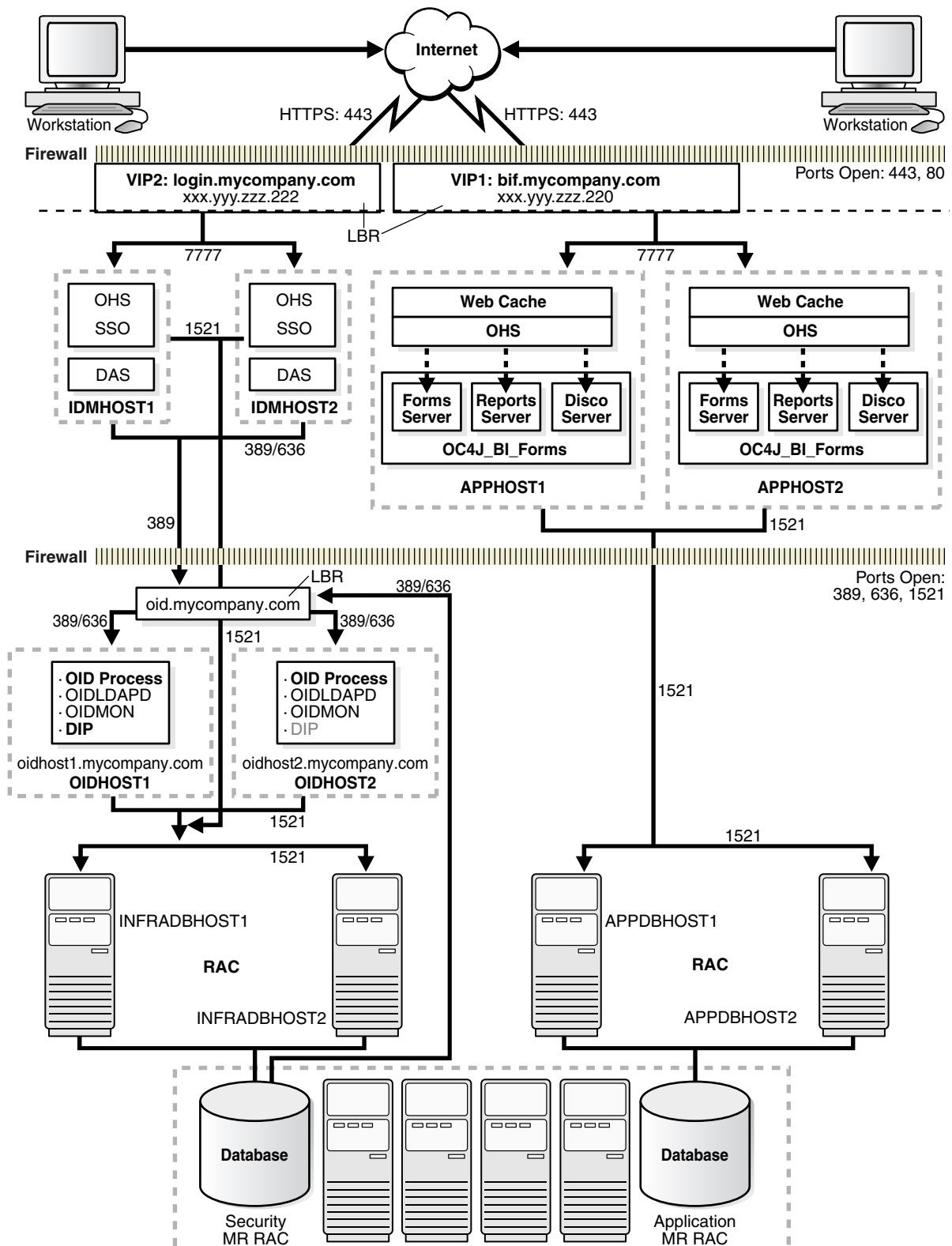
Server	Processor	Disk	Memory	TMP Directory	Swap
APPHOST, IDMHOST, OIDHOST, INFRADBHOST, and APPDBHOST	Pentium (32-bit), 450 MHz or greater	2.5 GB	1 GB	400 MB	1.5 GB

**Table 2–12 myBIFCompany Hardware Requirements (Solaris)**

Server	Processor	Disk	Memory	TMP Directory	Swap
APPHOST, IDMHOST, OIDHOST, INFRADBHOST and APPDBHOST	450 MHz or greater; Oracle recommends a multiple CPU computer	750 MB	512 MB	250 MB	1.5 GB



Figure 2-3 Enterprise Deployment Architecture for myBIFCompany.com



**Table 2–13 myBIFCompany Servers in Oracle Test Environment**

Server	Platform	Virtual Memory	TMP	RAM	CPU
INFRADBHOST1 and INFRADBHOST2	Windows 2000	2 GB	Not applicable	2 GB	3 GHz
OIDHOST1	Windows 2000	3 GB	Not applicable	2 GB	3 GHz
OIDHOST2	Windows 2000	2GB	Not applicable	2 GB	3 GHz
IDMHOST1 and IDMHOST2	Windows 2000	2 GB	Not applicable	3.75 GB	3 GHz
APPDBHOST1 and APPHOST2	Red Hat Linux 2.1	2 GB	2.5 GB	6 GB	4 CPU, 3 GHz
APPHOST1 and APPHOST2	Windows 2000	1.6 GB	Not applicable	3.75 GB	3 GHz

## 2.4 Understanding Deployment Variants

Figure 2–1, "Enterprise Deployment Architecture for myJ2EECompany.com", Figure 2–2, "Enterprise Deployment Architecture for myPortalCompany.com" and Figure 2–3, "Enterprise Deployment Architecture for myBIFCompany.com" show standard enterprise deployment architectures. Some characteristics of the standard enterprise deployment configuration are:

- A two-node Real Application Clusters (RAC) database on the Data Tier is used to provide high availability (multiple database instances access a shared database of data files).
- Oracle Internet Directory is installed on the Data Tier.
- OracleAS Single Sign-On (on the Identity Management tier [Figure 2–2](#)), or the Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider (on the Application Tier in [Figure 2–1](#)), is used for authentication and authorization.

Several variants exist for these and other elements of the enterprise deployment architectures. They are described in this section, categorized by the tier on which they are implemented (Data, Identity Management, Application, or Web). The variants enable you to achieve your deployment goals using fewer servers, different software, or alternative configurations.

### 2.4.1 Understanding Data Tier Variants

This section describes the variants for the Data Tier. The Data Tier is depicted in [Figure 2–2, "Enterprise Deployment Architecture for myPortalCompany.com"](#), and comprises the INFRADBHOST1 and INFRADBHOST2 computers.

#### 2.4.1.1 Using Multimaster Replication with Oracle Internet Directory

Multimaster replication is an Oracle Internet Directory software solution that ensures read and write access to Oracle Internet Directory at all times, if at least one of the computers in the system remains available. When a computer resumes functioning after unavailability, replication from the surviving computer resumes automatically and synchronizes the contents between the computers. In addition, changes made on one directory server instance are reflected on the second directory server instance.

Multimaster replication of Oracle Internet Directory differs from the standard configuration in that the Oracle Internet Directory server and the database are on the same computer, whereas in the standard configuration the first Oracle Internet Directory instance and a database instance occupy IDMHOST1 and INFRADBHOST1,

while the second Oracle Internet Directory instance and a database instance occupy IDMHOST2 and INFRADBHOST2. Thus, the replicated Oracle Internet Directory operates two fewer servers than the RAC configuration.

To implement multimaster replication in Oracle Internet Directory, follow the instructions in the *Oracle Internet Directory Administrator's Guide*, Oracle Internet Directory Replication Administration chapter, section titled "Installing and Configuring Multimaster Replication".

### 2.4.1.2 Using the Oracle Application Server Cold Failover Cluster (Identity Management) Solution

The OracleAS Cold Failover Cluster (Identity Management) solution is a hardware cluster comprising two computers. The computer that is actively executing an Infrastructure installation at any given time is called the primary (hot) node. If this node fails, the hardware cluster automatically diverts Infrastructure operations to the secondary (cold) node.

Each hardware cluster node is a standalone server that runs its own set of processes, but accesses a shared storage subsystem. The cluster can access the same storage, usually disks, from both nodes, but only the primary node has active access to the storage at any given time. If the primary node fails, the hardware cluster's software grants the secondary node access to the storage.

---

**Note:** For a detailed discussion of the OracleAS Cold Failover Cluster (Identity Management) solution, see the *Oracle Application Server High Availability Guide*

---

The OracleAS Cold Failover Cluster (Identity Management) solution differs from the standard configuration in the following ways:

- The Oracle Internet Directory server and the database are on the same computer, whereas in the standard configuration the first Oracle Internet Directory instance and a database instance occupy IDMHOST1 and INFRADBHOST1, while the second Oracle Internet Directory instance and a database instance occupy IDMHOST2 and INFRADBHOST2. Thus, the OracleAS Cold Failover Cluster (Identity Management) solution operates two fewer servers than the RAC configuration.
- In the event of node failure, clients will experience a brief interruption of service while the workload is diverted to the cold node.

**2.4.1.2.1 Implementing the OracleAS Cold Failover Cluster (Identity Management) Solution** To implement the OracleAS Cold Failover Cluster (Identity Management) solution:

1. Obtain and configure a hardware cluster.
2. Install and configure the Oracle Application Server instances on the cluster computers to use the OracleAS Cold Failover Cluster (Identity Management) solution. Follow the instructions in the *Oracle Application Server Installation Guide*, section 11.5, "Installing an OracleAS Cold Failover Cluster (Identity Management) Configuration".
3. Manage the OracleAS Cold Failover Cluster (Identity Management) solution, following the instructions from the *Oracle Application Server High Availability Guide*, section 6.3, "Managing Oracle Application Server Cold Failover Cluster (Identity Management)".

## 2.4.2 Understanding Identity Management Tier Variants

This section describes the variants for the Identity Management Tier. The Identity Management Tier is depicted in [Figure 2-2, "Enterprise Deployment Architecture for myPortalCompany.com"](#), and comprises the IDMHOST1 and IDMHOST2 computers.

### 2.4.2.1 Oracle Internet Directory: Data Tier or Identity Management Tier?

Oracle Internet Directory can be installed on the Identity Management Tier, along with OracleAS Single Sign-On and Oracle Delegated Administration Services. This is typical of configurations that provide a complete, local identity management system (Oracle Internet Directory and Oracle Application Server Single Sign-On) on one computer to applications located near that computer. See the *Oracle Identity Management Concepts and Deployment Planning Guide*, Chapter 3, "Oracle Identity Management Deployment Planning", section titled "Planning the Physical Network Topologies".

In the standard configuration, in which Oracle Internet Directory is installed on the Data Tier, Oracle Internet Directory and its metadata repository are behind a firewall, and is isolated from Internet traffic.

### 2.4.2.2 Oracle Internet Directory: AD/iPlanet Integration

Oracle Identity Management provides a set of components for integrating with other identity management environments, including various services and APIs, preconfigured directory connectivity solutions and standards support. For example, Oracle Identity Management allows for integration with various 3rd party directories, including Microsoft Active Directory and SunONE Directory Server.

By default, Oracle Directory Integration and Provisioning is installed as a component of Oracle Internet Directory. However, you can also install Oracle Directory Integration and Provisioning in a standalone installation. You should install a standalone instance of Oracle Directory Integration and Provisioning under the following circumstances:

- When you need Oracle Internet Directory to run on a separate host for performance reasons
- When the applications that you need to provision and synchronize required intensive processing
- You need to run multiple instances of Oracle Directory Integration and Provisioning for high availability

See the *Oracle Identity Management Integration Guide* for detailed information on configuration options.

### 2.4.2.3 Oracle Application Server Single Sign-On: Using Netegrity

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.

[http://www.oracle.com/technology/products/id\\_mgmt/partners/index.html](http://www.oracle.com/technology/products/id_mgmt/partners/index.html)

For example, Netegrity provides Siteminder Agent for Oracle Application Server. The agent delivers a mechanism to enable integration between heterogeneous, enterprise wide SiteMinder implementation with the OracleAS Single Sign-On environment. The agent provides enhanced security to protect Oracle Web-based resources, including session synchronization and revalidation of the user's SiteMinder session behind the DMZ in a trusted zone or corporate internal network prior to initiating the Oracle session.

For the current information on version, platform support and configuration guide, visit:

<http://www.netegrity.com>

#### 2.4.2.4 Oracle Application Server Single Sign-On: Windows Authentication

Windows native authentication is an authentication scheme for those who use Internet Explorer on Windows platforms. When this feature is enabled in OracleAS Single Sign-On, users log in to single sign-on partner applications automatically using Kerberos credentials obtained when the user logs in to a Windows computer.

Using the Simple Protected GSS-API Negotiation Protocol (SPNEGO), browsers that are Internet Explorer 5.0 and greater can automatically pass the user's Kerberos credentials to a Kerberos-enabled Web server when the server requests these credentials. The Web server can then decrypt the credentials and authenticate the user.

Before setting up Windows native authentication, you must first set up Active Directory (AD) Synchronization to Oracle Internet Directory. See the *Oracle Internet Directory Administrator's Guide* for instructions on how to do this.

### 2.4.3 Understanding Application Tier Variants

This section describes the variants for the Application Tier. The Application Tier is depicted in [Figure 2–1, "Enterprise Deployment Architecture for myJ2EECompany.com"](#), [Figure 2–2, "Enterprise Deployment Architecture for myPortalCompany.com"](#), and [Figure 2–3, "Enterprise Deployment Architecture for myBIFCompany.com"](#) and comprises the APPHOST1 and APPHOST2 computers.

#### 2.4.3.1 J2EE Applications: File Based or Database Repository?

An Oracle Application Server Farm is a collection of instances that share the same configuration management metadata repository. A farm can be either a Oracle Application Server File-Based Farm or Oracle Application Server Database-Based Farm.

Within these farm types, there are three types of metadata repository configuration: File-based (with standalone instance), File-based (with repository host instance) and Database:

- **File-based repository (standalone instance)** — Every instance includes a local file-based repository. In a standalone instance, this repository stores the configuration metadata for the instance. When an instance is part of an OracleAS Database-Based Farm or an OracleAS File-Based Farm, and the instance is not the repository host, the local file-based repository contains the Bill of Materials (BOM) that Distributed Configuration Management uses to validate that the instance is synchronized with the configuration metadata in the repository.
- **File-based repository (with repository host instance)** — When an instance is defined as the repository host for an OracleAS File-Based Farm, the repository for the instance contains the configuration metadata for all instances in the farm.

- **Database repository** - comprised of DCM schema. Storing the metadata repository in a database may be useful as part of a site's high availability and backup strategy. Using a database repository, the database serves as the repository host.

In all three metadata repository scenarios (database repository, file-based repository with a standalone instance, or file-based repository host instance), an instance always has a local file based repository. If the instance is not included in a farm, this is the sole storage for the configuration metadata for the instance.

The choice of database repository or file-based repository has a low impact on a system's availability. In case of repository failure or downtime, the J2EE cluster continues to operate. Only the distributed management features are unavailable during the repository downtime. [Table 2-14](#) compares repository types in light of operational considerations.

**Table 2-14 OracleAS File-Based Farm and OracleAS Database-Based Farm Comparison**

Consideration	Advantage
Number of computers in a farm	No known limitation for an OracleAS File-Based Farm or an OracleAS Database-Based Farm
Deployment frequency	Deployment is faster in an OracleAS File-Based Farm
Recovery for manageability	Recovery from a system failure is faster with OracleAS File-Based Farm
Reliability	High Availability features provided by the database (RAC, for example) are far superior to the OracleAS File-Based Farm
Rolling upgrade needs	There is less downtime for management involved in an OracleAS File-Based Farm rolling upgrade than in an OracleAS Database-Based Farm rolling upgrade

## 2.4.4 Understanding Web Server Tier Variants

This section describes the variants for the Web Server Tier. The Web Server Tier is depicted in [Figure 2-2, "Enterprise Deployment Architecture for myPortalCompany.com"](#), and comprises the APPHOST1 and APPHOST2 computers.

In [Figure 2-1, "Enterprise Deployment Architecture for myJ2EECompany.com"](#), the Web Server Tier comprises the WEBHOST1 and WEBHOST2 computers.

### 2.4.4.1 Oracle Application Server Web Cache Placement, Clustering and Deployment Considerations

OracleAS Web Cache is a content-aware server accelerator, or reverse proxy server, that improves the performance, scalability, and availability of Web sites that run on Oracle Application Server.

Oracle recommends configuring multiple instances of OracleAS Web Cache to run as members of a cache cluster. A cache cluster is a loosely coupled collection of cooperating OracleAS Web Cache cache instances that provide a single logical cache.

When deploying topologies described in this document, one variant is to place OracleAS Web Cache on a separate host. This is particularly useful in environments with large amounts of cacheable content. This architecture modification provides flexibility in choosing the number of computers to operate OracleAS Web Cache, as well as defining separate hardware profile for OracleAS Web Cache servers and J2EE or OracleAS Portal servers. Typically, a large amount of RAM and fast access to file storage are the most critical components in the performance of the OracleAS Web Cache server.

Another possibility is to place a firewall between OracleAS Web Cache and the Oracle HTTP Server; this would provide an additional layer of security.

---

**Note:** In an OracleAS Portal environment, specific configuration is needed to ensure that cache invalidation messages can reach, and be correctly routed to, the Web Server Tier.

---

For additional information on configuration variants with OracleAS Web Cache, see the *Oracle Application Server Web Cache Administrator's Guide*.

#### **2.4.4.2 Oracle HTTP Server: Forward and Reverse Proxies**

The architectures described in this guide can be deployed in environments with additional forward or reverse proxy servers.

Proxy scenarios change the way the clients' IP addresses are seen by the Oracle HTTP Server. This can be adjusted to better match Web applications' expectations by transferring the clients' IP addresses through proxies in additional HTTP headers and making the HTTP Server use the header values, either with explicit configuration or implicitly, by overall replacing the "physical" request connection information with the header values.

The Oracle HTTP Server and applications in an Oracle HTTP Server handle information about clients. Because clients are often identified by their IP addresses, scenarios in which reverse ("transparent") or forward ("normal") proxies are part of the whole system may require adjustments in how the client's IP addresses are seen by the Oracle HTTP Server.

For instructions on how to configure a reverse proxy using Oracle HTTP Server or the Internet Information Services (IIS) server, see [Section 9.2, "Configuring a Reverse Proxy for OracleAS Portal and OracleAS Single Sign-On"](#). For information on how to integrate OracleAS Web Cache with an additional proxy server, see the *Oracle Application Server Web Cache Administrator's Guide*.

#### **2.4.4.3 Oracle HTTP Server as a Standalone Web Server**

There are two ways to install Oracle HTTP Server on the Web Server Tier: as a standalone Web server, or as part of the J2EE and Web Cache installation type.

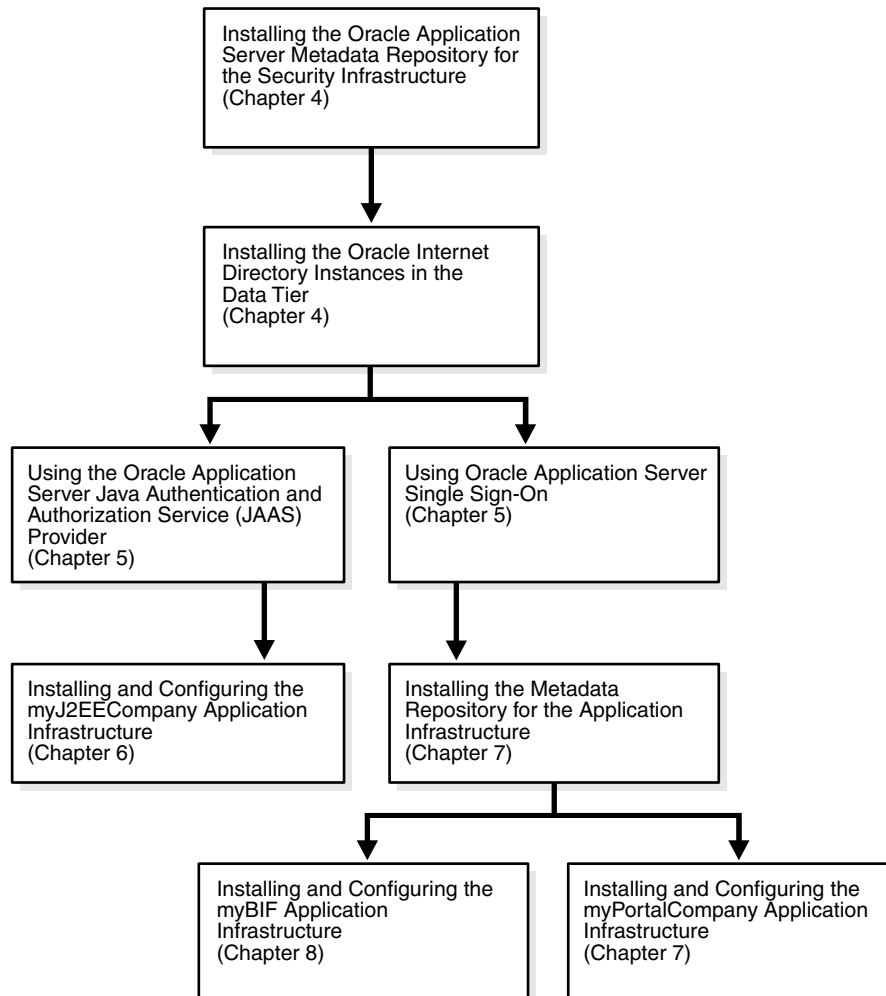
Some security plans discourage installation of any Java executables on the Web Server tier. For this reason, this guide presents the installation of the Oracle HTTP Server as a standalone Web server. The Oracle HTTP Server is managed by the `opmnctl` utility (invoked by the Start menu on Windows systems) instead of the Oracle Enterprise Manager 10g Application Server Control Console.

If it is acceptable or desirable to install all of the J2EE and Web Cache components on the Web Server tier, or you want to use the Oracle Enterprise Manager 10g Application Server Control Console to manage the Oracle HTTP Server, you may follow the instructions in [Section 9.3, "Configuring J2EE and Web Cache on the Web Tier"](#) on page 9-52.

## 2.5 How to Use this Guide: The Enterprise Deployment Configuration Process

The configuration process for each architecture is detailed in the following sections. You will select chapters from this guide to install and configure a given architecture. [Figure 2-4](#) depicts this progression of tasks and chapters.

**Figure 2-4 The Enterprise Deployment Configuration Process**



### 2.5.1 Installing and Configuring myJ2EE

1. Install the Metadata Repository on INFRADBHOST1 and INFRADBHOST2, as described in [Section 4.1, "Installing the Oracle Application Server Metadata Repository for the Security Infrastructure"](#) on page 4-1.
2. Install Oracle Internet Directory on OIDHOST1 and OIDHOST2, as described in [Section 4.2, "Installing the Oracle Internet Directory Instances in the Data Tier"](#) on page 4-7.
3. Configure the Load Balancing Router or proxy server and related components, as described in [Section 6.2, "Configuring the Load Balancing Router or Proxy Server"](#) on page 6-2.



4. Install an Oracle Application Server J2EE and Web Cache instance on APPHOST1 and APPHOST2, as described in [Section 6.3.1, "Installing the First Application Tier Application Server Instance on APPHOST1"](#) on page 6-2 and [Section 6.3.2, "Installing the Second Application Tier Application Server Instance on APPHOST2"](#) on page 6-6.
5. Create OC4J instances in the Oracle Application Server instance on APPHOST1, as described in [Section 6.3.3, "Creating OC4J Instances on the Application Tier"](#) on page 6-10.
6. Deploy applications on APPHOST1, as described in [Section 6.3.4, "Deploying J2EE Applications"](#) on page 6-11.
7. Create a DCM-Managed Oracle Application Server Cluster and add the instances to it, as described in [Section 6.3.5, "Creating a DCM-Managed Oracle Application Server Cluster on the Application Tier"](#) on page 6-12.
8. Install a standalone Oracle Application Server J2EE and Web Cache instance on WEBHOST1 and WEBHOST2, as described in [Section 6.4.1, "Installing the Oracle HTTP Servers on WEBHOST1 and WEBHOST2"](#) on page 6-13.
9. Configure the Manually Managed Oracle Application Server Cluster as described in [Section 6.5, "Configuring the Manually Managed Oracle Application Server Cluster"](#) on page 6-15.
10. Configure the Load Balancing Router as described in [Section 6.2, "Configuring the Load Balancing Router or Proxy Server"](#) on page 6-2.
11. Configure the Oracle HTTP Server with the Load Balancing Router as described in [Section 6.6, "Configuring the Oracle HTTP Server with the Load Balancing Router"](#) on page 6-16.
12. Configure OC4J routing as described in [Section 6.7, "Configuring OC4J Routing"](#) on page 6-17.
13. Configure application authentication and authorization with the Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider as described in [Section 5.2, "Option 2: Using the Oracle Application Server Java Authentication and Authorization Service \(JAAS\) Provider"](#) on page 5-15.

## 2.5.2 Installing and Configuring myPortal

1. Install the Metadata Repository on INFRADBHOST1 and INFRADBHOST2, as described in [Section 4.1, "Installing the Oracle Application Server Metadata Repository for the Security Infrastructure"](#) on page 4-1.
2. Configure the Load Balancing Router or proxy server, as described in [Section 7.2, "Configuring the Load Balancing Router or Proxy Server"](#) on page 7-11.
3. Install Oracle Internet Directory on OIDHOST1 and OIDHOST2, as described in [Section 4.2, "Installing the Oracle Internet Directory Instances in the Data Tier"](#) on page 4-7.
4. Install Oracle Delegated Administration Services, Oracle Application Server Single Sign-On and Oracle HTTP Server on IDMHOST1 and IDMHOST2, as described in [Section 5.1.1, "Installing the First Identity Management Configuration"](#) on page 5-1 and [Section 5.1.3, "Installing the Second Identity Management Configuration"](#) on page 5-8.

5. Install OracleAS Web Cache, OracleAS Portal, and Oracle HTTP Server on APPHOST1, as described in [Section 7.3.1, "Installing the First Application Server on APPHOST1"](#) on page 7-14.
6. Configure application server components on APPHOST1, as described in [Section 7.3.3, "Configuring the First Application Server on APPHOST1"](#) on page 7-24.
7. Configure the Load Balancing Router or proxy server and related components, as described in [Section 7.2, "Configuring the Load Balancing Router or Proxy Server"](#).
8. Install OracleAS Web Cache, OracleAS Portal, and Oracle HTTP Server on APPHOST2, as described in [Section 7.3.4, "Installing the Second Application Server on APPHOST2"](#) on page 7-40.
9. Configure application server components on APPHOST2, as described in [Section 7.3.5, "Configuring the Second Application Server on APPHOST2"](#) on page 7-48.
10. Complete the system configuration, as described in:  
[Section 7.3.6, "Configuring OracleAS Web Cache Clusters"](#) on page 7-59  
[Section 7.3.7, "Configuring Load Balancing and Monitoring"](#) on page 7-63  
[Section 7.3.8, "Enabling Session Binding on OracleAS Web Cache Clusters"](#) on page 7-63  
[Section 7.3.9, "Modifying the Oracle Application Server Welcome Page"](#) on page 7-65
11. Test the configuration, as described in [Section 7.4, "Testing the Application Server Tier"](#) on page 7-73.
12. If applicable, configure custom providers as described in [Section 7.5, "Configuring Custom Java Portal Development Kit \(JPDK\) Providers"](#) on page 7-76.

### 2.5.3 Installing and Configuring myBIF

1. Install the Metadata Repository on INFRADBHOST1 and INFRADBHOST2, as described in [Section 4.1, "Installing the Oracle Application Server Metadata Repository for the Security Infrastructure"](#) on page 4-1.
2. Configure the Load Balancing Router or proxy server, as described in [Section 7.2, "Configuring the Load Balancing Router or Proxy Server"](#) on page 7-11.
3. Install Oracle Internet Directory on OIDHOST1 and OIDHOST2, as described in [Section 4.2, "Installing the Oracle Internet Directory Instances in the Data Tier"](#) on page 4-7.
4. Install Oracle Delegated Administration Services, Oracle Application Server Single Sign-On and Oracle HTTP Server on IDMHOST1 and IDMHOST2, as described in [Section 5.1.1, "Installing the First Identity Management Configuration"](#) on page 5-1 and [Section 5.1.3, "Installing the Second Identity Management Configuration"](#) on page 5-8.
5. Install Oracle Application Server 10g Portal, Oracle Application Server 10g Discoverer, Oracle Application Server 10g Reports Services, and Oracle Application Server 10g Forms Services on APPHOST1, as described in [Section 8.3.1, "Installing the First Application Server on APPHOST1"](#) on page 8-2.

6. Configure application server components on APPHOST1, as described in [Section 8.3.2, "Configuring the First Application Server on APPHOST1"](#) on page 8-7.
7. Install Oracle Application Server 10g Portal, Oracle Application Server 10g Discoverer, Oracle Application Server 10g Reports Services, and Oracle Application Server 10g Forms Services on APPHOST2, as described in [Section 8.3.3, "Installing the Second Application Server on APPHOST2"](#) on page 8-10.
8. Configure application server components on APPHOST2, as described in [Section 8.3.4, "Configuring the Second Application Server on APPHOST2"](#) on page 8-10.

## 2.6 Selecting a Deployment Architecture

[Table 2–15](#) summarizes the configurations in terms of deployed components, the number of physical servers and tiers used, and the estimated duration of installation and configuration activities.

**Table 2–15** *Deployment Architecture Selection Data*

Installation Type/Components	Identity Management Components	myJ2EE	myPortal	MyBIF
Deployed components	OHS	Web Cache	Web Cache	Web Cache
	SSO / DAS	OHS	OHS	OHS
	OID	OC4J	Portal	OC4J
				Discoverer Forms
Number of servers (without database servers)	4	6	6	6
Number of server tiers	3	4	3	3 or 4
Average time to install and configure	5 hours	5 hours	10 hours	4 hours



---

## Before You Begin Installation

This chapter provides recommendations for a successful Enterprise Deployment installation.

### 3.1 Best Practices for Installing and Configuring Enterprise Deployments

Adherence to the following practices may save you time as you install and configure the architectures described in this guide:

- Before each configuration step, make a complete file system backup of the entire Oracle home, capturing the previous step on all computers at the same time. If there is a problem at any point during installation or configuration, you can then return to the previous state by restoring the backup to all computers at the same time.

---

**Note:** On UNIX systems, when using the `tar` utility, issue the `tar` or `untar` command as the root user. Some of the executables in Oracle software are owned by root. Backing up files in this way as the root user does not change ownership of the file system, or symbolic links inside folders and subfolders.

---

- Try to keep user IDs, group IDs, Oracle home paths and directory structures the same on both computers for each component installed.
- Use the static ports feature of the installer when installing components, to ensure that the same ports are used on both computers for each component. (Ideally, you would use the same `staticports.ini` file for the first and second installations of a given installation type on each tier.)

### 3.2 Hardware Sizing Guidelines

All Enterprise Deployment configurations in this guide use two servers for each tier to provide failover capability; however, this does not presume adequate computing resources for any application or user population. If the system workload increases such that performance is degraded, you can add servers to the configuration by repeating the instructions for the installation and configuration of the second server on the tier (WEBHOST2, APPHOST2, INFRADBHOST2) to add a third server where it is needed.

To determine hardware needs with a greater degree of precision, you might consider the options presented in [Table 3-1](#).

**Table 3–1 Hardware Sizing Options**

Option	Benefit	Disadvantage
Create a prototype of the deployment architecture and stress test it	<ul style="list-style-type: none"> <li>■ Accurate estimate; provides ability to extrapolate</li> <li>■ Accommodates custom scenarios and complex implementations</li> <li>■ Incorporates third-party components (firewalls, load balancing router); exposes performance and network-specific issues</li> </ul>	<ul style="list-style-type: none"> <li>■ Time and effort required to configure</li> <li>■ Additional software for load simulation required</li> </ul>
Use the iSizer tool	<ul style="list-style-type: none"> <li>■ Fast and easy to use</li> <li>■ Works best in common implementations with one component for each server</li> </ul>	<ul style="list-style-type: none"> <li>■ Inexact results for systems with third-party components, many custom implementation details</li> <li>■ Results difficult to extrapolate in multiple-component architectures</li> </ul>

### 3.3 Managing Oracle Application Server Component Connections

In order to ensure consistent availability of all services, you should ensure that the connection time out values for all Oracle Application Server components are set to a lower time out value than that on the firewall and Load Balancing Router. If the firewall or Load Balancing Router drops a connection without sending a TCP close notification message, then Oracle Application Server components will continue to try to use the connection when it is no longer available.

This guide provides the following instructions for configuring time out values in Oracle Application Server:

- [Section 7.6.1, "Firewall Considerations for OracleAS Portal"](#) on page 7-87.
- [Section 4.1.5, "Configuring the Time out Value in the sqlnet.ora File"](#) on page 4-6.

---

## Installing and Configuring the Security Infrastructure

This chapter provides instructions for creating the Data and Identity Management tiers, distributing the components into the DMZs shown in the Enterprise Deployment architecture depicted in [Figure 2-1, "Enterprise Deployment Architecture for myJ2EECompany.com"](#) on page 2-4 and [Figure 2-2, "Enterprise Deployment Architecture for myPortalCompany.com"](#) on page 2-6.

The Security Infrastructures for myJ2EECompany and myPortalCompany differ in one aspect: the myJ2EECompany architecture does not have an Identity Management tier as part of its Security Infrastructure. Consequently, you do not perform the steps in [Section 4.5, "Installing and Configuring Authentication Services for myPortalCompany.com"](#) when creating the myJ2EECompany architecture.

Before you perform the tasks in this chapter, a two-node Real Application Clusters (RAC) database must be installed. In this chapter, the server names for the database hosts are INFRADBHOST1 and INFRADBHOST2.

This chapter contains the following topics:

[Section 4.1, "Installing the Oracle Application Server Metadata Repository for the Security Infrastructure"](#) on page 4-1

[Section 4.2, "Installing the Oracle Internet Directory Instances in the Data Tier"](#) on page 4-7

[Section 4.3, "Configuring the Virtual Server to Use the Load Balancing Router"](#) on page 4-20

[Section 4.4, "Testing the Data Tier Components"](#) on page 4-20

[Section 4.5, "Installing and Configuring Authentication Services for myPortalCompany.com"](#) on page 4-21

### 4.1 Installing the Oracle Application Server Metadata Repository for the Security Infrastructure

You must install the OracleAS Metadata Repository before you install components into the Security DMZ. Oracle Application Server provides a tool, the Oracle Application Server Metadata Repository Creation Assistant, to create the OracleAS Metadata Repository in an existing database.

The OracleAS Metadata Repository Creation Assistant is available on the OracleAS Metadata Repository Creation Assistant CD-ROM or the Oracle Application Server DVD-ROM. You install the OracleAS Metadata Repository Creation Assistant in its own, separate Oracle home.

To install the OracleAS Metadata Repository, you must perform these steps:

1. Install the OracleAS Metadata Repository Creation Assistant, following the steps in [Section 4.1.1](#).
2. Ensure that the database meets the requirements specified in the "Database Requirements" section of the *Oracle Application Server Metadata Repository Creation Assistant User's Guide*. You can find this guide in the Oracle Application Server platform documentation library for the platform and version you are using. In addition, ensure that:
  - The database computer has at least 512 MB of swap space available for execution of the OracleAS Metadata Repository Creation Assistant
  - There are no dependencies of any kind related to the `ultrasearch` directory in the database's Oracle home. The OracleAS Metadata Repository Creation Assistant replaces this directory with a new version, renaming the existing version of the directory to `ultrasearch_timestamp`.
3. Execute the OracleAS Metadata Repository Creation Assistant, following the steps in [Section 4.1.2](#) or [Section 4.1.3](#).
  - To install into a database using raw devices, follow the steps in [Section 4.1.2](#), "Installing the Metadata Repository in a Database Using Raw Devices" on page 4-3.
  - To install into a database using Oracle Cluster File System, follow the steps in [Section 4.1.3](#), "Installing the Metadata Repository in an Oracle Cluster File System (OCFS)" on page 4-5.
4. Perform the post-installation step described in [Section 4.1.4](#).

### 4.1.1 Installing the OracleAS Metadata Repository Creation Assistant

Follow these steps to install the OracleAS Metadata Repository Creation Assistant into its own Oracle home:

1. Insert the OracleAS Metadata Repository Creation Assistant CD-ROM or the Oracle Application Server DVD-ROM.

---

**Note:** If your computer does not mount CD-ROMs or DVD-ROMs automatically, you must set the mount point manually.

---

2. Start the installer, using the method corresponding to the installation media:  
(CD-ROM)

On UNIX, issue this command: **runInstaller**

On Windows, double-click **setup.exe**

(DVD-ROM) Navigate to the `repca_utilities` directory and do one of the following:

On UNIX, issue this command: **runInstaller**

On Windows, double-click **setup.exe**

The **Welcome** screen appears.

3. Click **Next**.

The **Specify File Locations** screen appears.



4. In the **Name** field, specify a name for the OracleAS Metadata Repository Creation Assistant Oracle home. The Oracle home name must contain only alphanumeric characters and the underscore character, and be 128 characters or fewer.

In the **Destination** field, enter the full path to a new Oracle home in which to install the OracleAS Metadata Repository Creation Assistant, and click **Next**.

5. The **Launch Repository Creation Assistant** screen appears.

6. Select **No** and click **Next**.

The **Summary** screen appears.

7. Click **Install**.

The Configuration Assistants screen appears, executing the OracleAS Metadata Repository Creation Assistant, and indicating "In Progress".

8. When the OracleAS Metadata Repository Creation Assistant is no longer running, exit the OracleAS Metadata Repository Creation Assistant.

The **End of Installation** screen appears.

9. Click **Exit**, and then confirm your choice to exit.

## 4.1.2 Installing the Metadata Repository in a Database Using Raw Devices

Follow these steps to install the Metadata Repository into an existing two-node Real Application Clusters (RAC) database using raw devices:

1. Create raw devices for the OracleAS Metadata Repository, using the values in [Section B.2, "Tablespace Mapping to Raw Devices Sample File"](#) on page B-1.

**Tip:** The command to create tablespaces is specific to the volume manager used. For example, the command to create a tablespace in VERITAS Volume Manager is `vxassist`.

2. Create a file to map the tablespaces to the raw devices. Each line in the file has the format:

```
tablespace name=raw device file path
```

You can use the sample file shown in [Example B-1, "Tablespace to Raw Device Mapping \(Sample File\)"](#) on page B-2, replacing the file paths with the paths on your system. Append a 1 to the tablespace names, as shown in the sample file.

---

**Note:** Creating the sample file is not mandatory; you can enter the tablespace values into the Specify Tablespace Information screen during execution of the OracleAS Metadata Repository Creation Assistant.

---

3. Populate the `DBCA_RAW_CONFIG` environment variable with the full path and filename of the tablespace mapping file.
4. Ensure that the database and listener are running.

5. Ensure that the NLS\_LANG environment variable is not set to a non-English locale, or is set to `american_america.us7ascii`, with one of the following commands:

UNIX:

- **`unsetenv NLS_LANG`**
- **`setenv NLS_LANG american_america.us7ascii`**

Windows:

- **`set NLS_LANG=`**
- **`set NLS_LANG=american_america.us7ascii`**

---

**Note:** If you need to, you can set NLS\_LANG to its original value after executing the OracleAS Metadata Repository Creation Assistant.

---

6. Start the OracleAS Metadata Repository Creation Assistant from the OracleAS Metadata Repository Creation Assistant Oracle home with this command:

**`runRepca`**

The **Welcome** screen appears.

7. Click **Next**.

The **Specify Oracle Home Path** screen appears.

8. In the **Oracle Home** field, specify the full path of the database Oracle home.

In the **Log File Directory** field, specify the full path of the directory on the current computer in which you want the OracleAS Metadata Repository Creation Assistant to write its log files. Ensure correct input for the **Log File Directory** on this screen, as you will not be able to change it after you have proceeded beyond this screen.

9. Click **Next**.

The **Select Operation** screen appears.

10. Select **Load** and click **Next**.

The **Specify Database Connect** screen appears.

11. Enter the SYS user name and password and the host and port information. For example:

`infradbhost1.mycompany.com:1521,infradbhost2.mycompany.com:1521`

12. Click **Next**.

The **Specify Storage Options** screen appears.

13. Select **Regular or Cluster File System**.

The **Specify Tablespace Information** screen appears, displaying the values from the file specified by the DBCA\_RAW\_CONFIG environment variable.

14. Correct the values, if necessary, and click **Next**.

The **Warning: Check Disk Space** dialog appears if your SYSTEM and UNDO tablespaces are set to autoextend.

15. Check the disk space as specified in the dialog and click **OK**.

The **Loading Repository** screen appears. The tablespaces and schemas are created and populated.

The **Success** screen appears.

16. Click **OK**.

The OracleAS Metadata Repository Creation Assistant exits.

If the installation was unsuccessful, or you need more information, see the *Oracle Application Server Metadata Repository Creation Assistant User's Guide*.

### 4.1.3 Installing the Metadata Repository in an Oracle Cluster File System (OCFS)

Follow these steps to install the Metadata Repository into an existing two-node Real Application Clusters (RAC) database using an OCFS file system:

1. Ensure that the database and listener are running.
2. Start the OracleAS Metadata Repository Creation Assistant from the OracleAS Metadata Repository Creation Assistant Oracle home with this command:

**runRepca**

The **Welcome** screen appears.

3. Click **Next**.

The **Specify Oracle Home Path** screen appears.

4. In the **Oracle Home** field, specify the full path of the database Oracle home.

In the **Log File Directory** field, specify the full path of the directory on the current computer in which you want the OracleAS Metadata Repository Creation Assistant to write its log files. Ensure correct input for the **Log File Directory** on this screen, as you will not be able to change it after you have proceeded beyond this screen.

5. Click **Next**.

The **Select Operation** screen appears.

6. Select **Load** and click **Next**.

The **Specify Database Connect** screen appears.

7. Enter the SYS user password, select the **Real Application Clusters Database** option, and enter the host and port information. For example:

infradbhost1.mycompany.com:1521,infradbhost2.mycompany.com:1521

Enter the service name.

8. Click **Next**.

The **Specify Storage Options** screen appears.

9. Select **Regular or Cluster File System**.

The **Specify Tablespace Information** screen appears.

10. Select a directory option (**Use Same Directory for All Tablespaces** or **Use Individual Directories for Each Tablespace**) and complete the remaining fields. When specifying a directory, ensure that it is an existing, writable directory with sufficient free space. Click **Next**.

The **Warning: Check Disk Space** dialog appears if your SYSTEM and UNDO tablespaces are set to autoextend.

11. Check the disk space as specified in the dialog and click **OK**.

The **Loading Repository** screen appears. The tablespaces and schemas are created and populated.

The **Success** screen appears.

12. Click **OK**.

The OracleAS Metadata Repository Creation Assistant exits.

If the installation was unsuccessful, or you need more information, see the *Oracle Application Server Metadata Repository Creation Assistant User's Guide*.

#### 4.1.4 Updating the sqlnet.ora File for OracleAS Portal Communication

After you install the OracleAS Metadata Repository into the database, you must update the `sqlnet.ora` file, as follows:

Edit the `ORACLE_HOME/network/admin/sqlnet.ora` file to configure SQL\*Net settings to make the ORASSO\_PS schema accessible. Add LDAP to the `NAMES.DIRECTORY_PATH` entry as follows:

```
NAMES.DIRECTORY_PATH= (LDAP, TNSNAMES, ONAMES, HOSTNAME)
```

Without LDAP in this entry, errors will occur in OracleAS Portal when using the OracleAS Single Sign-On administration portlet.

#### 4.1.5 Configuring the Time out Value in the sqlnet.ora File

You must configure the `SQLNET.EXPIRE_TIME` parameter in the `sqlnet.ora` file on the application infrastructure database. For the OracleAS Single Sign-On server, the parameter must be updated on INFRADBHOST1 and INFRADBHOST2. For the myPortalCompany and myBIFCompany configurations, you will configure this parameter on APPDBHOST1 and APPDBHOST2.

Follow these steps to configure the time out value on the computers specified in the preceding paragraph:

1. Open the file `ORACLE_HOME/network/admin/sqlnet.ora` file (UNIX) or the `ORACLE_BASE/ ORACLE_HOME/network/admin/sqlnet.ora` file (Windows).
2. Set the `SQLNET.EXPIRE_TIME` parameter to a value lower than the TCP session time out value for the Load Balancing Router and firewall.
3. Restart the listener by issuing these commands in `ORACLE_HOME/bin`:

```
lsnrctl stop
```

```
lsnrctl start
```

## 4.2 Installing the Oracle Internet Directory Instances in the Data Tier

Follow these steps to install the Oracle Internet Directory components (OIDHOST1 and OIDHOST2) into the data tier with the Metadata Repository. The procedures are very similar, but the selections in the configuration options screen differ.

---

**Note:** Ensure that the clocks are synchronized between the two computers on which you intend to install the Oracle Internet Directory instances. Errors will occur if this is not done.

---

### 4.2.1 Installing the First Oracle Internet Directory

The OracleAS Metadata Repository must be running before you perform this task. Follow these steps to install Oracle Internet Directory on OIDHOST1:

1. Ensure that the system, patch, kernel and other requirements are met. These are listed in the *Oracle Application Server Quick Installation and Upgrade Guide* in the the Oracle Application Server platform documentation library for the platform and version you are using.
2. Ensure that ports 389 and 636 are not in use by any service on the computer by issuing these commands for the operating system you are using. (If the port is not in use, no output is returned from the command.)

On UNIX:

```
netstat -an | grep "389"
```

```
netstat -an | grep "636"
```

On Windows:

```
netstat -an | findstr :389
```

```
netstat -an | findstr :636
```

If the port is in use (if the command returns output identifying the port), you must free the port.

In UNIX:

Remove the entries for ports 389 and 636 in the `/etc/services` file and restart the services, or restart the computer.

In Windows:

Stop the component that is using the port.

3. Copy the `staticport.ini` file from the `Disk1/stage/Response` directory to the Oracle home directory.
4. Edit the `staticport.ini` file to assign the following custom ports:

```
Oracle Internet Directory port = 389
```

```
Oracle Internet Directory (SSL) port = 636
```

---

**Note:** See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-1 for more information.

---

5. Start the Oracle Universal Installer as follows:  
On UNIX, issue this command: **runInstaller**  
On Windows, double-click **setup.exe**  
The **Welcome** screen appears.
6. Click **Next**.  
On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.
7. Specify the directory you want to be the `oraInventory` directory and the operating system group that has permission to write to it.
8. Click **Next**.  
On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.
9. Open a window and run the script, following the prompts in the window.
10. Return to the Oracle Universal Installer screen and click **Next**.  
The **Specify File Locations** screen appears with default locations for:
  - The product files for the installation (Source)
  - The name and path to an Oracle home (Destination)

---

---

**Note:** Ensure that the Oracle home directory path for `OIDHOST1` is the same as the path to the Oracle home location of `OIDHOST2`. For example, if the path to the Oracle home on `OIDHOST1` is:

`/u01/app/oracle/product/AS10gOID`

then the path to the Oracle home on `OIDHOST2` must be:

`/u01/app/oracle/product/AS10gOID`

---

---

11. Specify the **Destination Name** and **Path**, if different from the default, and click **Next**.  
The **Select a Product to Install** screen appears.

**Figure 4–1 Oracle Universal Installer Select a Product to Install Screen**

12. Select OracleAS Infrastructure 10g, as shown in [Figure 4–1](#), and click **Next**.

The **Select Installation Type** screen appears.

13. Select **Identity Management**, as shown in [Figure 4–2](#), and click **Next**.

**Figure 4–2 Oracle Universal Installer Select Installation Type Screen**

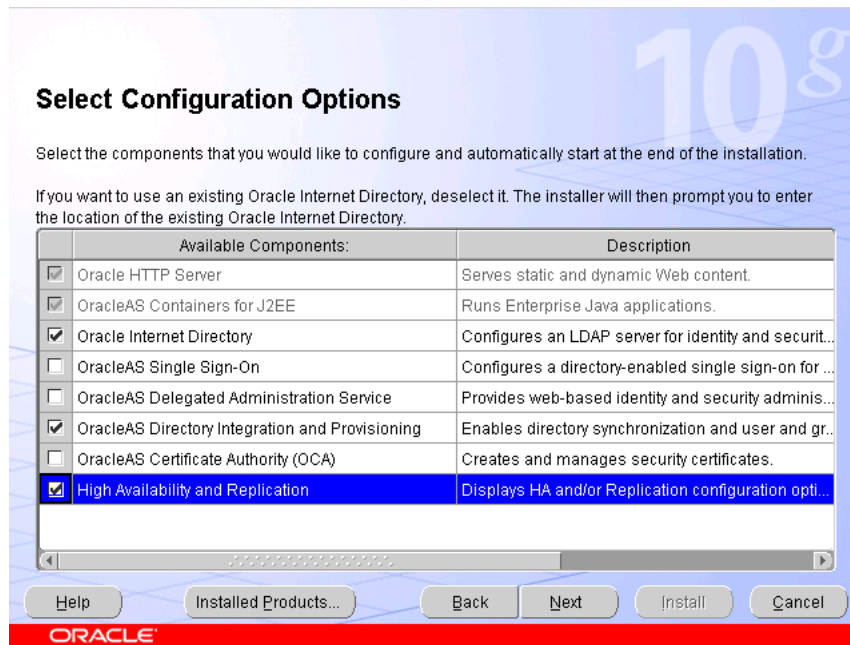
The **Product-Specific Prerequisite Checks** screen appears.

14. Click **Next**.

The **Confirm Pre-Installation Requirements** screen appears.

15. Ensure that the requirements are met, check the box for each, and click **Next**.  
The **Select Configuration Options** screen appears.

**Figure 4–3 Oracle Universal Installer Select Configuration Options Screen**



16. Select **Oracle Internet Directory**, **OracleAS Directory Integration and Provisioning**, and **High Availability and Replication**, as shown in [Figure 4–3](#), and click **Next**.

The **Specify Port Configuration Options** screen appears.

**Figure 4–4 Oracle Universal Installer Specify Port Configuration Options Screen**





17. Select **Manual**, as shown in [Figure 4-4](#), and click **Next**.

The **Specify Repository** screen appears.

18. Provide the DBA login and computer information as shown in [Figure 4-5](#) and click **Next**.

**Figure 4-5 Oracle Universal Installer Specify Repository Screen**

**Specify Repository**

Provide a DBA login to the database containing the Oracle Application Server Metadata Repository that you want to use.

Username:

Password:

Hostname and Port:

Example for a single instance database: Host:1521

Example for a 10g Real Application Clusters database or above:  
Virtual\_hostname\_on\_node1:1521^Virtual\_hostname\_on\_node2:1521...

Example for a 9i Real Application Clusters database: Host1:1521^Host2:1521...

Service Name:

Example: asdb.mydomain.com

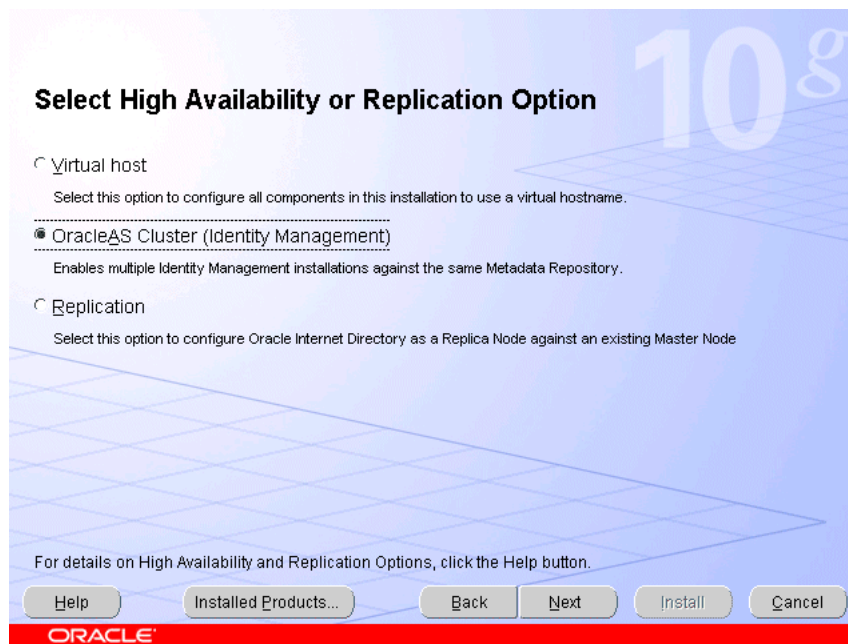
Help Installed Products... Back **Next** Install Cancel

ORACLE

The **Select High Availability or Replication Option** screen appears.

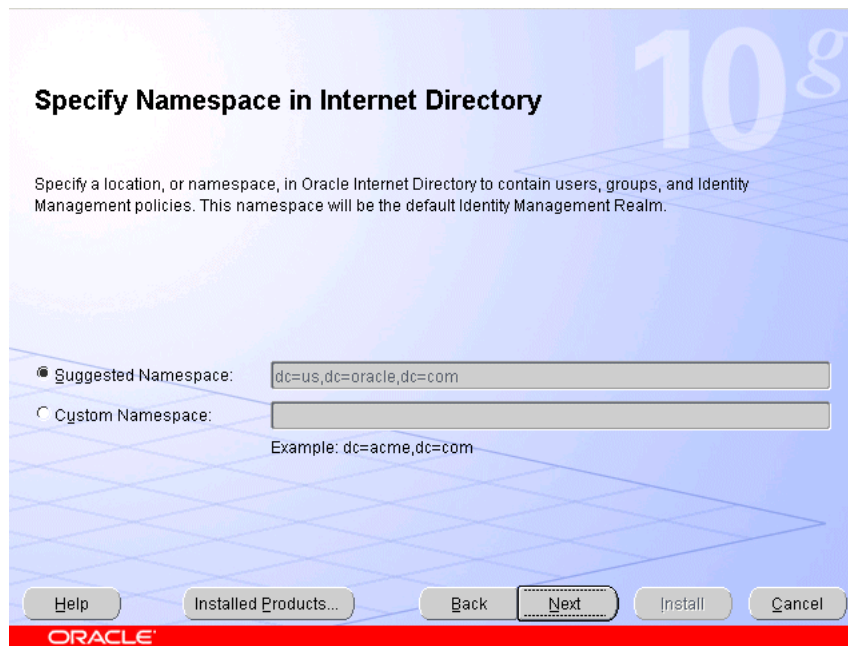
19. Select **OracleAS Cluster (Identity Management)**, as shown in [Figure 4-6](#), and click **Next**.

**Figure 4–6 Oracle Universal Installer Select High Availability or Replication Option Screen**



The **Specify Namespace in Internet Directory** screen appears.

**Figure 4–7 Oracle Universal Installer Specify Namespace in Internet Directory**



20. Click **Next** to specify the default **Suggested Namespace** shown in [Figure 4–7](#), or enter values for the **Custom Namespace** and click **Next**.

The **Specify Instance Name and ias\_admin Password** screen appears.

21. Specify the instance name and password and click **Next**.

The **Summary** screen appears.

22. Review the selections to ensure that they are correct (if they are not, click **Back** to modify selections on previous screens), and click **Install**.

The **Install** screen appears with a progress bar. On UNIX systems, a dialog opens prompting you to run the `root.sh` script.

23. Open a window and run the script.

The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.

24. Click **Exit**, and then confirm your choice to exit.

## 4.2.2 Installing the Second Oracle Internet Directory

The OracleAS Metadata Repository and the first Oracle Internet Directory must be running before you perform this task. Follow these steps to install Oracle Internet Directory on OIDHOST2:

1. Ensure that the system, patch, kernel and other requirements are met. These are listed in the *Oracle Application Server Quick Installation and Upgrade Guide* in the the Oracle Application Server platform documentation library for the platform and version you are using.
2. Ensure that ports 389 and 636 are not in use by any service on the computer by issuing these commands for the operating system you are using. (If the port is not in use, no output is returned from the command.)

On UNIX:

```
netstat -an | grep "389"
```

```
netstat -an | grep "636"
```

On Windows:

```
netstat -an | findstr :389
```

```
netstat -an | findstr :636
```

If the port is in use (if the command returns output identifying the port), you must free the port.

In UNIX:

Remove the entries for ports 389 and 636 in the `/etc/services` file and restart the services, or restart the computer.

In Windows:

Stop the component that is using the port.

3. Copy the `staticport.ini` file from the `Disk1/stage/Response` directory to the Oracle home directory.
4. Edit the `staticport.ini` file and uncomment, and update these entries:

```
Oracle Internet Directory port = 389
```

```
Oracle Internet Directory (SSL) port = 636
```

---

**Note:** See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-1 for more information.

---

5. Start the Oracle Universal Installer as follows:

On UNIX, issue this command: **runInstaller**

On Windows, double-click **setup.exe**

The **Welcome** screen appears.

6. Click **Next**.

On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.

7. Specify the directory you want to be the `oraInventory` directory and the operating system group that has permission to write to it.

8. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.

9. Open a window and run the script, following the prompts in the window.

10. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

- The product files for the installation (Source)
- The name and path to an Oracle home (Destination)

---

**Note:** Ensure that the Oracle home directory path for `OIDHOST1` is the same as the path to the Oracle home location of `OIDHOST2`. For example, if the path to the Oracle home on `OIDHOST1` is:

`/u01/app/oracle/product/AS10gOID`

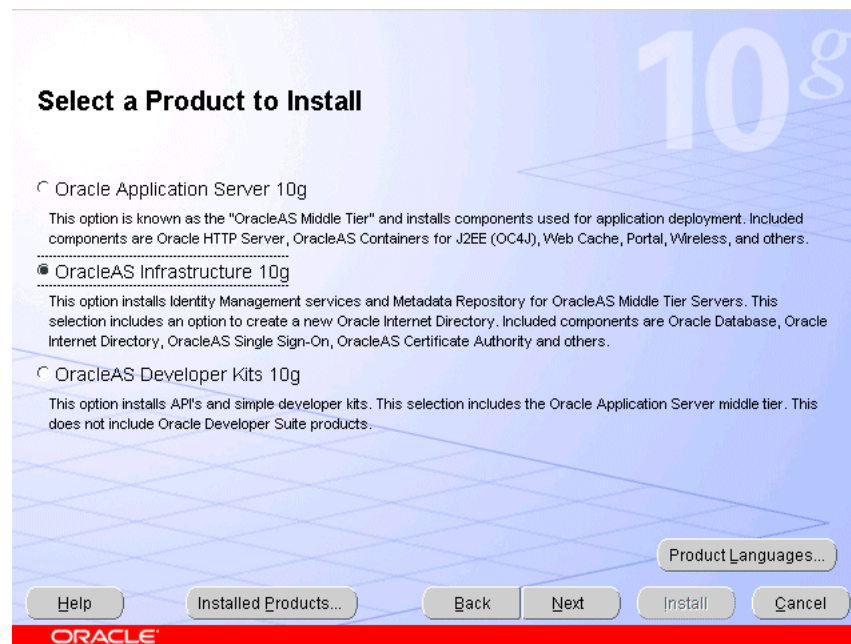
then the path to the Oracle home on `OIDHOST2` must be:

`/u01/app/oracle/product/AS10gOID`

---

11. Specify the **Destination Name** and **Path**, if different from the default, and click **Next**.

The **Select a Product to Install** screen appears.

**Figure 4–8 Oracle Universal Installer Select a Product to Install Screen**

12. Select OracleAS Infrastructure 10g, as shown in [Figure 4–8](#), and click **Next**.

The **Select Installation Type** screen appears.

13. Select **Identity Management**, as shown in [Figure 4–9](#), and click **Next**.

**Figure 4–9 Oracle Universal Installer Select Installation Type Screen**

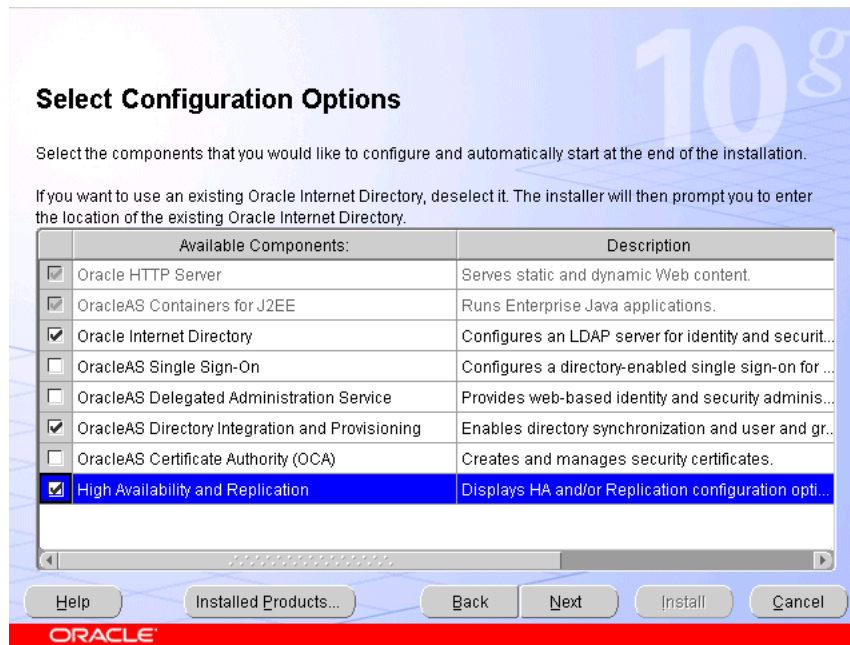
The **Product-specific Prerequisite Checks** screen appears.

14. Click **Next**.

The **Confirm Pre-Installation Requirements** screen appears.

15. Ensure that the requirements are met, check the box for each, and click **Next**.  
The **Select Configuration Options** screen appears.

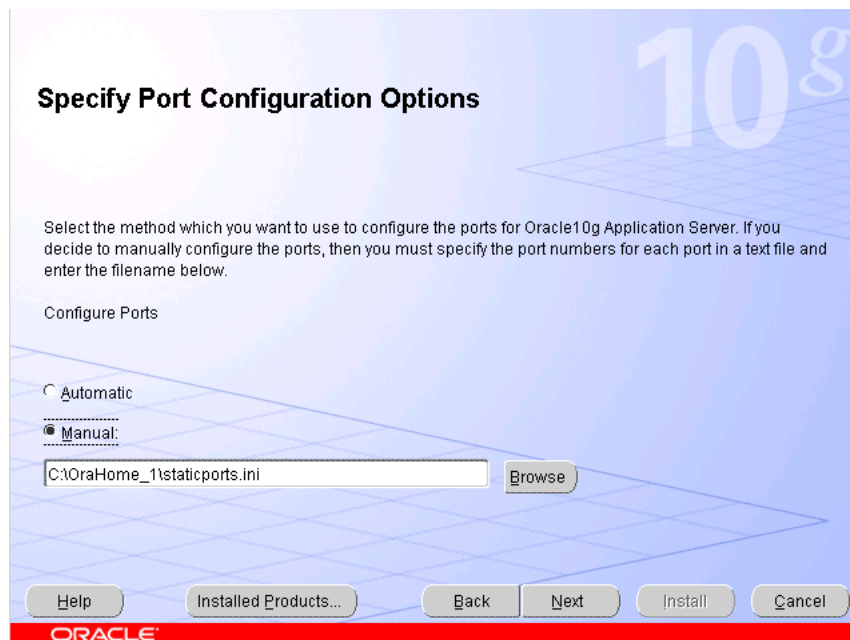
**Figure 4–10 Oracle Universal Installer Select Configuration Options Screen**



16. Select **Oracle Internet Directory**, **OracleAS Directory Integration and Provisioning**, and **High Availability and Replication**, as shown in [Figure 4–10](#), and click **Next**.

The **Specify Port Configuration Options** screen appears.

**Figure 4–11 Oracle Universal Installer Specify Port Configuration Options Screen**



17. Select **Manual**, as shown in [Figure 4-11](#), and click **Next**.

The **Specify Repository** screen appears.

18. Provide the DBA login and computer information as shown in [Figure 4-12](#) and click **Next**.

**Figure 4-12 Oracle Universal Installer Specify Repository Screen**

**Specify Repository**

Provide a DBA login to the database containing the Oracle Application Server Metadata Repository that you want to use.

Username:

Password:

Hostname and Port:

Example for a single instance database: Host:1521

Example for a 10g Real Application Clusters database or above:  
Virtual\_hostname\_on\_node1:1521^Virtual\_hostname\_on\_node2:1521...

Example for a 9i Real Application Clusters database: Host1:1521^Host2:1521...

Service Name:

Example: asdb.mydomain.com

Buttons: Help, Installed Products..., Back, **Next**, Install, Cancel

ORACLE

A dialog opens, prompting you to synchronize the system time of the primary Oracle Internet Directory computer and the system time on the computer on which you are installing.

19. Synchronize the system time on the computers and click **OK**.

The **Specify ODS Password** screen appears.

20. Specify the ODS password (by default, the `ias_admin` password) as shown in [Figure 4-13](#) and click **Next**.



**Figure 4–13 Oracle Universal Installer Specify ODS Password Screen**

**Specify ODS Password**

Specify the password for the ODS Schema for this Metadata Repository.

Password:

Help Installed Products... Back **Next** Install Cancel

ORACLE

The **Specify OID Login** screen appears.

21. Specify the user name and password, as shown in [Figure 4–14](#), and click **Next**.

**Figure 4–14 Oracle Universal Installer Specify OID Login Screen**

**Specify OID Login**

Enter your username and password to connect/login to the Oracle Internet Directory at the hostname and port stada19.us.oracle.com:389. You need to be the Oracle Internet Directory Superuser or a Single Sign-On user. Use cn=orcladmin as the username if you are the Oracle Internet Directory Superuser. Use your Single Sign-On username if you are a Single Sign-On user with the appropriate install privileges.

Username:

Password:

Help Installed Products... Back **Next** Install Cancel

ORACLE

The **Specify Instance Name and ias\_admin Password** screen appears.

22. Specify the instance name and password and click **Next**.

The **Summary** screen appears.



23. Review the selections to ensure that they are correct (if they are not, click **Back** to modify selections on previous screens), and click **Install**.

The **Install** screen appears with a progress bar. On UNIX systems, a dialog opens prompting you to run the `root.sh` script.

24. Open a window and run the script.

The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.

25. Click **Exit**, and then confirm your choice to exit.

## 4.3 Configuring the Virtual Server to Use the Load Balancing Router

You must configure the Load Balancing Router to perform these functions:

- Listen on oid.mycompany.com on ports 389 and 636.
- Balance the requests received on ports 389 and 636 to oidhost1.mycompany.com and oidhost2.mycompany.com on ports 389 and 636.
- Monitor the heartbeat of the OID processes on both computers. If an OID process stops on one of the computers, the Load Balancing Router must route the LDAP traffic to the surviving computer.

---

**Note:** Some tuning of the Load Balancing Router's monitoring interval and time out values may be required to ensure system availability. If the interval or time out value is too long, the Load Balancing Router will not detect service failures in time; if it is too short, the Load Balancing Router may incorrectly infer that a server is down.

For example, suppose the Load Balancing Router maps the virtual IP address oid.mycompany.com to the two Oracle Internet Directory servers for round robin load balancing, and the monitoring scheme attempts an ldapbind at 10-second intervals.

If the Oracle Internet Directory on APPHOST1 is down, then the Load Balancing Router directs all traffic to the Oracle Internet Directory on APPHOST2 only.

However, there is a 10-second interval during which the Load Balancing Router is unaware that the Oracle Internet Directory on APPHOST1 is down. There is also a 30-second time out period. During this period, the Load Balancing Router continues to direct traffic to both Oracle Internet Directory servers in round robin mode, and ldapbind failures will occur when it attempts connections to the Oracle Internet Directory on APPHOST1.

---

## 4.4 Testing the Data Tier Components

Perform these steps to test the Data Tier components:

1. Ensure that you can connect to each Oracle Internet Directory instance and the Load Balancing Router, using this command:

```
ldapbind -p 389 -h OIDHOST1
```

```
ldapbind -p 389 -h OIDHOST2
```

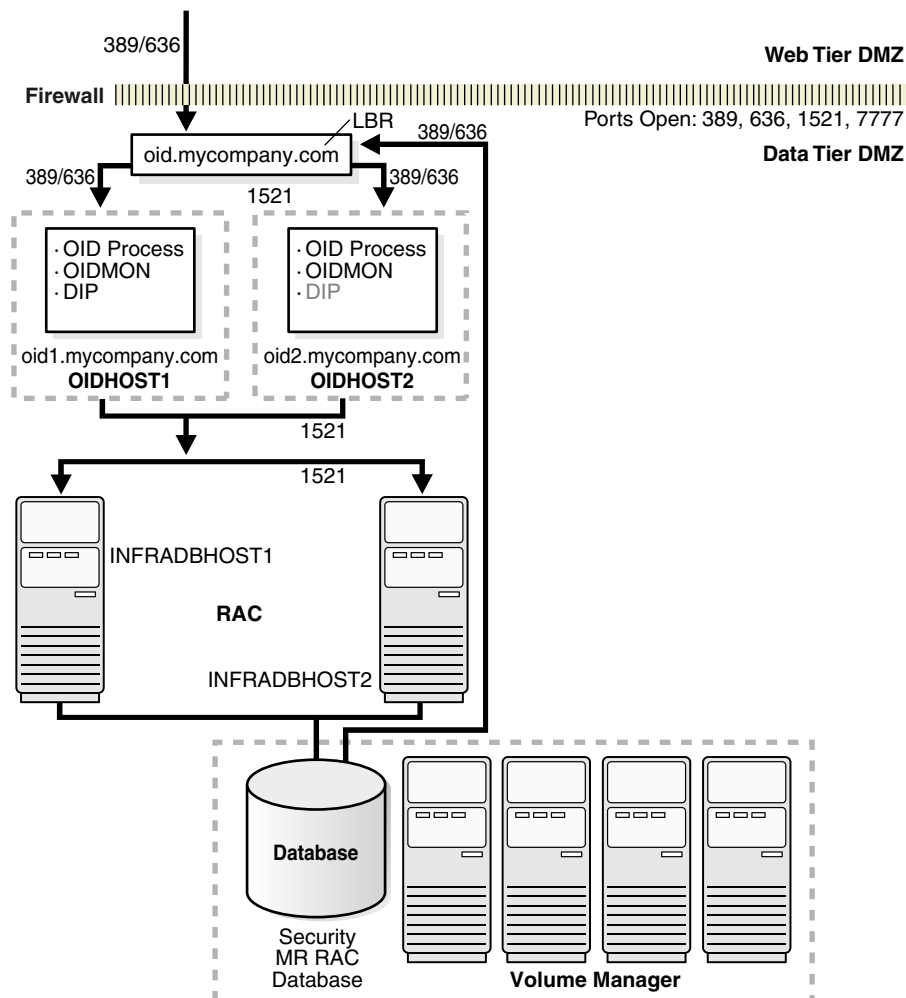
```
ldapbind -p 389 -h oid.mycompany.com
```

2. Start the oidadmin tool on each Oracle Internet Directory instance in *ORACLE\_HOME/bin* with this command:

```
oidadmin
```

The Data Tier configuration is now as shown in [Figure 4–15](#).

**Figure 4–15 Data Tier Configuration**



## 4.5 Installing and Configuring Authentication Services for myPortalCompany.com

If you are creating a Security Infrastructure for the myPortalCompany configuration shown in [Figure 2–2, "Enterprise Deployment Architecture for myPortalCompany.com"](#) on page 2-6, you must configure authentication services on the Identity Management Tier (IDMHOST1 and IDMHOST2). myPortalCompany uses Oracle Application Server Single Sign-On for authentication.

Follow the steps in [Section 5.1, "Option 1: Using Oracle Application Server Single Sign-On"](#) to install and test OracleAS Single Sign-On.

---

**Note:** You must configure the Load Balancing Router (login.mycompany.com) shown in [Figure 5–17, "Identity Management Tier Configuration"](#) for persistent HTTP sessions.

---



---

# Installing and Configuring Authentication Services

This chapter provides instructions for setting up authentication services. The following options exist for providing authentication services in Enterprise Deployment configurations:

- [Section 5.1, "Option 1: Using Oracle Application Server Single Sign-On"](#)
- [Section 5.2, "Option 2: Using the Oracle Application Server Java Authentication and Authorization Service \(JAAS\) Provider"](#)

## 5.1 Option 1: Using Oracle Application Server Single Sign-On

If you are creating a Security Infrastructure for the myPortalCompany configuration shown in [Figure 2–2, "Enterprise Deployment Architecture for myPortalCompany.com"](#) on page 2-6, or the myBIFCompany configuration shown in [Figure 2–3, "Enterprise Deployment Architecture for myBIFCompany.com"](#) you must configure OracleAS Single Sign-On on IDMHOST1 and IDMHOST2. Do not perform the steps in this section if you are configuring myJ2EECompany.

After the Data Tier is complete, follow these steps to install the Identity Management components (IDMHOST1 and IDMHOST2).

---

**Note:** You must configure the Load Balancing Router (login.mycompany.com) shown in [Figure 5–17, "Identity Management Tier Configuration"](#) for persistent HTTP sessions.

---

### 5.1.1 Installing the First Identity Management Configuration

Follow these steps to install Identity Management on IDMHOST1:

1. Ensure that the system, patch, kernel and other requirements are met. These are listed in the *Oracle Application Server Quick Installation and Upgrade Guide* in the the Oracle Application Server platform documentation library for the platform and version you are using.
2. Copy the `staticport.ini` file from the `Disk1/stage/Response` directory to the Oracle home directory.

3. Edit the `staticport.ini` file and uncomment these entries:

```
Oracle HTTP Server port = 7777
Oracle HTTP Server Listen port = 7777
Application Server Control port = 1810
```

---

**Note:** See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-1 for more information.

---

4. Start the Oracle Universal Installer as follows:

On UNIX, issue this command: **runInstaller**

On Windows, double-click **setup.exe**

The **Welcome** screen appears.

5. Click **Next**.

On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.

6. Specify the directory you want to be the `oraInventory` directory and the operating system group that has permission to write to it.

7. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.

8. Open a window and run the script, following the prompts in the window.

9. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

- The product files for the installation (Source)
- The name and path to an Oracle home (Destination)

---

**Note:** Ensure that the Oracle home directory path for `IDMHOST1` is the same as the path to the Oracle home location of `IDMHOST2`. For example, if the path to the Oracle home on `IDMHOST1` is:

```
/u01/app/oracle/product/AS10gSSO
```

then the path to the Oracle home on `IDMHOST2` must be:

```
/u01/app/oracle/product/AS10gSSO
```

---

10. Specify the **Destination Name** and **Path**, if different from the default, and click **Next**.

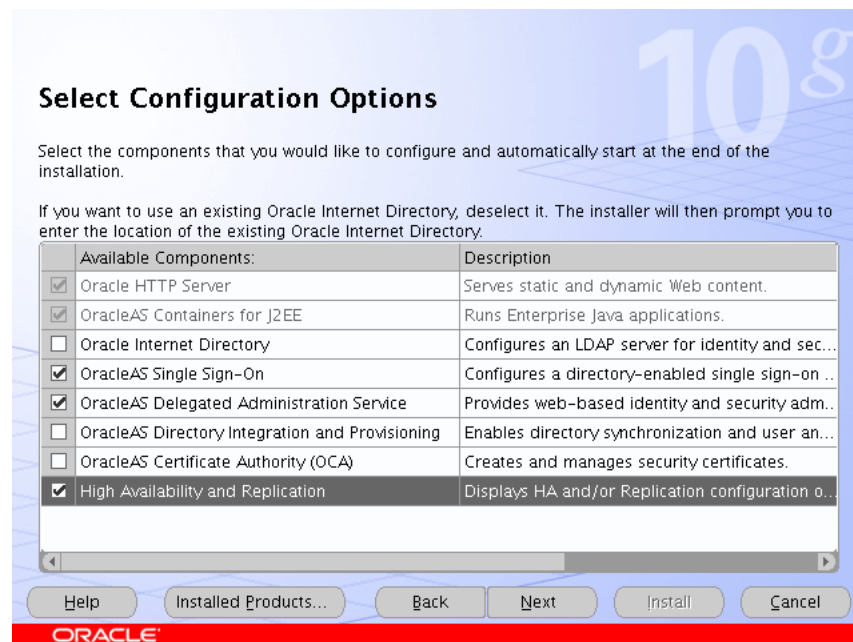
The **Select a Product to Install** screen appears.

**Figure 5–1 Oracle Universal Installer Select a Product to Install Screen**

11. Select OracleAS Infrastructure 10g, as shown in [Figure 5–1](#), and click **Next**.  
The **Select Installation Type** screen appears.

**Figure 5–2 Oracle Universal Installer Select Installation Type Screen**

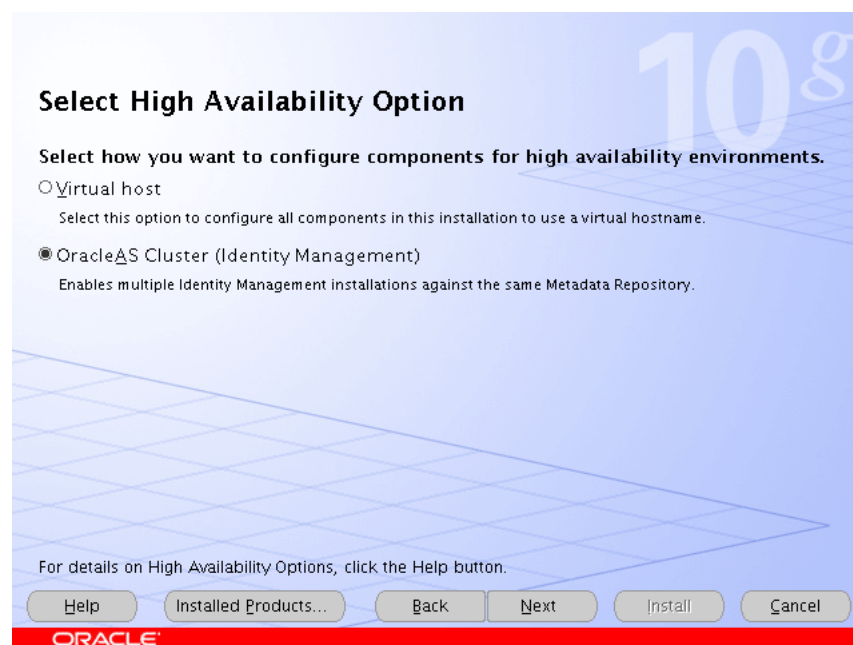
12. Select **Identity Management**, as shown in [Figure 5–2](#), and click **Next**.  
The **Confirm Pre-Installation Requirements** screen appears.
13. Ensure that the requirements are met and click **Next**.  
The **Select Configuration Options** screen appears.

**Figure 5–3 Oracle Universal Installer Select Configuration Options Screen**

14. Select **OracleAS Single Sign-On**, **Oracle Delegated Administration Services**, and **High Availability and Replication**, as shown in [Figure 5–3](#).

The **Specify Port Configuration Options** screen appears.

15. Select **Manual**, specify the location of the `staticports.ini` file, and click **Next**.  
The **Select High Availability Option** screen appears.

**Figure 5–4 Oracle Universal Installer Select High Availability Option Screen**

16. Select **OracleAS Cluster (Identity Management)**, as shown in [Figure 5–4](#), and click **Next**.



The **Create or Join an OracleAS Cluster (Identity Management)** screen appears.

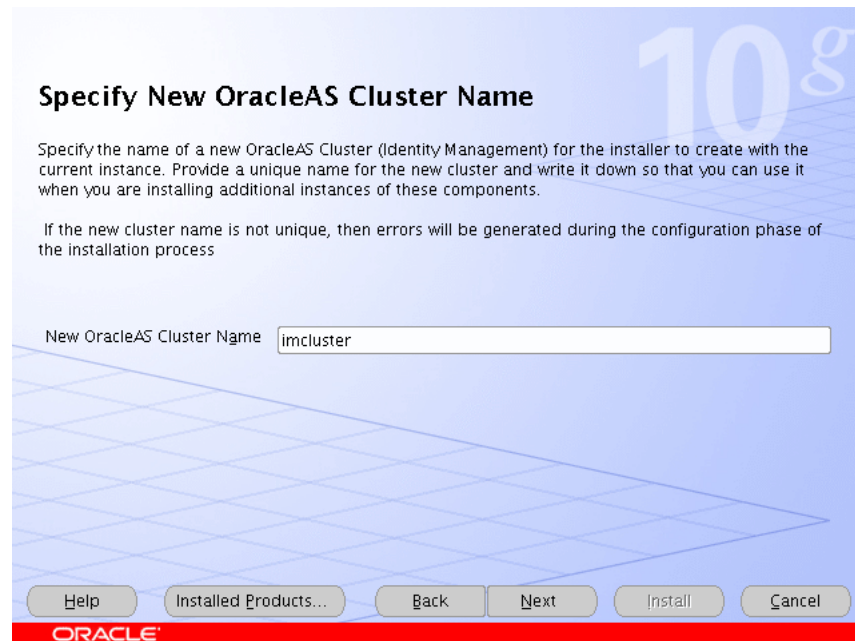
**Figure 5–5 Oracle Universal Installer Create or Join an OracleAS Cluster (Identity Management) Screen**



17. Select **Create a New OracleAS Cluster**, as shown in [Figure 5–5](#), and click **Next**.

The **Specify New OracleAS Cluster Name** screen appears.

**Figure 5–6 Oracle Universal Installer Specify New OracleAS Cluster Name Screen**



18. Complete the **New OracleAS Cluster Name** field with a name for the cluster, as shown in [Figure 5–6](#), and click **Next**.

---

**Note:** Write down the cluster name. You will need to provide it in subsequent installations of instances that will join the cluster.

---

The **Specify LDAP Virtual Host and Ports** screen appears.

**Figure 5–7 Oracle Universal Installer Specify LDAP Virtual Host and Ports Screen**

**Specify LDAP Virtual Host and Ports**

Specify the virtual server host and ports to manage LDAP connections made by Oracle Delegated Administration Services and OracleAS Single Sign-On to Oracle Internet Directory (OID). The virtual host must already be configured to accept and route LDAP connections through the virtual server name and ports specified below. If your virtual server is not configured to manage LDAP connection to OID, please specify OID host and ports information.

Both Ports are required.

Hostname:

SSL Port:

Non-SSL Port:

ORACLE

19. Enter the name of the Load Balancing Router, the SSL port, and the non-SSL port, as shown in [Figure 5–7](#).

20. Click **Next**.

The **Specify OID Login** screen appears.

21. Complete the fields and click **Next**.

The **Specify HTTP Load Balancer Host and Listen Ports** screen appears.

**Figure 5–8 Oracle Universal Installer Specify HTTP Load Balancer Host and Listen Ports Screen**

**Specify HTTP Load Balancer Host and Listen Ports**

Specify HTTP Load Balancer Host and Listen Ports to manage HTTP connections made by client applications to Oracle Delegated Administration Services and OracleAS Single Sign-On. Note that when you enable SSL (Secure Socket Layer) for the HTTP Listen port, the HTTP load balancer port will also be automatically SSL enabled.

HTTP Listener:

Port:

☐ Enable SSL

HTTP Load Balancer:

Hostname:

Port:

☒ Enable SSL

Help Installed Products... Back Next Install Cancel

ORACLE

22. Enter the listen port of the HTTP Server and the host name and port of the HTTP Load Balancer, enabling the SSL option for the load balancer, as shown in [Figure 5–8](#).

23. Click **Next**.

The **Specify Instance Name and ias\_admin Password** screen appears.

24. Specify the instance name and password and click **Next**.

The **Summary** screen appears.

25. Review the selections to ensure that they are correct (if they are not, click **Back** to modify selections on previous screens), and click **Install**.

The **Install** screen appears with a progress bar. On UNIX systems, a dialog opens prompting you to run the `root.sh` script.

26. Open a window and run the script.

The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.

27. Click **Exit**, and then confirm your choice to exit.

## 5.1.2 Testing the Identity Management Components With Oracle Internet Directory

Follow these steps to test the first Identity Management installation with the Oracle Internet Directory:

1. Stop all components on OIDHOST1, using this command:

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

2. Ensure that all components on OIDHOST2 are running:

```
ORACLE_HOME/opmn/bin/opmnctl status
```

3. Access the following URLs:

```
https://login.mycompany.com/pls/orasso
```

```
https://login.mycompany.com/oiddas
```

## 5.1.3 Installing the Second Identity Management Configuration

Follow these steps to install Identity Management on IDMHOST2:

1. Ensure that the system, patch, kernel and other requirements are met. These are listed in the *Oracle Application Server Quick Installation and Upgrade Guide* in the the Oracle Application Server platform documentation library for the platform and version you are using.
2. Copy the `staticport.ini` file from the `Disk1/stage/Response` directory to the Oracle home directory.
3. Edit the `staticport.ini` file and uncomment these entries:

```
Oracle HTTP Server port = 7777
Oracle HTTP Server Listen port = 7777
Application Server Control port = 1810
```

---

---

**Note:** See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-1 for more information.

---

---

4. Start the Oracle Universal Installer as follows:  
On UNIX, issue this command: **runInstaller**  
On Windows, double-click **setup.exe**  
The **Welcome** screen appears.
5. Click **Next**.  
On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.
6. Specify the directory you want to be the `oraInventory` directory and the operating system group that has permission to write to it.
7. Click **Next**.  
On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.
8. Open a window and run the script, following the prompts in the window.

9. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

- The product files for the installation (Source)
- The name and path to an Oracle home (Destination)

---

**Note:** Ensure that the Oracle home directory path for IDMHOST1 is the same as the path to the Oracle home location of IDMHOST2. For example, if the path to the Oracle home on IDMHOST1 is:

```
/u01/app/oracle/product/AS10gSSO
```

then the path to the Oracle home on IDMHOST2 must be:

```
/u01/app/oracle/product/AS10gSSO
```

---

10. Specify the **Destination Name** and **Path**, if different from the default, and click **Next**.

The **Select a Product to Install** screen appears.

**Figure 5–9 Oracle Universal Installer Select a Product to Install Screen**



11. Select OracleAS Infrastructure 10g, as shown in [Figure 5–9](#), and click **Next**.

The **Select Installation Type** screen appears.

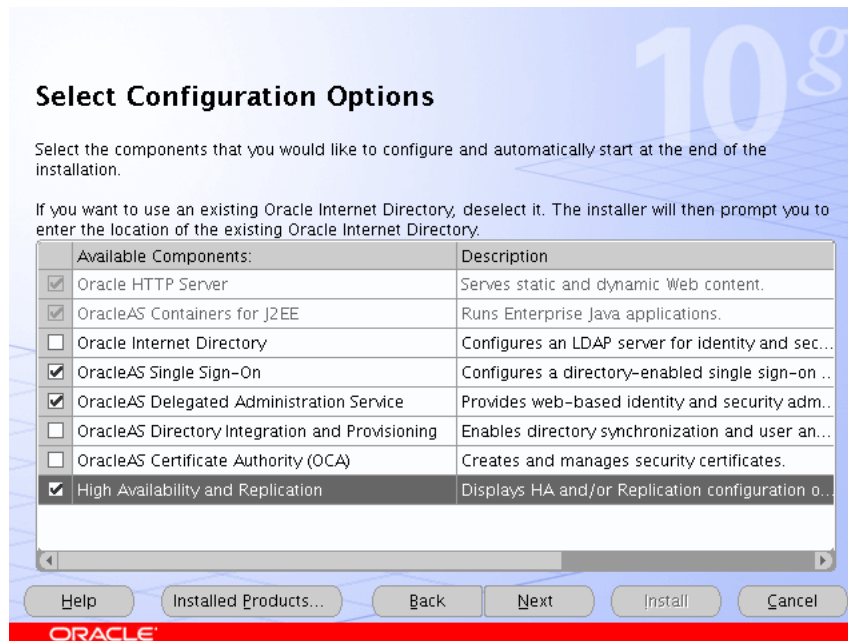
**Figure 5–10 Oracle Universal Installer Select Installation Type Screen**

12. Select **Identity Management** as shown in [Figure 5–10](#), and click **Next**.

The **Confirm Pre-Installation Requirements** screen appears.

13. Ensure that the requirements are met and click **Next**.

The **Select Configuration Options** screen appears.

**Figure 5–11 Oracle Universal Installer Select Configuration Options Screen**

14. Select **OracleAS Single Sign-On**, **Oracle Delegated Administration Services**, and **High Availability and Replication**, as shown in [Figure 5–11](#).



## 15. Click Next.

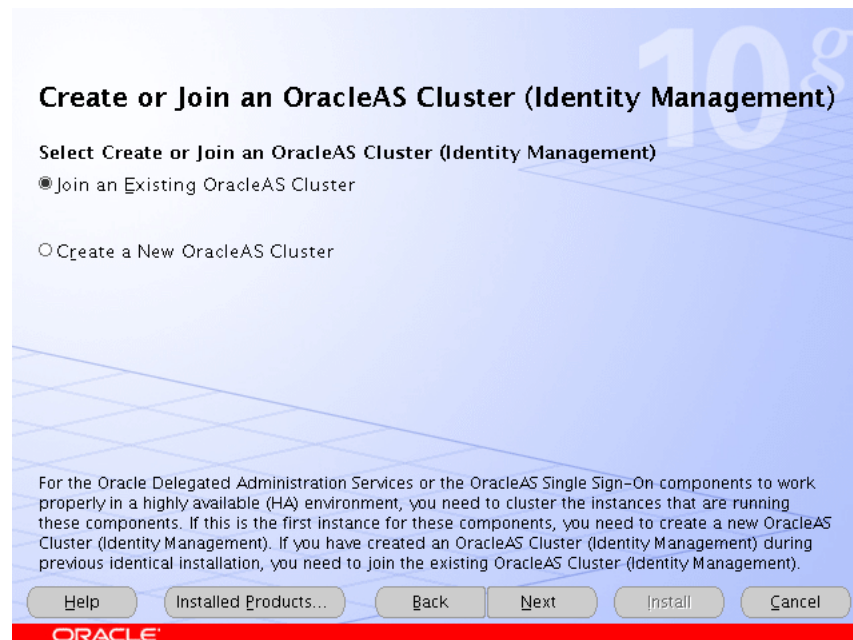
The **Select High Availability Option** screen appears.

**Figure 5–12 Oracle Universal Installer Select High Availability Option Screen**

16. Select **OracleAS Cluster (Identity Management)**, as shown in [Figure 5–12](#), and click **Next**.

The **Create or Join an OracleAS Cluster (Identity Management)** screen appears.

**Figure 5–13 Oracle Universal Installer Create or Join an OracleAS Cluster (Identity Management) Screen**



17. Select **Join an Existing OracleAS Cluster**, as shown in [Figure 5-5](#), and click **Next**.  
The **Specify Existing OracleAS Cluster Name** screen appears.

**Figure 5-14 Oracle Universal Installer Specify Existing OracleAS Cluster Name Screen**

**Specify Existing OracleAS Cluster Name**

Specify an existing OracleAS Cluster (Identity Management) for the current instance to join. The cluster was created during a previous identical installation.

If the existing cluster name is not accurate then, errors will be generated during the configuration phase of the installation process.

Existing OracleAS Cluster Name

Help Installed Products... Back Next Install Cancel

ORACLE

18. Complete the **Existing OracleAS Cluster Name** field with the name you provided for the cluster when installing the first instance, as shown in [Figure 5-6](#), and click **Next**.

The **Specify LDAP Virtual Host and Ports** screen appears.

**Figure 5-15 Oracle Universal Installer Specify LDAP Virtual Host and Ports Screen**

**Specify LDAP Virtual Host and Ports**

Specify the virtual server host and ports to manage LDAP connections made by Oracle Delegated Administration Services and OracleAS Single Sign-On to Oracle Internet Directory (OID). The virtual host must already be configured to accept and route LDAP connections through the virtual server name and ports specified below. If your virtual server is not configured to manage LDAP connection to OID, please specify OID host and ports information.

Both Ports are required.

Hostname:

SSL Port:

Non-SSL Port:

Help Installed Products... Back Next Install Cancel

ORACLE



19. Enter the name of the Load Balancing Router, the SSL port, and the non-SSL port, as shown in [Figure 5-7](#).

20. Click **Next**.

The **Specify OID Login** screen appears.

21. Complete the fields and click **Next**.

The **Specify HTTP Load Balancer Host and Listen Ports** screen appears.

**Figure 5-16 Oracle Universal Installer Specify HTTP Load Balancer Host and Listen Ports Screen**

**Specify HTTP Load Balancer Host and Listen Ports**

Specify HTTP Load Balancer Host and Listen Ports to manage HTTP connections made by client applications to Oracle Delegated Administration Services and OracleAS Single Sign-On. Note that when you enable SSL (Secure Socket Layer) for the HTTP Listen port, the HTTP load balancer port will also be automatically SSL enabled.

HTTP Listener:

Port:

☐ Enable SSL

HTTP Load Balancer:

Hostname:

Port:

☒ Enable SSL

Help Installed Products... Back Next Install Cancel

ORACLE

22. Enter the listen port of the HTTP Server and the host name and port of the HTTP Load Balancer, enabling the SSL option for the load balancer, as shown in [Figure 5-16](#).

23. Click **Next**.

The **Specify Instance Name and ias\_admin Password** screen appears.

24. Specify the instance name and password and click **Next**.

The **Summary** screen appears.

25. Review the selections to ensure that they are correct (if they are not, click **Back** to modify selections on previous screens), and click **Install**.

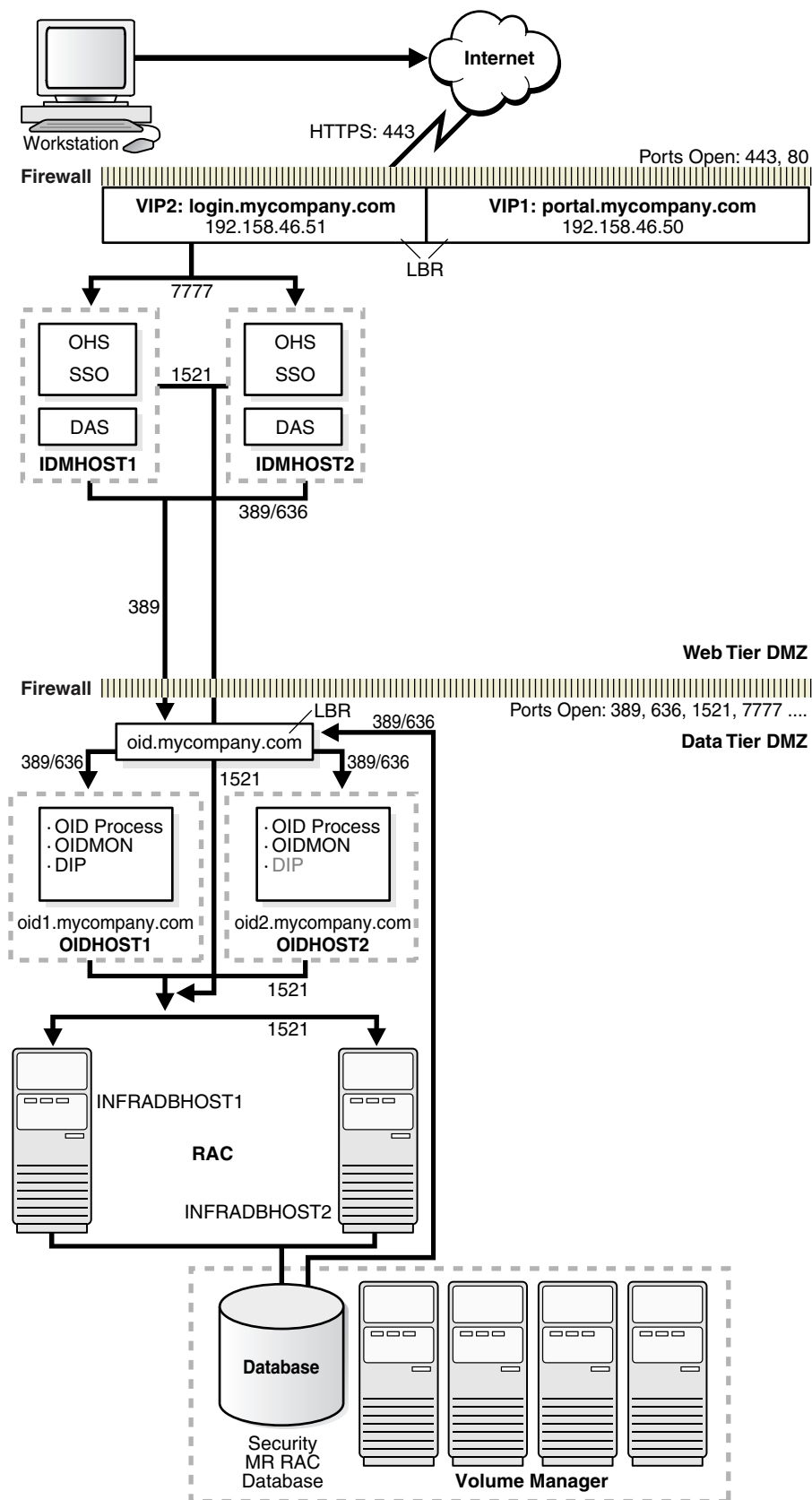
The **Install** screen appears with a progress bar. On UNIX systems, a dialog opens prompting you to run the `root.sh` script.

26. Open a window and run the script.

The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.

27. Click **Exit**, and then confirm your choice to exit.

The Identity Management configuration is now as shown in [Figure 5-17](#).

**Figure 5–17 Identity Management Tier Configuration**

### 5.1.4 Testing the Identity Management Tier Components

After both Identity Management configurations are complete, test the configurations as follows:

1. Stop all components on APPHOST1, using this command:  
`ORACLE_HOME/opmn/bin/opmnctl stopall`
2. Ensure that all components on APPHOST2 are running, using this command:  
`ORACLE_HOME/opmn/bin/opmnctl status`
3. Access the following URLs from two browsers:  
`https://login.mycompany.com/pls/orasso`  
`https://login.mycompany.com/oiddas`
4. Start all components from APPHOST1, using this command:  
`ORACLE_HOME/opmn/bin/opmnctl startall`
5. Stop all components on APPHOST2, using this command:  
`ORACLE_HOME/opmn/bin/opmnctl stopall`
6. Ensure that the login session is still valid for the `orasso` and `oiddas` logins.

## 5.2 Option 2: Using the Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider

The Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider (also referred to as JAZN) LDAP-based provider is used for authentication and authorization to the OC4J applications.

In the myJ2EECompany configuration, this provider is used without Oracle Application Server Single Sign-On, because communication to the data tier is prohibited (Oracle Application Server Single Sign-On requires Portal Services access to the database). This section explains how to configure the Oracle Application Server instances on the application tier to use the JAZN LDAP provider.

For instructions on how to use Oracle Enterprise Manager 10g to manage the data in this provider, see Chapter 8 in the *Oracle Application Server Containers for J2EE Security Guide*.

You will need to follow the steps in this section on both Oracle Application Server instances (APPHOST1 and APPHOST2) that will use the JAZN LDAP provider. Ensure that you specify the same Oracle Internet Directory computer for APPHOST1 and APPHOST2—that is, the load balancing router for OIDHOST1 and OIDHOST2.

Ensure that the middle tier instance is stopped and the Oracle Internet Directory instance is running. Start the Oracle Enterprise Manager 10g Application Server Control Console, if necessary, and perform these steps:

1. On the **Application Server** page, click the **Infrastructure** link.  
 The **Infrastructure** page appears.
2. In the **Identity Management** section, click **Configure**.  
 The **Configure Identity Management: Internet Directory** page appears.
3. In the **Host** field, enter the host name of the Load Balancing Router (for example, `oid.mycompany.com`, in [Figure 2-1](#)).

4. In the **Port** field, enter **389**.

5. Click **Next**.

The **Configure Identity Management: Login** page appears.

6. In the **User Name** field, enter the name of the user (in the IASAdmins group) that can log in to Oracle Internet Directory.

7. In the **Password** field, enter the user's password.

8. Click **Next**.

The **Configure Identity Management: Validation** page appears.

9. Ensure that the **Oracle Internet Directory Host** and **Oracle Internet Directory Port** values are correct.

10. If the values are correct, click **Finish**. (If not, click **Back**, and then click **Back** again to navigate to the **Configure Identity Management: Internet Directory** page and correct the **Host** and **Port** fields.)

A message appears notifying you that the configuration was successful.

### 5.2.1 Adding Administrative Users and Groups to Oracle Internet Directory for the OracleAS JAAS Provider

To use the OracleAS JAAS Provider, you must populate Oracle Internet Directory with certain user entries. The *Oracle Application Server Containers for J2EE Security Guide*, section titled "Creating Administrative Users and Groups for JAZN/LDAP", provides instructions for loading the entries.

---

## Installing and Configuring the myJ2EECompany Application Infrastructure

This chapter provides instructions for creating the Data, E-Business and Web Server tiers, distributing the software components into the DMZs shown in the Enterprise Deployment architecture for myJ2EECompany shown in [Figure 2-1](#) on page 2-4.

Before you perform the tasks in this chapter, a two-node Real Application Clusters (RAC) database must be installed. In this chapter, the server names for the database hosts are APPDBHOST1 and APPDBHOST2. Ideally, these are separate physical databases from INFRADBHOST1 and INFRADBHOST2. In addition to isolating the security components, separate application databases provide the flexibility needed to maintain and tune application and security parameters separately.

This chapter contains the following topics:

[Section 6.1, "Installing and Configuring the Security Infrastructure"](#) on page 6-1

[Section 6.2, "Configuring the Load Balancing Router or Proxy Server"](#) on page 6-2

[Section 6.3, "Installing and Configuring the Application Tier"](#) on page 6-2

[Section 6.4, "Installing and Configuring the Web Tier"](#) on page 6-13

[Section 6.5, "Configuring the Manually Managed Oracle Application Server Cluster"](#) on page 6-15

[Section 6.6, "Configuring the Oracle HTTP Server with the Load Balancing Router"](#) on page 6-16

[Section 6.7, "Configuring OC4J Routing"](#) on page 6-17

[Section 6.8, "Configuring Application Authentication and Authorization"](#) on page 6-18

### 6.1 Installing and Configuring the Security Infrastructure

The security infrastructure for myJ2EECompany contains the components depicted in [Figure 4-15, "Data Tier Configuration"](#). The Security Infrastructures for myJ2EECompany and myPortalCompany differ in one aspect: the myJ2EECompany architecture does not have an Identity Management tier as part of its Security Infrastructure. The Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider is used instead of Oracle Application Server Single Sign-On, so there is no Identity Management Tier in the myJ2EECompany configuration. The OracleAS JAAS Provider is referred to as the JAZN LDAP User Manager in the Deploy Applications: User Manager screen in the Oracle Enterprise Manager 10g Application Server Control Console.

The Oracle Internet Directory administration utility `oiddas` is required for Oracle Internet Directory administration. `oiddas` is installed in the application server environment with the Oracle Internet Directoryserver.

To install and configure this security infrastructure:

1. Follow all instructions in [Section 4.1, "Installing the Oracle Application Server Metadata Repository for the Security Infrastructure"](#) on page 4-1.
2. Follow all instructions in [Section 4.2, "Installing the Oracle Internet Directory Instances in the Data Tier"](#) on page 4-7.
3. Follow all instructions in [Section 4.3, "Configuring the Virtual Server to Use the Load Balancing Router"](#) on page 4-20.
4. Follow all instructions in [Section 4.4, "Testing the Data Tier Components"](#) on page 4-20.

## 6.2 Configuring the Load Balancing Router or Proxy Server

If you are using a Load Balancing Router (`myapp.mycompany.com`, shown in [Figure 2-1, "Enterprise Deployment Architecture for myJ2EECompany.com"](#) on page 2-4), it must be configured to receive client requests and balance them to the two Oracle HTTP Server instances on the Web tier. See the load balancing router documentation for instructions.

If you are using a proxy server, follow the instructions in [Section 9.2, "Configuring a Reverse Proxy for OracleAS Portal and OracleAS Single Sign-On"](#) on page 9-25.

## 6.3 Installing and Configuring the Application Tier

The application tier consists of multiple computers hosting middle tier Oracle Application Server instances in an Oracle Application Server File-Based Farm. Each instance contains multiple Oracle Application Server Containers for J2EE instances, hosting deployed applications. In the complete configuration, requests are balanced among the OC4J instances on the application tier computers to create a performant and fault tolerant application environment. [Figure 2-1, "Enterprise Deployment Architecture for myJ2EECompany.com"](#) on page 2-4, shows the application tier (APPHOST1 and APPHOST2).

### 6.3.1 Installing the First Application Tier Application Server Instance on APPHOST1

Follow these steps to install the first Oracle Application Server middle tier on APPHOST1:

1. Ensure that the system, patch, kernel and other requirements are met as specified in the *Oracle Application Server Installation Guide*. You can find this guide in the Oracle Application Server platform documentation library for the platform and version you are using.
2. Copy the `staticports.ini` file from the `Disk1/stage/Response` directory to a local directory, such as `TMP`. You will provide the path to this file during installation.
3. Edit the `staticport.ini` file to assign the following custom ports:

```
Oracle HTTP Server port = 7777
Oracle HTTP Server Listen port = 7778
Application Server Control port = 1810
```

---

**Notes:** Ensure that these ports are not already in use by any other service on the computer. Using the Static Ports feature to install the the Application Server Tier ensures that the port assignments will be consistent, if the ports are correctly specified in the file and the port is not already in use. If a port is incorrectly specified, the Oracle Universal Installer will assign the default port. If a port is already in use, the Oracle Universal Installer will select the next available port.

See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-2 for more information.

---

4. Start the Oracle Universal Installer as follows:

On UNIX, issue this command: **runInstaller**

On Windows, double-click **setup.exe**

The **Welcome** screen appears.

5. Click **Next**.

On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.

6. Specify the directory you want to be the `oraInventory` directory and the operating system group that has write permission to it.

7. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.

8. Open a window and run the script, following the prompts in the window.

9. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

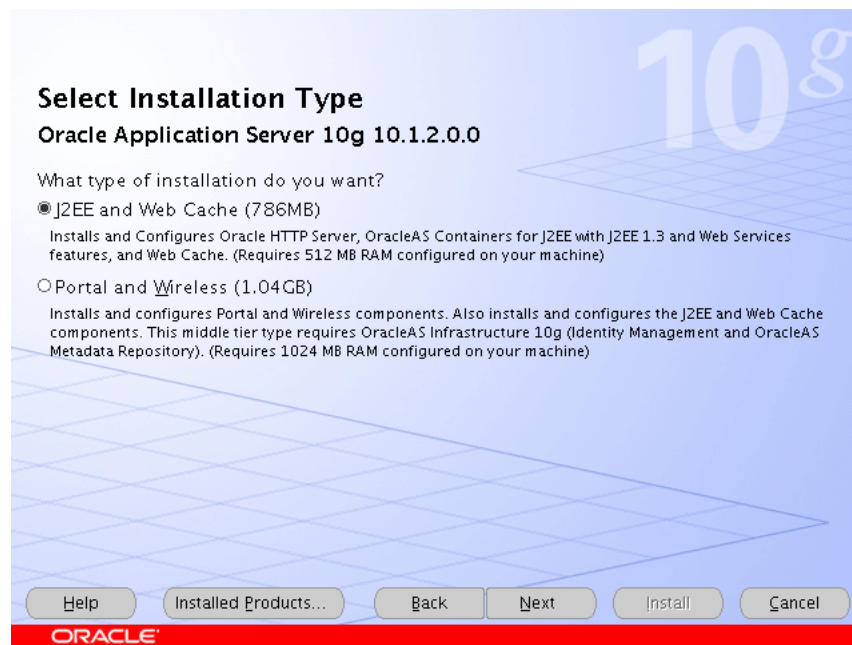
- The product files for installation (Source)
- The name and path to the Oracle home (Destination)

10. Click **Next**.

The **Select a Product to Install** screen appears.

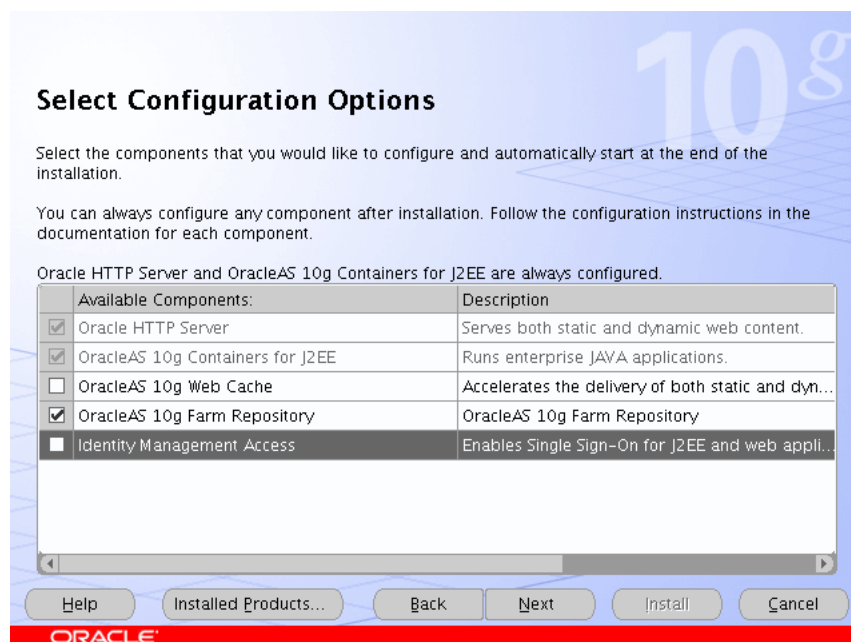
**Figure 6–1 Oracle Universal Installer Select a Product to Install Screen**

11. Select **Oracle Application Server 10g**, as shown in [Figure 6–1](#), and click **Next**.  
The **Select Installation Type** screen appears.

**Figure 6–2 Oracle Universal Installer Select Installation Type Screen**

12. Select **J2EE and Web Cache**, as shown in [Figure 6–2](#), and click **Next**.  
The **Confirm Pre-Installation Requirements** screen appears.
13. Ensure that the requirements are met and click **Next**.
14. The **Select Configuration Options** screen appears.



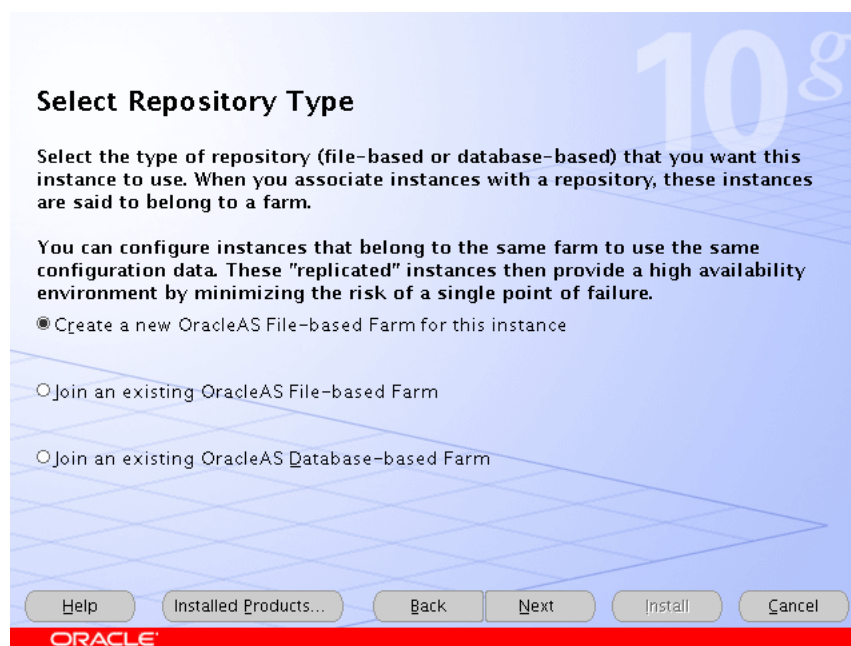
**Figure 6–3 Oracle Universal Installer Select Configuration Options Screen**

15. Select **OracleAS 10g Farm Repository**, as shown in [Figure 6–3](#), and click **Next**.

The **Specify Port Configuration Options** screen appears.

16. Select **Manual**, specify the location of the `staticports.ini` file, and click **Next**.

The **Select Repository Type** screen appears.

**Figure 6–4 Oracle Universal Installer Select Repository Type Screen**

17. Select **Create a new OracleAS File-based Farm for this instance**, as shown in [Figure 6–4](#), and click **Next**.

The **Specify Instance Name and ias\_admin Password** screen appears.

18. Specify an instance name and the Oracle Application Server administrator's password and click **Next**.

The **Summary** screen appears.

19. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `root.sh` script.

20. Open a window and run the script, following the prompts in the window.
21. Return to the Oracle Universal Installer screen and click **Next**.

The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.

22. Click **Exit**, and then confirm your choice to exit.
23. Verify that the installation was successful by viewing the application server instance in Oracle Enterprise Manager 10g. Start a browser and access:

**http://hostname:1810**

## 6.3.2 Installing the Second Application Tier Application Server Instance on APPHOST2

Follow these steps to install the second Oracle Application Server middle tier on APPHOST2:

1. Ensure that the system, patch, kernel and other requirements are met as specified in the *Oracle Application Server Installation Guide*. You can find this guide in the Oracle Application Server platform documentation library for the platform and version you are using.
2. Copy the `staticports.ini` file from the `Disk1/stage/Response` directory to a local directory, such as `TMP`. You will provide the path to this file during installation.
3. Edit the `staticport.ini` file to assign the following custom ports:

```
Oracle HTTP Server port = 7777
Oracle HTTP Server Listen port = 7778
Application Server Control port = 1810
```

---

**Notes:** Ensure that these ports are not already in use by any other service on the computer. Using the Static Ports feature to install the Application Server Tier ensures that the port assignments will be consistent, if the ports are correctly specified in the file and the port is not already in use. If a port is incorrectly specified, the Oracle Universal Installer will assign the default port. If a port is already in use, the Oracle Universal Installer will select the next available port.

See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-2 for more information.

---

4. Start the Oracle Universal Installer as follows:

On UNIX, issue this command: **runInstaller**

On Windows, double-click **setup.exe**

The **Welcome** screen appears.

5. Click **Next**.

On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.

6. Specify the directory you want to be the oraInventory directory and the operating system group that has write permission to it.

7. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.

8. Open a window and run the script, following the prompts in the window.

9. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

- The product files for installation (Source)
- The name and path to the Oracle home (Destination)

10. Click **Next**.

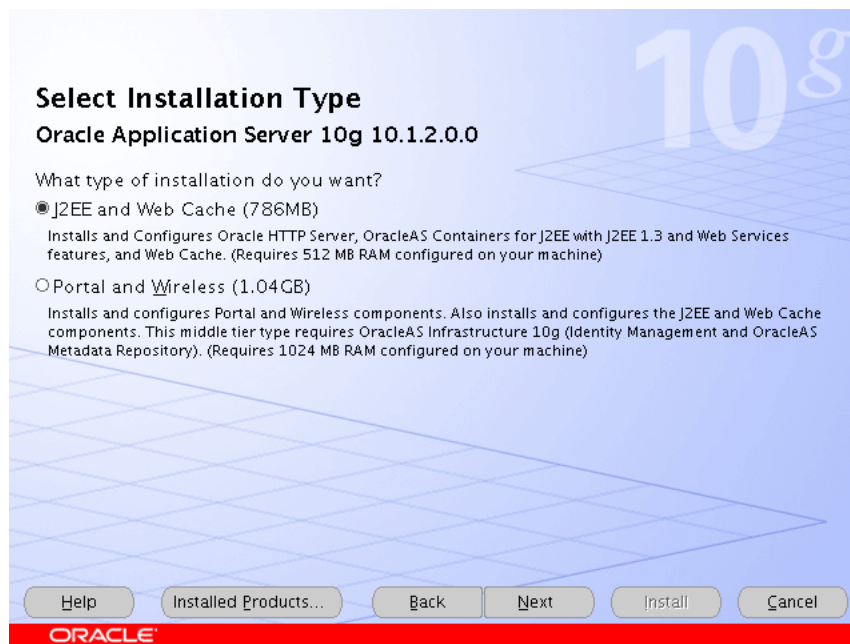
The **Select a Product to Install** screen appears.

**Figure 6–5 Oracle Universal Installer Select a Product to Install Screen**



11. Select **Oracle Application Server 10g**, as shown in [Figure 6–5](#), and click **Next**.

The **Select Installation Type** screen appears.

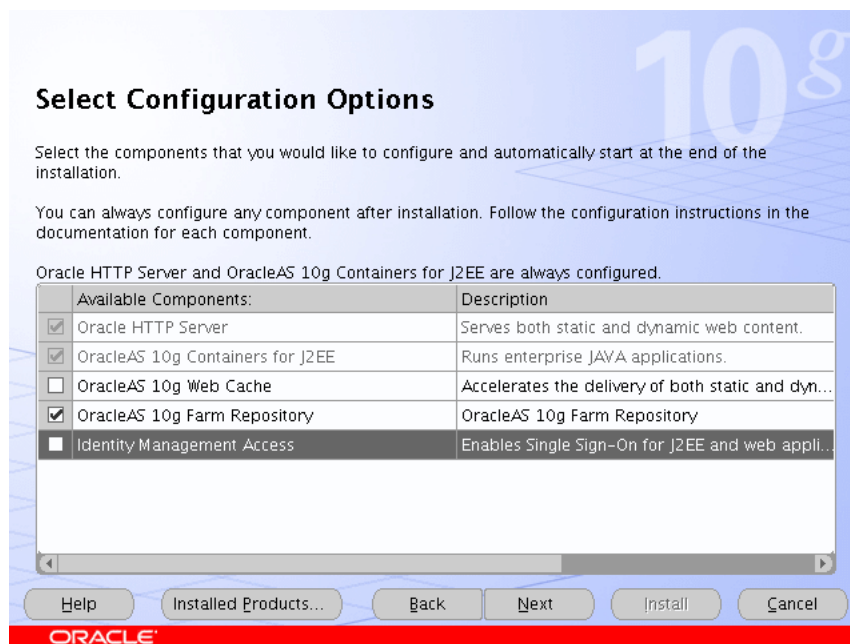
**Figure 6–6 Oracle Universal Installer Select Installation Type Screen**

12. Select **J2EE and Web Cache**, as shown in [Figure 6–6](#), and click **Next**.

The **Confirm Pre-Installation Requirements** screen appears.

13. Ensure that the requirements are met and click **Next**.

The **Select Configuration Options** screen appears.

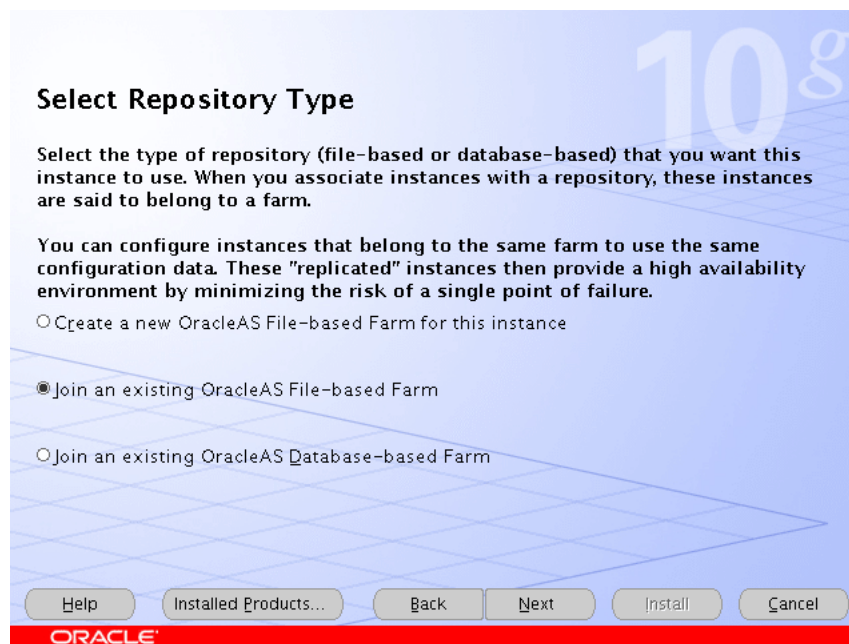
**Figure 6–7 Oracle Universal Installer Select Configuration Options Screen**

14. Select **OracleAS 10g Farm Repository**, as shown in [Figure 6–7](#), and click **Next**.

The **Specify Port Configuration Options** screen appears.

15. Select **Manual**, specify the location of the `staticports.ini` file, and click **Next**.

**Figure 6–8 Oracle Universal Installer Select Repository Type Screen**



16. Select **Join an existing OracleAS File-based Farm**, as shown in [Figure 6–8](#), and click **Next**.

The **Specify File-based Farm Repository** screen appears.

17. Ensure that the DCM daemon is running on APPHOST1 by following these steps:

- a. Open a window and issue this command in `APPHOST1_ORACLE_HOME/opmn/bin`:

```
opmnctl status
```

- b. Verify that the dcm-daemon appears as below (status **Alive**):

```
Processes in Instance: OrclAS1.apphost1.mycompany.com
-----+-----+-----+-----+
ias-component | process-type | pid | status
-----+-----+-----+-----+
LogLoader    | logloaderd   | N/A | Down
dcm-daemon   | dcm-daemon   | 28685 | Alive
DSA          | DSA          | N/A | Down
HTTP_Server  | HTTP_Server  | 28802 | Alive
OC4J         | home         | 28810 | Alive
```

- c. If the dcm-daemon status is **Down**, issue this command:

```
opmnctl startproc ias-component=dcm-daemon
```

18. Return to the Oracle Universal Installer and specify the host name of APPHOST1, and the DCM Discovery Port on which the OracleAS File-based Farm Repository listens, and click **Next**.

---

**Note:** The port range 7100-7179 is used for communication between DCM instances. The first installed instance of an OracleAS File-Based Farm on a computer has port 7100 assigned as its DCM Discovery Port. A subsequently installed instance will use port 7101, and so on.

---

The **Specify Instance Name and ias\_admin Password** screen appears.

19. Specify an instance name and the Oracle Application Server administrator's password and click **Next**.

The **Summary** screen appears.

20. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `root.sh` script.

21. Open a window and run the script, following the prompts in the window.
22. Return to the Oracle Universal Installer screen and click **Next**.

The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.

23. Click **Exit**, and then confirm your choice to exit.
24. Verify that the installation was successful by viewing the application server instance in Oracle Enterprise Manager 10g. Start a browser and access:

**http://hostname:1810**

### 6.3.3 Creating OC4J Instances on the Application Tier

Follow the steps in this section on APPHOST1 only to create OC4J instances. The instances you create will be replicated to APPHOST2 when you join the instances to a DCM-Managed OracleAS Cluster, joining APPHOST1 first. The first member of the DCM-Managed OracleAS Cluster provides the base configuration to the entire cluster.

1. On the **Oracle Enterprise Manager 10g Farm** page, select the APPHOST1 instance.

The **Application Server** page for the instance appears.

2. Click **Create OC4J Instance**.

The **Create OC4J Instance** page appears.

3. Enter the name for the OC4J instance and click **Create**.

---

**Note:** Do not use a host name, Oracle home, or an IP address in the OC4J instance name.

---

A confirmation screen appears.

4. Click **OK**.

The **Application Server** page appears.

### 6.3.4 Deploying J2EE Applications

Follow the steps in this section on APPHOST1 only to deploy applications. The applications you deploy will be replicated to APPHOST2 when you join the instances to a DCM-Managed OracleAS Cluster, joining APPHOST1 first. The first member of the DCM-Managed OracleAS Cluster provides the base configuration to the entire cluster.

Before you perform the steps in this section, you must perform the steps in [Section 5.2, "Option 2: Using the Oracle Application Server Java Authentication and Authorization Service \(JAAS\) Provider"](#). Otherwise, JAZN LDAP User Manager will not appear as a selection so that you can perform Step 8.

1. On the **Oracle Enterprise Manager 10g Farm** page, select the APPHOST1 instance.  
The **Application Server** page for the instance appears.
2. Click the link for the OC4J instance for the application deployment.  
The page for the OC4J instance appears.
3. Click the **Applications** link.  
The **Applications** page for the OC4J instance appears.
4. Click **Deploy EAR File**.  
The **Deploy Application** page appears.
5. Click **Browse** and navigate to the EAR file you want to deploy.  
The **J2EE Application** field is populated with the path to the EAR file.
6. Complete the **Application Name** field and click **Continue**.  
The **Deploy Application: URL Mapping for Web Modules** screen appears.
7. Specify the URL mapping for the application and click **Next**.  
The **Deploy Application: User Manager** screen appears.
8. Select **Use JAZN LDAP User Manager** and click **Next**.  
The **Deploy Application: Review** screen appears, with the name of the EAR file to deploy, the deployment destination instance, and the URL mapping specified. (If you need to change any information, you can click the **Back** button to navigate to the previous screen).
9. Click **Deploy**.  
A confirmation screen appears.
10. Click **OK**.  
The **Applications** page for the OC4J instance appears with the application in the **Deployed Applications** table.
11. Modify the `ORACLE_HOME/j2ee/oc4j`  
`instance/application-deployments/application`  
`name/orion-application.xml` file to remove `auth-method="SSO"` from the `<jazn>` tag.



---

**Note:** By default, when an application is deployed using Oracle Enterprise Manager 10g to specify use of the JAZN LDAP User Manager, Application Server Control Console automatically sets the auth-method to "SSO", so you must remove the auth-method="SSO" when OracleAS Single Sign-On is not used for authentication.

---

## 6.3.5 Creating a DCM-Managed Oracle Application Server Cluster on the Application Tier

The Oracle Application Server instances on the Application Tier can be treated as one entity by clients and the system administrator if they belong to a DCM-Managed OracleAS Cluster.

The Oracle Application Server Farm (to which all of the application server instances belong, currently as standalone instances) was created during installation. Creating a cluster and its member instances is a two-step process: first, you create the cluster, then, you join instances to it.

### 6.3.5.1 Creating the DCM-Managed OracleAS Cluster

Follow these steps on the Application Tier to create a DCM-Managed OracleAS Cluster:

1. On the **Oracle Enterprise Manager 10g Farm** page, click **Create Cluster**.

The **Create Cluster** page appears.

2. Enter the cluster name and click **Create**.

A confirmation screen appears.

3. Click **OK**.

The **Farm** page appears.

### 6.3.5.2 Joining Application Server Instances to the DCM-Managed OracleAS Cluster

Follow these steps on the Application Tier to join the Oracle Application Server instances to the DCM-Managed OracleAS Cluster on APPHOST1:

1. On the **Oracle Enterprise Manager 10g Farm** page, select the APPHOST1 instance.

---

**Note:** The first instance to join a cluster provides the base configuration for the cluster. The base configuration is always applied to all instances that join the cluster subsequently. APPHOST1 is joined to the cluster first, so that APPHOST2 will inherit APPHOST1's configuration when APPHOST2 joins the cluster.

---

2. Click **Join Cluster**.

The **Join Cluster** page appears.

3. Select the cluster created in [Section 6.3.5.1](#) and click **Join**.

A confirmation screen appears.

4. Click **OK**.

The **Farm** page appears.



5. Start the cluster created in [Section 6.3.5.1](#).
6. Start the APPHOST1 instance.
7. Select the APPHOST2 instance.
8. Click **Join Cluster**.  
The **Join Cluster** page appears.
9. Select the cluster created in [Section 6.3.5.1](#) and click **Join**.  
A confirmation screen appears.
10. Click **OK**.  
The **Farm** page appears.
11. Start the APPHOST2 instance.
12. Verify that the OC4J applications deployed on APPHOST1 are accessible from APPHOST2.

### 6.3.6 Modifying the Oracle Enterprise Manager 10g Application Server Control Console Welcome Page

You must modify the Oracle Enterprise Manager 10g Application Server Control Console to prevent display of internal server names. Follow the instructions on [Section 7.3.9, "Modifying the Oracle Application Server Welcome Page"](#) on page 7-65.

## 6.4 Installing and Configuring the Web Tier

The Web Tier consists of multiple standalone Oracle HTTP Servers, which route requests to the OC4J instances on the application tier computers.

### 6.4.1 Installing the Oracle HTTP Servers on WEBHOST1 and WEBHOST2

Obtain the standalone Oracle HTTP Server from the Oracle Application Server Companion CD, included in the Oracle Application Server CD Pack. Follow these steps to install an Oracle HTTP Server on WEBHOST1 and WEBHOST2:

1. Start the Oracle Universal Installer as follows:  
On UNIX, issue this command: **runInstaller**  
On Windows, double-click **setup.exe**  
The **Welcome** screen appears.
2. Click **Next**.  
On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.
3. Specify the directory you want to be the oraInventory directory and the operating system group that has write permission to it.
4. Click **Next**.  
On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.
5. Open a window and run the script, following the prompts in the window.
6. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

- The product files for installation (Source)
- The name and path to the Oracle home (Destination)

7. Click **Next**.

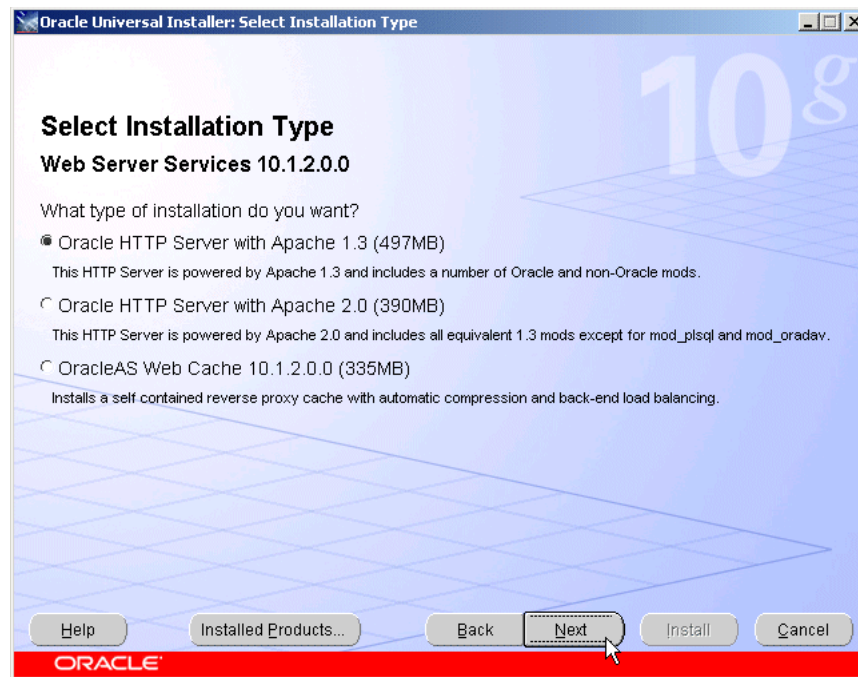
The **Select a Product to Install** screen appears.

**Figure 6–9 Oracle Universal Installer Select a Product to Install Screen**



8. Select **Web Server Services**, as shown in [Figure 6–9](#), and click **Next**.

The **Select Installation Type** screen appears.

**Figure 6–10 Oracle Universal Installer Select Installation Type Screen**

9. Select **Oracle HTTP Server with Apache 1.3** and click **Next**.

The **Summary** screen appears.

10. Click **Install**.

The **Install** screen appears. When processing completes, the **Next** button activates.

11. Click **Next**.

The **Configuration Assistants** screen appears. When the configuration completes, the **End of Installation** screen appears.

12. Click **Exit**, and then confirm your choice to exit.

13. Verify that the installation was successful by viewing the Oracle HTTP Server server home page. Start a browser and access **http://hostname:7777**.

## 6.5 Configuring the Manually Managed Oracle Application Server Cluster

To enable communication between the Web Server Tier and the Application Tier, you must create a Manually Managed Oracle Application Server Cluster of the standalone Oracle HTTP Servers and the DCM-Managed OracleAS Cluster on the Application Tier. You do this by editing the `ons.conf` file, the configuration file for the Oracle Notification Server component of Oracle Process Manager and Notification Server. The Oracle Notification Server is the transport mechanism for communication between Oracle Application Server components. It operates according to a publish-subscribe model, in which a component receives notifications through its subscription to ONS. For a complete description of OPMN functionality, see the *Oracle Process Manager and Notification Server Administrator's Guide*.

The `ons.conf` file on WEBHOST1 and WEBHOST2 must contain the hostname and Oracle Notification Server remote listening port of each server in the Manually Managed OracleAS Cluster. For example, the `ons.conf` file for the DCM-Managed OracleAS Cluster for myJ2EE would resemble the following:

```
nodes=apphost1.mycompany.com:6201,apphost2.mycompany.com:6202,
webhost1.mycompany.com:6200,webhost2.mycompany.com:6203
```

The ONS remote listening port of each server is identified in the `opmn.xml` file by the `remote` attribute of the `notification-server` element, shown in bold in the example `opmn.xml` file for `webhost1.mycompany.com`.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <opmn xmlns="http://www.mycompany.com/ias-instance">
- <notification-server>
  <port local="6100" remote="6200" request="6003" />
  <log-file path="$ORCL_HOME\opmn\logs\ons.log" level="4"rotation-size=.../>
  <ssl enabled="true" wallet-file="$ORACLE_HOME\opmn\conf\ssl.wlt\default" />
</notification-server>
```

Follow these steps to configure the Manually Managed OracleAS Cluster:

1. Copy the `ons.conf` file from APPHOST1 to WEBHOST1.
2. Add the host names for WEBHOST1 and WEBHOST2 to the file.
3. Copy the file to WEBHOST2.
4. Reload OPMN on WEBHOST1 and WEBHOST2 by issuing this command:

```
opmnctl reload
```

## 6.6 Configuring the Oracle HTTP Server with the Load Balancing Router

This procedure associates incoming requests with the Load Balancing Router hostname and port in the `myJ2EECompany` configuration shown in [Figure 2-1](#). Perform these steps on WEBHOST1 and WEBHOST2:

1. Open the Oracle HTTP Server configuration file:
 

```
ORACLE_HOME/Apache/Apache/conf/httpd.conf
```
2. Perform the following steps:
  - a. Add the `LoadModule certheaders_module` directive for the appropriate platform.

UNIX:

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

Windows:

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

- b. Add the following lines to create a `NameVirtualHost` directive and a `VirtualHost` container for `myapp.mycompany.com` and port 443.

```
NameVirtualHost *:7777
<VirtualHost *:7777>
  ServerName myapp.mycompany.com
  Port 443
  ServerAdmin you@your.address
  RewriteEngine On
  RewriteOptions inherit
  SimulateHttps On
</VirtualHost>
```

---

**Notes:** The `LoadModule` directives (in particular, the `LoadModule rewrite_module` directive) must appear in the `httpd.conf` file at a location preceding the `VirtualHost` directives. The server must load all modules before it can execute the directives in the `VirtualHost` container.

It is a good idea to create the `VirtualHost` directives at the end of the `httpd.conf` file.

---

3. Save the `httpd.conf` file.
4. Restart the components using these commands in `ORACLE_HOME/opmn/bin`:

```
opmnctl stopall
```

```
opmnctl startall
```

## 6.7 Configuring OC4J Routing

`mod_oc4j`, an Oracle HTTP Server module, performs the request routing to the OC4J instances over the AJP13 protocol. The routing configuration is specified in the `mod_oc4j.conf` file. (The `mod_oc4j.conf` file is referenced by the main server configuration file for Oracle HTTP Server, `httpd.conf`, with an `Include` directive.) The `mod_oc4j.conf` file is located in:

```
ORACLE_HOME/Apache/Apache/conf/mod_oc4j.conf
```

For complete descriptions of all directives and their uses, see the *Oracle HTTP Server Administrator's Guide*.

The default file at installation resembles [Example 6-1](#):

### Example 6-1 `mod_oc4j.conf` File

```
LoadModule oc4j_module modules/ApacheModuleOc4j.dll
<IfModule mod_oc4j.c>
  <Location /oc4j-service>
    SetHandler oc4j-service-handler
    Order deny,allow
    Deny from all
    Allow from localhost my-pc.mycompany.com my-pc
  </Location>

  Oc4jMount /j2ee/*
  Oc4jMount /webapp home
  Oc4jMount /webapp/* home
  Oc4jMount /cabo home
  Oc4jMount /cabo/* home
  Oc4jMount /IsWebCacheWorking home
  Oc4jMount /IsWebCacheWorking/* home
</IfModule>
```

Before you configure `mod_oc4j.conf` on `WEBHOST1` and `WEBHOST2`, copy the `mod_oc4j.conf` file from `APPHOST1` to `WEBHOST1`.

Follow these steps on `WEBHOST1`:

1. Open the `ORACLE_HOME/Apache/Apache/conf/mod_oc4j.conf` file.

2. Add an `Oc4JConnTimeout` directive to specify a time out value smaller than the time out value used by the firewall between the Web tier and the Application Tier. For example:

```
Oc4JConnTimeout 10
```

3. Modify the `Oc4JMount` directives to specify the cluster to which requests should be load balanced. [Example 6–2](#) shows the directive for routing to a cluster.

The syntax for the `Oc4JMount` directive is:

```
Oc4JMount path [destination]
```

*path* is the context root of the application and *destination* is an `ajp13` destination, a cluster, or an instance. `cluster` is the default destination type. [Example 6–2](#) shows complete syntax of the directive for a cluster destination, the default destination type. It is not necessary to specify the OC4J instance when routing requests to applications deployed in the home OC4J instance.

**Example 6–2 OC4JMount Directive to Route to FAQApp in the J2EEApps cluster**

```
Oc4JMount /FAQApp/* cluster://myCluster:myOC4JInstance
```

**Example 6–3 OC4JMount Directive to Load Balance Requests to FAQApp on Multiple Instances**

```
Oc4JMount /FAQApp/* instance://myOracleASInstance:myOC4JInstance,  
anotherHost:anotherOracleASInstance:anotherOC4JInstance...
```

**Example 6–4 OC4JMount Directive to Route to FAQApp Using the AJP13 Protocol**

```
Oc4JMount /FAQApp/* ajp13://myHost:8888
```

4. Save and close the file.
5. Copy the file from WEBHOST1 to WEBHOST2.
6. Restart the Oracle HTTP Server on WEBHOST1 and WEBHOST2.

## 6.8 Configuring Application Authentication and Authorization

The Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider (also referred to as JAZN) LDAP-based provider is used for authentication and authorization to the OC4J applications.

In the `myJ2EECompany` configuration, this provider is used without Oracle Application Server Single Sign-On, because communication to the data tier is prohibited (Oracle Application Server Single Sign-On requires Portal Services access to the database). This section explains how to configure the Oracle Application Server instances on the application tier to use the JAZN LDAP provider.

For instructions on how to use Oracle Enterprise Manager 10g to manage the data in this provider, see Chapter 8 in the *Oracle Application Server Containers for J2EE Security Guide*.

To configure an Oracle Application Server instance to use the JAZN LDAP provider, follow the instructions in [Section 5.2, "Option 2: Using the Oracle Application Server Java Authentication and Authorization Service \(JAAS\) Provider"](#).

---

# Installing and Configuring the myPortalCompany Application Infrastructure

This chapter provides instructions for creating the Application and Web Server tiers of the myPortalCompany architecture, distributing the software components into the DMZs shown in the Enterprise Deployment architecture depicted in [Figure 2-2](#) on page 2-6.

Before you perform the tasks in this chapter, a two-node Real Application Clusters (RAC) database must be installed. In this chapter, the server names for the database hosts are APPDBHOST1 and APPDBHOST2. Ideally, these are separate physical databases from INFRADBHOST1 and INFRADBHOST2. In addition to isolating the security components, separate application databases provide the flexibility needed to maintain and tune application and security parameters separately.

This chapter contains the following topics:

[Section 7.1, "Installing the Metadata Repository for the Application Infrastructure"](#)

[Section 7.2, "Configuring the Load Balancing Router or Proxy Server"](#)

[Section 7.3, "Installing the Application Tier"](#)

[Section 7.4, "Testing the Application Server Tier"](#)

[Section 7.5, "Configuring Custom Java Portal Development Kit \(JSDK\) Providers"](#)

[Section 7.6, "Setting the OracleAS Single Sign-On Query Path URL for External Applications"](#)

---

**Note:** For detailed information on OracleAS Portal and its configurations, see the *Oracle Application Server Portal Configuration Guide*.

---

## 7.1 Installing the Metadata Repository for the Application Infrastructure

You must install the OracleAS Metadata Repository before you install components into the Application Infrastructure. Oracle Application Server provides a tool, the Oracle Application Server Metadata Repository Creation Assistant, to create the OracleAS Metadata Repository in an existing database.

The OracleAS Metadata Repository Creation Assistant is available on the OracleAS Metadata Repository Creation Assistant CD-ROM or the Oracle Application Server DVD-ROM. You install the OracleAS Metadata Repository Creation Assistant in its own, separate Oracle home.



To install the OracleAS Metadata Repository, you must perform these steps:

1. Install the OracleAS Metadata Repository Creation Assistant, following the steps in [Section 4.1.1](#).
2. Ensure that the database meets the requirements specified in the "Database Requirements" section of the *Oracle Application Server Metadata Repository Creation Assistant User's Guide*. You can find this guide in the Oracle Application Server platform documentation library for the platform and version you are using. In addition, ensure that:
  - The database computer has at least 512 MB of swap space available for execution of the OracleAS Metadata Repository Creation Assistant
  - There are no dependencies of any kind related to the `ultrasearch` directory in the database's Oracle home. The OracleAS Metadata Repository Creation Assistant replaces this directory with a new version, renaming the existing version of the directory to `ultrasearch_timestamp`.
3. Execute the OracleAS Metadata Repository Creation Assistant, following the steps in [Section 4.1.2](#) or [Section 4.1.3](#).
  - To install into a database using raw devices, follow the steps in [Section 7.1.1, "Installing the Metadata Repository in a Database Using Raw Devices"](#) on page 7-4.
  - To install into a database using Oracle Cluster File System, follow the steps in [Section 7.1.2, "Installing the Metadata Repository in an Oracle Cluster File System \(OCFS\)"](#) on page 7-7.
4. Perform the post-installation step described in [Section 4.1.4](#).

### 7.1.1 Installing the Metadata Repository in a Database Using Raw Devices

Follow these steps to install the Metadata Repository into an existing two-node Real Application Clusters (RAC) database using raw devices:

1. Create raw devices for the Oracle Application Server Metadata Repository, using the values in [Section B.2, "Tablespace Mapping to Raw Devices Sample File"](#) on page B-1.

**Tip:** The command to create tablespaces is specific to the volume manager used. For example, the command to create a tablespace in VERITAS Volume Manager is `vxassist`.

2. Create a file to map the tablespaces to the raw devices. Each line in the file has the format:

```
tablespace name=raw device file path
```

You can use the sample file shown in [Example B-1, "Tablespace to Raw Device Mapping \(Sample File\)"](#) on page B-2, replacing the file paths with the paths on your system. Append a 1 to the tablespace names, as shown in the sample file.

---

---

**Note:** Creating the sample file is not mandatory; you can enter the tablespace values into the Specify Tablespace Information screen during execution of the OracleAS Metadata Repository Creation Assistant.

---

---



3. Populate the `DBCA_RAW_CONFIG` environment variable with the full path and filename of the tablespace mapping file.
4. Ensure that the database and listener are running.
5. Ensure that the `NLS_LANG` environment variable is not set to a non-English locale, or is set to `american_america.us7ascii`, with one of the following commands:

- **`unsetenv NLS_LANG`**
- **`setenv NLS_LANG american_america.us7ascii`**

---

**Note:** If you need to, you can set `NLS_LANG` to its original value after executing the OracleAS Metadata Repository Creation Assistant.

---

6. Start the OracleAS Metadata Repository Creation Assistant from the OracleAS Metadata Repository Creation Assistant Oracle home with this command:

**`runRepca`**

The **Welcome** screen appears.

7. Click **Next**.

The **Specify Oracle Home Path** screen appears.

8. In the **Oracle Home** field, specify the full path of the database Oracle home.

In the **Log File Directory** field, specify the full path of the directory on the current computer in which you want the OracleAS Metadata Repository Creation Assistant to write its log files. Ensure correct input for the **Log File Directory** on this screen, as you will not be able to change it after you have proceeded beyond this screen.

9. Click **Next**.

The **Select Operation** screen appears.

10. Select **Load and Register** and click **Next**.

The **Specify Database Connect** screen appears.

11. Enter the SYS user name and password and the host and port information. For example:

```
infradbhost1.mycompany.com:1521,infradbhost2.mycompany.com:1521
```

12. Click **Next**.

The **Specify Storage Options** screen appears.

13. Select **Regular or Cluster File System**.

The **Specify Tablespace Information** screen appears, displaying the values from the file specified by the `DBCA_RAW_CONFIG` environment variable.

14. Correct the values, if necessary, and click **Next**.

The **Warning: Check Disk Space** dialog appears if your `SYSTEM` and `UNDO` tablespaces are set to `autoextend`.

15. Check the disk space as specified in the dialog and click **OK**.

The **Specify Oracle Internet Directory Connect** screen appears.

16. Enter the virtual host name for the Oracle Internet Directory, `oid.mycompany.com`, and port 389.

The **Specify Login for Oracle Internet Directory** screen appears.

17. Enter the user name and password to log in to Oracle Internet Directory. Note that:
  - The user must belong to the `iASAdmins` group.
  - You can provide the user's simple name (for example, `jdoe`) or the user's Distinguished Name (DN) (for example, `cn=orcladmin`).
  - If the Oracle Internet Directory has multiple realms, you must enter the realm that contains the specified user. (The realm value is not used if you log in as `cn=orcladmin`, since the superuser does not belong to any realm.)

18. Click **Next**.

The **Specify Oracle Context** screen appears.

19. Specify the location in Oracle Internet Directory in which the OracleAS Metadata Repository will be installed, and click **Next**.

The **Loading Repository** screen appears. The tablespaces and schemas are created and populated.

The **Success** screen appears.

20. Click **OK**.

The OracleAS Metadata Repository Creation Assistant exits.

## 7.1.2 Installing the Metadata Repository in an Oracle Cluster File System (OCFS)

Follow these steps to install the Metadata Repository into an existing two-node Real Application Clusters (RAC) database using an OCFS file system:

1. Ensure that the database and listener are running.
2. Start the OracleAS Metadata Repository Creation Assistant from the OracleAS Metadata Repository Creation Assistant Oracle home with this command:

```
runRepca
```

The **Welcome** screen appears.

3. Click **Next**.

The **Specify Oracle Home Path** screen appears.

4. In the **Oracle Home** field, specify the full path of the database Oracle home.

In the **Log File Directory** field, specify the full path of the directory on the current computer in which you want the OracleAS Metadata Repository Creation Assistant to write its log files. Ensure correct input for the **Log File Directory** on this screen, as you will not be able to change it after you have proceeded beyond this screen.

5. Click **Next**.

The **Select Operation** screen appears.

6. Select **Load and Register** and click **Next**.

The **Specify Database Connect** screen appears.

7. Enter the SYS user name and password and the host and port information. For example:

```
infradbhost1.mycompany.com:1521,infradbhost2.mycompany.com:1521
```

8. Click **Next**.

The **Specify Storage Options** screen appears.

9. Select **Regular or Cluster File System**.

The **Specify Tablespace Information** screen appears.

10. Select a directory option (**Use Same Directory for All Tablespaces** or **Use Individual Directories for Each Tablespace**) and complete the remaining fields. When specifying a directory, ensure that it is an existing, writable directory with sufficient free space. Click **Next**.

The **Warning: Check Disk Space** dialog appears if your SYSTEM and UNDO tablespaces are set to autoextend.

11. Check the disk space as specified in the dialog and click **OK**.

The **Specify Oracle Internet Directory Connect** screen appears.

12. Enter the virtual host name for the Oracle Internet Directory, `oid.mycompany.com`, and port 389.

The **Specify Login for Oracle Internet Directory** screen appears.

13. Enter the user name and password to log in to Oracle Internet Directory. Note that:

- The user must belong to the iASAdmins group.
- You can provide the user's simple name (for example, `jdoe`) or the user's Distinguished Name (DN) (for example, `cn=orcladmin`).
- If the Oracle Internet Directory has multiple realms, you must enter the realm that contains the specified user. (The realm value is not used if you log in as `cn=orcladmin`, since the superuser does not belong to any realm.)

14. Click **Next**.

The **Specify Oracle Context** screen appears.

15. Specify the location in Oracle Internet Directory in which the OracleAS Metadata Repository will be installed, and click **Next**.

The **Loading Repository** screen appears. The tablespaces and schemas are created and populated.

The **Success** screen appears.

16. Click **OK**.

The OracleAS Metadata Repository Creation Assistant exits.

## 7.2 Configuring the Load Balancing Router or Proxy Server

If you are using a Load Balancing Router, it must be configured to enable the following:

- A virtual IP address (VIP1) that listens for requests to portal.mycompany.com on port 443 (an HTTPS listening port), and balances them to the Application tier OracleAS Web Cache running on APPHOST1 port 7777 (an HTTP listening port). You must configure the Load Balancing Router to perform the protocol conversion.
- The virtual IP address VIP1 listens for requests to portal.mycompany.com on port 7777 (an HTTP listening port), and balances them to the Application tier OracleAS Web Cache on APPHOST1 port 7777 (an HTTP listening port). Port 7777 on the Load Balancing Router receives the HTTP loop-back requests made by the Parallel Page Engine on APPHOST1. This 7777 port also receives requests from the Portal Metadata Repository for web provider design time messages. This configuration may require a Network Address Translation (NAT) rule in the Load Balancing Router in order for the loop-back request from the PPE to succeed.

---

---

**Note:** For security reasons, port 7777 on the Load Balancing Router should not be visible to external users.

---

---

- The virtual IP address VIP1 listens for requests to portal.mycompany.com on port 9401 (an HTTP listening port), and balances them to the Application Tier OracleAS Web Cache on APPHOST1 port 9401 (an HTTP listening port). Port 9401 port on the Load Balancing Router receives invalidation messages from the OracleAS Portal Repository when content that is cached in OracleAS Web Cache becomes stale. This configuration might require a Network Address Translation (NAT) rule in the Load Balancing Router in order for the invalidation requests from the OracleAS Portal repository to succeed.

---

---

**Note:** VIP1 listens on 443 for external traffic, on port 7777 for Parallel Page Engine loop-back messages, and port 9401 for invalidation messages.

For security reasons, port 9401 on the Load Balancing Router should not be visible to external users.

---

---

- HTTP monitoring of OracleAS Web Cache. The Load Balancing Router must be configured to detect an inoperative computer and stop routing requests to it until it is functioning again. Two OracleAS Web Cache ports must be monitored: the HTTP request port and the invalidation port.

To monitor port 7777, use the following URL in the Load Balancing Router configuration:

*hostname:port/\_oracle\_http\_server\_webcache\_static\_.html*

For example:

*http://apphost1.mycompany.com:7777/\_oracle\_http\_server\_webcache\_static\_.html*

If the Load Balancing Router receives a response from this URL, then the OracleAS Web Cache instance is running. If not, then the process or the server is down, and the Load Balancing Router will forward all requests to the surviving computer.

To monitor port 9401, use the following URL in the Load Balancing Router configuration:

`http://hostname.domain.com:9401`

For example:

`http://apphost1.mycompany.com:9401`

The Load Balancing Router sends an HTTP request to this URL; the response header resembles the following:

`HTTP/1.0`

The Load Balancing Router must be configured to detect the string HTTP in the first line of the response header. Thus, when the Load Balancing Router detects HTTP in the first line of the response header, the invalidation port is available. If not, then all invalidation requests are routed to the surviving computer.

If you are using a proxy server, follow the instructions in [Section 9.2, "Configuring a Reverse Proxy for OracleAS Portal and OracleAS Single Sign-On"](#).

---

---

**Note:** You must also update the `sqlnet.ora` file to prevent connection time outs related to the Load Balancing Router and firewall. See [Section 4.1.5, "Configuring the Time out Value in the `sqlnet.ora` File"](#) on page 4-6.

---

---

## 7.3 Installing the Application Tier

Follow the tasks in this section to install the Application Tier components (APPHOST1 and APPHOST2) into the Application tier:

[Section 7.3.1, "Installing the First Application Server on APPHOST1"](#)

[Section 7.3.3, "Configuring the First Application Server on APPHOST1"](#)

[Section 7.3.4, "Installing the Second Application Server on APPHOST2"](#)

[Section 7.3.5, "Configuring the Second Application Server on APPHOST2"](#)

[Section 7.3.6, "Configuring OracleAS Web Cache Clusters"](#)

[Section 7.3.7, "Configuring Load Balancing and Monitoring"](#)

[Section 7.3.8, "Enabling Session Binding on OracleAS Web Cache Clusters"](#)

[Section 7.3.9, "Modifying the Oracle Application Server Welcome Page"](#)

### 7.3.1 Installing the First Application Server on APPHOST1

Follow these steps to install an Oracle Application Server middle tier on APPHOST1:

1. Ensure that the system, patch, kernel and other requirements are met as specified in the *Oracle Application Server Installation Guide*. You can find this guide in the Oracle Application Server platform documentation library for the platform and version you are using.
2. Copy the `staticport.ini` file from the `Disk1/stage/Response` directory to a local directory, such as TMP.

3. Edit the `staticport.ini` file to assign the following custom ports:

```
Oracle HTTP Server port = 7777
Oracle HTTP Server Listen port = 7778
Web Cache HTTP Listen port = 7777
Web Cache Administration port = 9400
Web Cache Invalidation port = 9401
Web Cache Statistics port = 9402
Application Server Control port = 1810
```

---

**Notes:** Ensure that these ports are not already in use by any other service on the computer. Using the Static Ports feature as described to install the Application Server Tier ensures that the port assignments will be consistent with the documentation in this section, if the ports are correctly specified in the file and the port is not already in use. Otherwise:

- If a port is incorrectly specified, then the Oracle Universal Installer will assign the default port.
- If a port is already in use, then the Oracle Universal Installer will assign the next available port.

See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-2 for more information.

Port 80 is open on the firewall only to accept and redirect requests using the HTTP (non-secure) protocol. Requests using the HTTP protocol (in the form `http://www.mycompany.com`) are redirected to port 443. Requests using the HTTPS, or secure, protocol (in the form `https://www.mycompany.com`) are managed by port 443.

---

4. Start the Oracle Universal Installer as follows:

On UNIX, issue this command: **runInstaller**

On Windows, double-click **setup.exe**

The **Welcome** screen appears.

5. Click **Next**.

On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.

6. Specify the directory you want to be the `oraInventory` directory and the operating system group that has write permission to it.
7. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.

8. Open a window and run the script, following the prompts in the window.
9. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

- The product files for installation (Source)
- The name and path to the Oracle home (Destination)

10. Specify the path and click **Next**.

The **Select a Product to Install** screen appears.

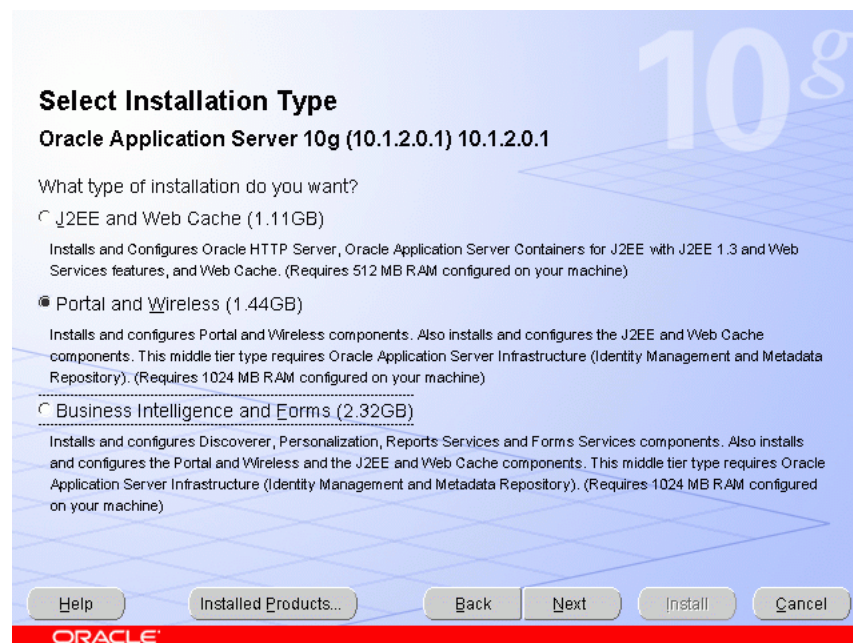
**Figure 7-1 Oracle Universal Installer Select a Product to Install Screen**



11. Select **Oracle Application Server 10g**, as shown in [Figure 7-1](#), and click **Next**.

The **Select Installation Type** screen appears.

**Figure 7-2 Oracle Universal Installer Select Installation Type Screen**

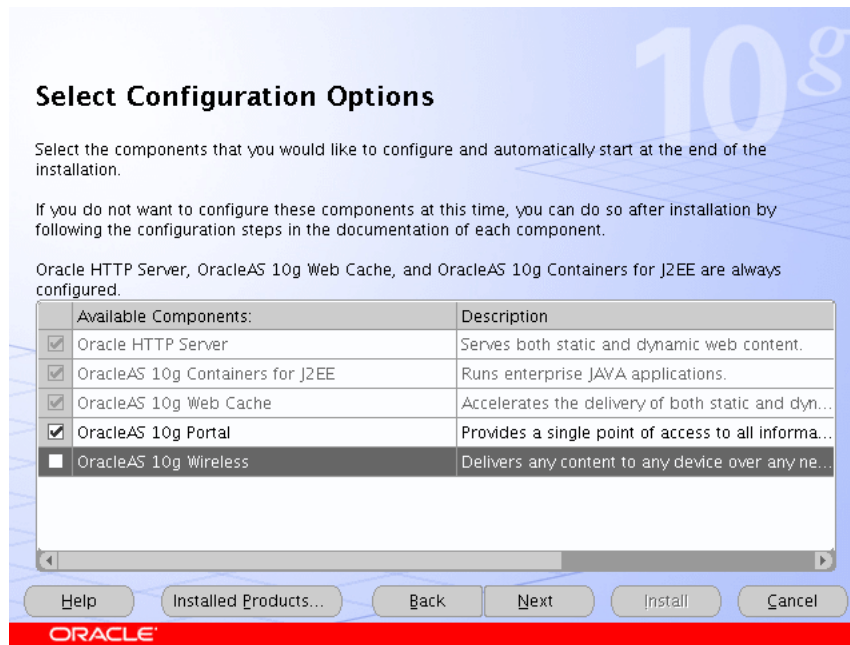


12. Select **Portal and Wireless**, as shown in [Figure 7-2](#), and click **Next**.

The **Confirm Pre-Installation Requirements** screen appears.

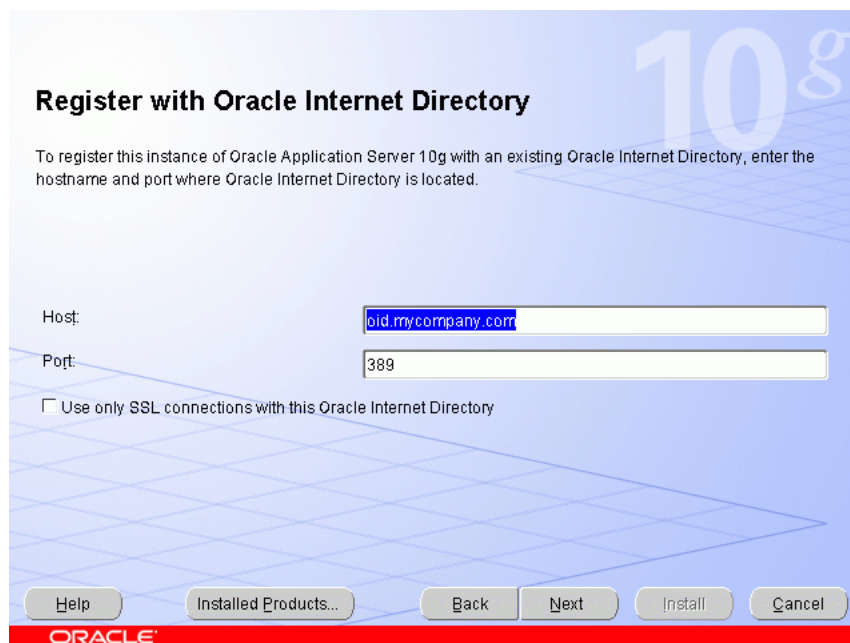
13. Ensure that the requirements are met and click **Next**.
14. The **Select Configuration Options** screen appears.

**Figure 7–3 Oracle Universal Installer Select Configuration Options Screen**



15. Select **OracleAS 10g Portal**, as shown in [Figure 7–3](#), and click **Next**.  
The **Specify Port Configuration Options** screen appears.
16. Select **Manual**, specify the location of the `staticports.ini` file, and click **Next**.
17. The **Register with Oracle Internet Directory** screen appears.

**Figure 7–4 Oracle Universal Installer Register with Oracle Internet Directory Screen**





18. Enter the host name and port of the Oracle Internet Directory load balancing router. Do not select the SSL configuration option.

19. Click **Next**.

The **Specify OID Login** screen appears.

20. Enter the user name and the password and click **Next**.

The **Select OracleAS 10g Metadata Repository** screen appears, displaying a drop-down list of connect strings that the installer detected.

21. Select the connect string for the application Metadata Repository database (on APPDBHOST1 and APPDBHOST2) and click **Next**.

The **Specify Instance Name and ias\_admin Password** screen appears.

22. Specify an instance name and the Oracle Application Server administrator's password and click **Next**.

The **Summary** screen appears.

23. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `root.sh` script.

24. Open a window and run the script, following the prompts in the window.

25. Return to the Oracle Universal Installer screen and click **Next**.

The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.

26. Click **Exit**, and then confirm your choice to exit.

27. Verify that the installation was successful by accessing the OracleAS Portal page at:

**`http://apphost1.mycompany.com:7777/pls/portal`**

28. Access the `ORACLE_HOME/portal/conf/iasconfig.xml` file. The contents of the file are shown in the subsequent example:

```
<IASConfig XSDVersion="1.0">
  <IASInstance Name="portal1.apphost1.mycompany.com"
Host="apphost1.mycompany.com">
    <WebCacheComponent ListenPort="7777" InvalidationPort="9401"
InvalidationUsername="invalidator"
InvalidationPassword="@Bc1McTtma3AWSaWWNcKWrL8My70JWKuGZA=="
SSLEnabled="false" AdminPort="9400"/>
    <EMComponent ConsoleHTTPPort="1810" SSLEnabled="false"/>
  </IASInstance>
  <IASInstance Name="ias-1.oid.mycompany.com" Host="oid.mycompany.com">
    <OIDComponent AdminPassword="@BQawXztGTg7lxfli+kN/597S100HPbDXFQ=="
AdminDN="cn=orcladmin" SSLEnabled="false" LDAPPort="389"/>
  </IASInstance>
  <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="@BRnRWtvlme0pk1rSfv/X5oS3LtsWk8wKTA=="
ConnectString="cn=ptmr,cn=oraclecontext">
    <WebCacheDependency ContainerType="IASInstance"
Name="portal1.apphost1.mycompany.com"/>
    <OIDDependency ContainerType="IASInstance"
Name="ias-1.oid.mycompany.com"/>
    <EMDependency ContainerType="IASInstance"
Name="portal1.apphost1.mycompany.com"/>
  </PortalInstance>
</IASConfig>
```

---

**Note:** The value `portal1` in the `IASInstance` element is the instance name specified in step 22.

---

## 7.3.2 Configuring Load Balancing and Monitoring

The Load Balancing Router must be configured to:

- Balance requests to `portal.mycompany.com` on port 443 (an HTTPS listening port) to the Application tier OracleAS Web Cache running on `APPHOST1` port 7777 (an HTTP listening port).
- Balance requests to `portal.mycompany.com` on port 7777 (an HTTP listening port) to the Application tier OracleAS Web Cache on `APPHOST1` port 7777 (an HTTP listening port). Port 7777 on the Load Balancing Router receives the HTTP loop-back requests made by the Parallel Page Engine on `APPHOST1`. This configuration requires a Network Address Translation (NAT) rule in the Load Balancing Router in order for the loop-back request from the PPE to succeed.
- Balance requests to `portal.mycompany.com` on port 9401 (an HTTP listening port) to the Application Tier OracleAS Web Cache on `APPHOST1` port 9401 (an HTTP listening port). Port 9401 port on the Load Balancing Router receives invalidation messages from the OracleAS Portal Repository when content that is cached in OracleAS Web Cache becomes stale. This configuration might require a Network Address Translation (NAT) rule in the Load Balancing Router in order for the invalidation requests from the OracleAS Portal repository to succeed.

- Monitor OracleAS Web Cache. The Load Balancing Router must be configured to detect an inoperative computer and stop routing requests to it until it is functioning again. Two OracleAS Web Cache ports must be monitored: the HTTP request port and the invalidation port.

Use this URL in the Load Balancing Router configuration to monitor HTTP request port 7777:

`host name:port/_oracle_http_server_webcache_static_.html`

for example:

`http://apphost1.mycompany.com:7777/_oracle_http_server_webcache_static_.html`

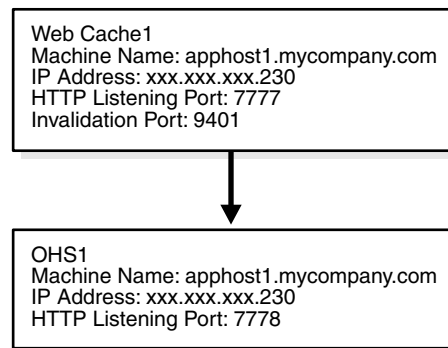
To monitor invalidation port 9401, use this URL:

`http://apphost1.mycompany.com:9401/_oracle_http_server_webcache_static_.html`

### 7.3.3 Configuring the First Application Server on APPHOST1

Upon installation of the first application server, the `iasconfig.xml` file shown at the end of [Section 7.3.1](#) yields an OracleAS Web Cache configuration with the functionality shown in [Figure 7-5](#).

**Figure 7-5 Pre-Configuration Listener Setup on First Application Server**



The configuration of the OracleAS Portal application server tier on APPHOST1 consists of the following tasks:

- [Executing the SSL Configuration Tool on APPHOST1](#)
- [Re-Setting the Oracle Enterprise Manager 10g Link](#)
- [Configuring the Portal Tools Providers on APPHOST1](#)
- [Re-registering mod\\_osso on APPHOST1](#)
- [Verifying Connectivity for Invalidation Messages from the Database to the OracleAS Web Cache on APPHOST1 through the Load Balancing Router](#)
- [Enabling Monitoring of the Load Balancing Router's OracleAS Portal Host and Port Settings](#)
- [Testing the Configuration on APPHOST1](#)

### 7.3.3.1 Executing the SSL Configuration Tool on APPHOST1

Follow these steps to use the SSL Configuration Tool to configure SSL on APPHOST1:

1. Set the `ORACLE_HOME` environment variable to the Oracle home in which OracleAS Portal resides.
2. Verify that the Oracle Internet Directory server is running by issuing this command in `ORACLE_HOME/bin`:
3. Create a file, `ORACLE_HOME/configMyPortal.xml` file to include the following:

```
<sslconfig>
  <mid_tier>
    <virtual_address ssl="on" host="portal.mycompany.com" port="443" inv_
port="9401" ssl_terminate="lbr"/>
    <lbr loopback_port="7777"/>
  </mid_tier>
</sslconfig>
```

4. Issue this command in `ORACLE_HOME/bin`:

```
./SSLConfigTool -config_w_file ORACLE_HOME/configMyPortal.xml
-opwd orcladmin password-ptl_inv_pwd webcache invalidation
password
```

In the preceding command, *orcladmin password* is the Oracle administrator password, and *webcache invalidation password* is the invalidation password for OracleAS Web Cache.

5. Log in to the OracleAS Single Sign-On Administration page as the Administrator, and use the **Administer Partner Applications** page to delete the entry for the partner application **apphost1.mycompany.com**. The OracleAS Single Sign-On Administration URL is:

`http://login.mycompany.com/pls/orasso`

6. Configure the OmniPortlet and Web Clipping Provider registration URLs to go through the HTTP port of the Load Balancing Router:
  - a. Access the OracleAS Portal page at `https://portal.mycompany.com/pls/portal` and log in as the portal administrator.
  - b. Click the **Navigator** link.
  - c. Click the **Providers** tab.
  - d. Click the **Registered Providers** link.
  - e. Click the **Edit Registration** link for OmniPortlet.
  - f. Click the **Connection** tab and change the beginning of the provider registration URL from `https://portal.mycompany.com/` to `http://portal.mycompany.com:7777/`.
  - g. Perform steps **e** and **f** for the Web Clipping Provider.

### 7.3.3.2 Re-Setting the Oracle Enterprise Manager 10g Link

To prevent access to Oracle Enterprise Manager 10g from the outside, the link provided by OracleAS Portal must be changed back to point to the internal server. To do this, on APPHOST1, issue the following command in *ORACLE\_HOME/portal/conf*:

```
HOME/portal/conf:
```

```
ptlconfig -dad portal -em
```

### 7.3.3.3 Configuring the Portal Tools Providers on APPHOST1

You must configure the OracleAS Portal Tools providers (OmniPortlet and OracleAS Web Clipping) to work in this configuration. Follow these steps on APPHOST1 to configure the Portal Tools Provider:

1. Configure OmniPortlet to use a shared preference store. (By default, the OmniPortlet provider uses the file-based preference store. However, in a multiple middle tier environment, you must use a shared preference store, such as the database preference store *DBPreferenceStore*.) To configure OmniPortlet to use *DBPreferenceStore*, perform the following steps:

- a. Navigate to the directory *ORACLE\_HOME/j2ee/OC4J\_Portal/applications/jpdk/jpdk/doc/dbPreferenceStore*.
- b. Create a user on the database containing the *PORTAL* schema, and grant create resource and connect privileges, using the *create user* and *grant connect* commands in SQL\*Plus. Substitute the actual password in the following command. Do not use the default password of *welcome*, as this poses a security risk.

```
create user prefstore identified by password;
```

```
grant connect, resource to prefstore;
```

- c. Connect as user *prefstore* and execute the *jpdk\_preference\_store2.sql* script by issuing this command:

```
@jpdk_preference_store2
```

- d. Edit the *ORACLE\_HOME/j2ee/OC4J\_Portal/config/data-sources.xml* file to add the entry in the subsequent example:

```
<data-source
  class="com.evermind.sql.DriverManagerDataSource"
  name="omniPortletprefStore"
  location="jdbc/UnPooledConnection"
  xa-location="jdbc/xa/XAConnection"
  ejb-location="jdbc/PooledConnection"
  connection-driver="oracle.jdbc.driver.OracleDriver"
  username="prefstore"
  password="password"
  url="jdbc:oracle:thin:@(description=(address_list=
(address=(host=appdbhost1.mycompany.com)(protocol=tcp)(port=1521))
(address=(host=appdbhost2.mycompany.com)(protocol=tcp)(port=1521))
(load_balance=yes)(failover=yes))(connect_data=(service_name= db9i)))"
  inactivity-timeout="30"
/>
```

---

**Note:** Embedding passwords in deployment and configuration files poses a security risk. If you do not want to use a clear text password in the `data-sources.xml` file, you can create an indirect password by following the steps in [Section 7.3.3.4, "Creating an Indirect Password"](#) on page 7-32.

---

- e. Edit the `ORACLE_HOME/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet/provider.xml` file to edit the `preferenceStore` tag as shown in the subsequent example:

```
<provider class="oracle.webdb.reformlet.ReformletProvider">
  <vaultId>0</vaultId>
  <session>true</session>
  <preferenceStore
class="oracle.portal.provider.v2.preference.DBPreferenceStore">
    <name>omniPortletprefStore</name>
    <connection>jdbc/PooledConnection</connection>
  </preferenceStore>
```

- f. Restart the OC4J\_Portal instance.

2. Optionally, you can change the settings for the HTTP proxy configuration, or the repository used by OmniPortlet and OracleAS Web Clipping.

You can change the settings on the Portal Tools **Edit Provider** pages accessible from the Portal Tools providers' test pages. The test pages are located at the following URLs:

- OmniPortlet provider test page on APPHOST1:  
**`http://apphost1.mycompany.com:7777/portalTools/omniPortlet/providers/omniPortlet`**
- Web Clipping provider test page on APPHOST1:  
**`http://apphost1.mycompany.com:7777/portalTools/webClipping/providers/webClipping`**

3. Verify that OmniPortlet and the Web Clipping Provider work properly through the HTTP port of the Load Balancing Router, by accessing the test pages at the following URLs:

OmniPortlet Provider:

**`http://portal.mycompany.com:7777/portalTools/omniPortlet/providers/omniPortlet`**

---

**Note:** If the "No Portlets Available" message appears under the **Portlet** Information section in the **OmniPortlet Provider** test page, then the provider may not be configured correctly. Review Step 1 to ensure correct configuration. The **Portlet Information** section should list the following:

OmniPortlet

Simple Parameter Form

---

Web Clipping Provider:

**`http://portal.mycompany.com:7777/portalTools/webClipping/providers/webClipping`**

---

**Note:** If, while accessing the test pages, you are prompted to examine the site's certificate, accept the certificate.

---

#### 7.3.3.4 Creating an Indirect Password

As an alternative to using a cleartext password in the `data-sources.xml` file, you can create an indirect password by following these steps:

1. Edit the `ORACLE_HOME/j2ee/OC4J_Portal/config/jazn-data.xml` file to add the `prefstore` user in the `jazn.com` realm, as shown in bold:

```
<realm>
  <name>jazn.com</name>
  <users>
    <user>
      <name>prefstore</name>
      <display-name>OmniPortlet prefstore</display-name>
      <description>OmniPortlet prefstore</description>
      <credentials>!welcome</credentials>
    </user>
    <user>
      ...
```

---

**Note:** Place the actual password in the `credentials` element, preceded directly by the `!` character. In the preceding example, the password is `'welcome'`. The next time OC4J reads the `jazn-data.xml` file, it will rewrite the file with the password obfuscated.

---

2. Edit the `ORACLE_HOME/j2ee/OC4J_Portal/config/data-sources.xml` file again to use the indirect password by replacing the password attribute as follows:

**password="->jazn.com/prefstore"**

---

**Note:** For more information, see the *Oracle Application Server Containers for J2EE Security Guide*.

---

### 7.3.3.5 Re-registering mod\_osso on APPHOST1

1. Set the `ORACLE_HOME` environment variable to the current Oracle home.
2. Edit the SSO registration script `ORACLE_HOME/sso/bin/ssoreg` as shown in [Example 7-1](#), and then execute it. [Example 7-1](#) shows the usage of `ssoreg.sh` on UNIX; on Windows, the script name is `ssoreg.bat`.

---

**Note:** The script shown in [Example 7-1](#) has multiple lines for readability only. When you execute the script, all parameters are on a single continuous line.

---

#### **Example 7-1 ssoreg Usage on UNIX**

```
ORACLE_HOME/sso/bin/ssoreg.sh
-site_name portal.mycompany.com
-mod_osso_url https://portal.mycompany.com
-config_mod_osso TRUE
-oracle_home_path ORACLE_HOME
-config_file ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
-admin_info cn=orcladmin
-virtualhost
```

A partner application, **portal.mycompany.com**, is created.

3. Restart the Oracle HTTP Server by issuing this command:

**ORACLE\_HOME/opmn/bin/opmnctl restartproc process-type=HTTP\_Server**

4. Access the following URL:

**https://login.mycompany.com/pls/orasso**

5. Log in to the OracleAS Single Sign-On Administration page as the Administrator, and use the **Administer Partner Applications** page to delete the entry for the partner application **apphost1.mycompany.com**.



### 7.3.3.6 Verifying Connectivity for Invalidation Messages from the Database to the OracleAS Web Cache on APPHOST1 through the Load Balancing Router

When a cached OracleAS Portal object is modified, the OracleAS Portal metadata repository database sends an invalidation message to OracleAS Web Cache to invalidate that object. Since the target configuration has two instances of OracleAS Web Cache, the invalidation message must be load balanced across both OracleAS Web Cache instances. This is an example of component level load balancing.

Before you proceed with this verification, ensure that messages can be sent from the computer hosting the database to the Load Balancing Router. To do this, issue the following command from INFRADBHOST1 and INFRADBHOST2:

```
telnet portal.mycompany.com 9401
```

Verify that no connection failure message is returned.

### 7.3.3.7 Enabling Monitoring of the Load Balancing Router's OracleAS Portal Host and Port Settings

You must first configure a certificate in Oracle Enterprise Manager 10g on APPHOST1 in order to successfully monitor the OracleAS Portal metrics using the Oracle Enterprise Manager 10g Application Server Control Console. Perform these steps to configure the Application Server Control Console to recognize the Certificate Authority that was used by the Web Site to support HTTPS:

1. Obtain the Certificate of the Web site's Certificate Authority, as follows:
  - a. In Microsoft Internet Explorer, connect to the HTTPS URL of the application server you are attempting to monitor.
  - b. Double-click the lock icon at the bottom of the browser screen, which indicates that you have connected to a secure Web site. The browser displays the **Certificate** dialog box, which describes the Certificate used for this Web site. Other browsers offer a similar mechanism to view the Certificate detail of a Web Site.
  - c. Click the **Certificate Path** tab, and select the first entry in the list of certificates.
  - d. Click **View Certificate** to display a second **Certificate** dialog box.
  - e. Click the **Details** tab in the **Certificate** window.
  - f. Click **Copy to File** to display the **Certificate Manager Export** wizard.
  - g. In the **Certificate Manager Export** wizard, select Base64 encoded X.509 (.CER) as the format you want to export, and save the certificate to a text file with an easily identifiable name, such as `ias_certificate.cer`.
  - h. Open the certificate file using a text editor, and confirm that the content of the certificate file looks similar to the content in the subsequent example:

```
-----BEGIN CERTIFICATE-----
MIIDBzCCAnCgAwIBAgIQTs4NcImNY3JAs5edi/5RkTANBgkqhkiG9w0BAQQFADCB
...
base64 certificate content
...
-----END CERTIFICATE-----
```

2. Update the list of Certificate Authorities, as follows:
  - a. Locate the `b64InternetCertificate.txt` file in the `ORACLE_HOME/sysman/config` directory. This file contains a list of Base64 Certificates.

- b. Edit the `b64InternetCertificate.txt` file and add the contents of the certificate file you just exported to the end of the file, taking care to include all the Base64 text of the certificate, including the `BEGIN` and `END` lines.
- c. Use the `orapki` utility to update the `monwallet` Oracle wallet by issuing the following command:

```
ORACLE_HOME/bin/orapki wallet add -wallet ORACLE_
HOME/sysman/config/monwallet -trusted_cert -cert certificate
location
```

In the preceding command, *certificate location* is the full path to the location of the `ias_certificate.cer` file.

- d. When prompted, enter a password for the `monwallet` wallet file. The default password is `welcome`.

Perform these steps to enable monitoring of the Load Balancing Router's front-end host and port settings for OracleAS Portal:

1. Open the `ORACLE_HOME/sysman/emd/targets.xml` file.
2. Locate the OracleAS Portal targets, for example, `TYPE="oracle_portal"`.
3. Edit the `PortalListeningHostPort` property so that it points to the Load Balancing Router. For example:

```
<Property NAME="PortalListeningHostPort"
VALUE="https://portal.mycompany.com:443" />
```

4. Save and close the `targets.xml` file.
5. Reload the `targets.xml` file in the Application Server Control Console by issuing this command in `ORACLE_HOME/bin`:

```
emctl reload
```

6. Restart the Application Server Control Console by issuing the following commands in `ORACLE_HOME/bin`:

```
emctl stop iasconsole
```

```
emctl start iasconsole
```

### 7.3.3.8 Testing the Configuration on APPHOST1

1. Perform the following tests:
  - a. Access OracleAS Web Cache and Oracle HTTP Server through the Load Balancing Router with following URL:

```
https://portal.mycompany.com
```

- b. Test the connection to the Oracle Application Server Metadata Repository through the Load Balancing Router, by accessing the following URL:

```
https://portal.mycompany.com/pls/portal/http.p?cbuf=test
```

The response should be `test`. If this is the result, the Oracle Application Server middle-tier was able to connect to the OracleAS Metadata Repository. If it is not, review `APPHOST1_ORACLE_HOME/Apache/Apache/logs/error_log` and `APPHOST1_ORACLE_HOME/j2ee/OC4J_Portal/application-deployments/portal/OC4J_Portal_default_island_1/application.log` for information on how to resolve the error.

- c. Test the Oracle AS Portal using following URL (ensure that you can log in):  
**`https://portal.mycompany.com/pls/portal`**
- d. Verify that content is being cached in OracleAS Web Cache on APPHOST1, using Web Cache Administrator. Under **Monitoring**, click **Popular Requests**. Select **Cached** from the **Filtered Objects** drop-down list, and click **Update**.  
If you accessed OracleAS Portal, portal content (for example, URLs that contain `/pls/portal`) will appear.  
If there is no portal content, open another browser and log in to OracleAS Portal. Return to the **Popular Requests** page, and click **Update** to refresh the page content.
- e. Add a portlet to a page, and then verify that the new content is present. If the new content does not display properly, or if errors occur, then the OracleAS Web Cache invalidation is not configured correctly.

### 7.3.4 Installing the Second Application Server on APPHOST2

Follow these steps to install an Oracle Application Server middle tier on APPHOST2:

1. Ensure that the system, patch, kernel and other requirements are met as specified in the *Oracle Application Server Installation Guide*. You can find this guide in the Oracle Application Server platform documentation library for the platform and version you are using.
2. Copy the `staticport.ini` file from the `Disk1/stage/Response` directory to a local directory, such as `TMP`.
3. Edit the `staticport.ini` file to assign the following custom ports:

```
Oracle HTTP Server port = 7777
Oracle HTTP Server Listen port = 7778
Web Cache HTTP Listen port = 7777
Web Cache Administration port = 9400
Web Cache Invalidation port = 9401
Web Cache Statistics port = 9402
Application Server Control port = 1810
```

---

**Notes:** Ensure that these ports are not already in use by any other service on the computer. Using the Static Ports feature as described to install the Application Server Tier ensures that the port assignments will be consistent with the documentation in this section, if the ports are correctly specified in the file and the port is not already in use. Otherwise:

- If a port is incorrectly specified, then the Oracle Universal Installer will assign the default port.
- If a port is already in use, then the Oracle Universal Installer will assign the next available port.

See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-2 for more information.

---

4. Start the Oracle Universal Installer as follows:

On UNIX, issue this command: **runInstaller**

On Windows, double-click **setup.exe**

The **Welcome** screen appears.

5. Click **Next**.

On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.

6. Specify the directory you want to be the `oraInventory` directory and the operating system group that has write permission to it.

7. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.

8. Open a window and run the script, following the prompts in the window.

9. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

- The product files for installation (Source)
- The name and path to the Oracle home (Destination)

---

**Note:** Ensure that the Oracle home directory path for APPHOST2 is the same as the path to the Oracle home location of APPHOST1. For example, if the path to the Oracle home on APPHOST1 is:

`/u01/app/oracle/product/AS10gPortal`

then the path to the Oracle home on APPHOST2 must be:

`/u01/app/oracle/product/AS10gPortal`

All instructions for copying files from one computer to another assume this convention.

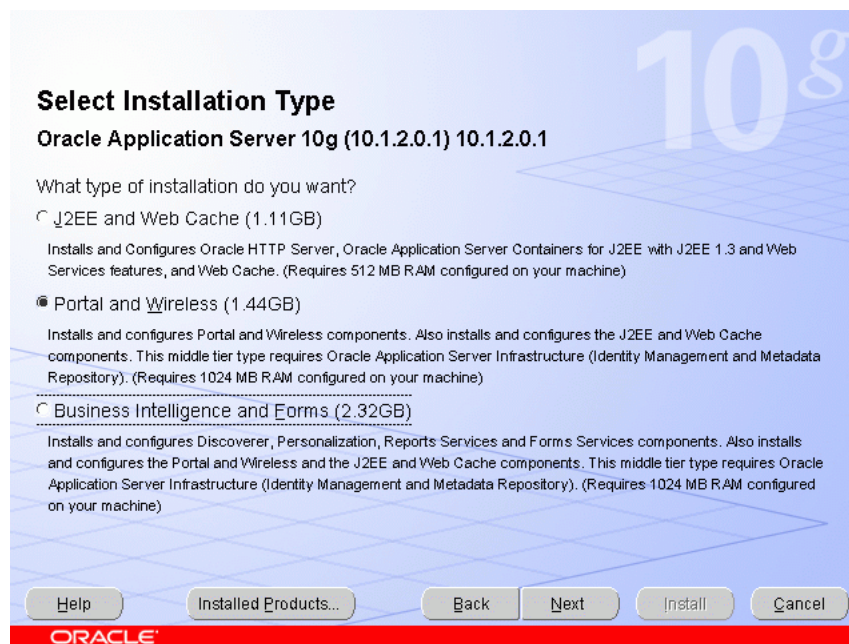
---

10. Specify the path and click **Next**.

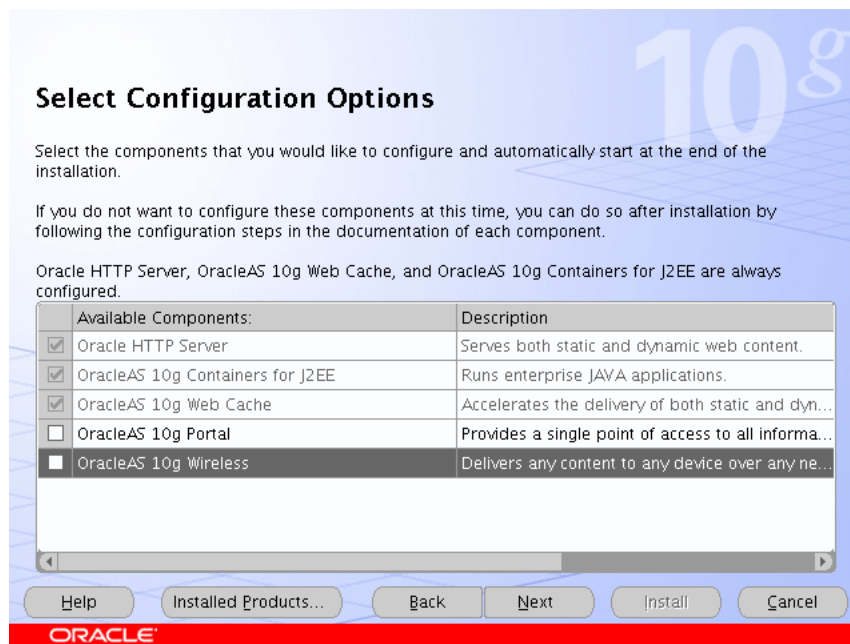
The **Select a Product to Install** screen appears.

**Figure 7-6 Oracle Universal Installer Select a Product to Install Screen**

11. Select **Oracle Application Server 10g**, as shown in [Figure 7-6](#), and click **Next**.  
The **Select Installation Type** screen appears.

**Figure 7-7 Oracle Universal Installer Select Installation Type Screen**

12. Select **Portal and Wireless**, as shown in [Figure 7-7](#), and click **Next**.  
The **Confirm Pre-Installation Requirements** screen appears.
13. Ensure that the requirements are met and click **Next**.
14. The **Select Configuration Options** screen appears.

**Figure 7–8 Oracle Universal Installer Select Configuration Options Screen**

15. Do not select any configuration options, as shown in [Figure 7–8](#), and click **Next**.

---

**Note:** Selecting the Oracle Application Server 10g Portal option in this screen now will overwrite the previously created configuration entries. For more information, refer to the *Oracle Application Server Portal Configuration Guide*, section titled "Configuring OracleAS Portal During and After Installation".

---

The **Specify Port Configuration Options** screen appears.

16. Select **Manual**, specify the location of the `staticports.ini` file, and click **Next**.
17. The **Register with Oracle Internet Directory** screen appears.

**Figure 7–9 Oracle Universal Installer Register with Oracle Internet Directory Screen**

18. Enter the host name and port of the Oracle Internet Directory load balancing router. Do not select the SSL configuration option.

19. Click **Next**.

The **Specify OID Login** screen appears.

20. Enter the user name and the password and click **Next**.

The **Select OracleAS 10g Metadata Repository** screen appears, displaying the connect string for the repository database that the installer detected.

21. Click **Next**.

The **Specify Instance Name and ias\_admin Password** screen appears.

22. Specify an instance name and the Oracle Application Server administrator's password that you specified in the first installation and click **Next**.

---

**Note:** The passwords must be the same in order to use OracleAS Web Cache clustering functionality.

---

The **Summary** screen appears.

23. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `root.sh` script.

24. Open a window and run the script, following the prompts in the window.

25. Return to the Oracle Universal Installer screen and click **Next**.

The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.

26. Click **Exit**, and then confirm your choice to exit.

## 7.3.5 Configuring the Second Application Server on APPHOST2

The configuration of the OracleAS Portal application server tier on APPHOST2 consists of the following tasks:

- [Enabling Portal on APPHOST2](#)
- [Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST2](#)
- [Configuring the Parallel Page Engine Loop-Back with the Load Balancing Router on APPHOST2](#)
- [Modifying the Portal Dependency Settings \(iasconfig.xml\) File on APPHOST2](#)
- [Configuring the Portal Tools Providers on APPHOST2](#)
- [Re-registering mod\\_osso on APPHOST2](#)

### 7.3.5.1 Enabling Portal on APPHOST2

The first task is to configure OracleAS Portal, using the Oracle Enterprise Manager 10g Application Server Control Console. Follow these steps to configure OracleAS Portal, beginning on the Application Server page:

1. Click **Configure Component**.  
The **Select Component** page appears.
2. Select **Portal** from the drop-down list.  
The **Login** page appears.
3. Enter the `ias_admin` password and click **Finish**.  
The configuration process may take 10-20 minutes to complete.

Before you continue with the OracleAS Portal application server configuration, ensure that:

- You are able to resolve `portal.mycompany.com` from APPHOST2, either with DNS or with an entry in the hosts file, such that it contacts the Load Balancing Router. To ensure you can resolve `portal.mycompany.com`:
  - Issue this command from APPHOST2:  

```
nslookup portal.mycompany.com
```

  
The IP address for the Load Balancing Router should be returned.
- You are able to contact port 7777 on `portal.mycompany.com` from APPHOST2. Issue this command on APPHOST2:  

```
telnet portal.mycompany.com 7777
```

  
Verify that no connection failure message is returned.

### 7.3.5.2 Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST2

This step associates the components on which OracleAS Portal depends with the Load Balancing Router, `portal.mycompany.com` on port 443.

1. Access the Oracle Enterprise Manager 10g Application Server Control Console.
2. Click the link for the APPHOST2 installation.
3. Click the **HTTP Server** link.



4. Click the **Administration** link.
5. Click **Advanced Server Properties**.
6. Open the `httpd.conf` file.
7. Perform the following steps:

- a. Add the `LoadModule certheaders_module` directive for the appropriate platform.

UNIX:

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

Windows:

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

- b. Add the following lines to create a `NameVirtualHost` directive and a `VirtualHost` container for **portal.mycompany.com** and port **443**.

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName portal.mycompany.com
    Port 443
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
    SimulateHttps On
</VirtualHost>
```

---

**Notes:** The `LoadModule` directives (in particular, the `LoadModule rewrite_module` directive) must appear in the `httpd.conf` file at a location preceding the `VirtualHost` directives. The server must load all modules before it can execute the directives in the `VirtualHost` container.

It is a good idea to create the `VirtualHost` directives at the end of the `httpd.conf` file.

---

- c. Create a second `NameVirtualHost` directive and a `VirtualHost` container for **apphost2.mycompany.com** and port **7777**.

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName apphost2.mycompany.com
    Port 7777
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
</VirtualHost>
```

8. Save the `httpd.conf` file, and restart the Oracle HTTP Server when prompted.
9. Copy the `APPHOST1_ORACLE_HOME/Apache/modplsql/conf/dads.conf` file to `APPHOST2_ORACLE_HOME/Apache/modplsql/conf/`.
10. Copy the `APPHOST1_ORACLE_HOME/Apache/oradav/conf/oradav.conf` file to `APPHOST2_ORACLE_HOME/Apache/oradav/conf/`.

11. Copy the `APPHOST1_ORACLE_HOME/Apache/modplsql/conf/cache.conf` file to `APPHOST2_ORACLE_HOME/Apache/modplsql/conf/cache.conf`.
12. Save the manual configuration changes to the DCM repository by issuing this command in `APPHOST2_ORACLE_HOME/dcm/bin`:  
**`dcmctl updateconfig -ct ohs`**
13. Use the Application Server Control Console to access the `mod_plsql` configuration pages.
14. Select the portal DAD and click **Edit**.
15. Click **Apply**.

The required `mod_rewrite` and `mod_oc4j` directives are added.

### 7.3.5.3 Configuring the Parallel Page Engine Loop-Back with the Load Balancing Router on APPHOST2

In this step, you enable (non-SSL) loop-back communication between the Load Balancing Router and the Parallel Page Engines on APPHOST1 and APPHOST2. If the OracleAS Web Cache on APPHOST1 is down, the Parallel Page Engine can loop back to the OracleAS Web Cache on APPHOST2 through the Load Balancing Router to reach Portal Services. This is an example of component-level high availability.

Follow these steps to create the loop-back configuration:

1. Open the `APPHOST2_ORACLE_HOME/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF/web.xml` file.
2. Locate the Page servlet section and add the lines shown in bold:

```
<servlet>
<servlet-name>page</servlet-name>
  <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
    <init-param>
      <param-name>useScheme</param-name>
      <param-value>http</param-value>
    </init-param>
    <init-param>
      <param-name>usePort</param-name>
      <param-value>7777</param-value>
    </init-param>
    <init-param>
      <param-name>httpsports</param-name>
      <param-value>443</param-value>
    </init-param>
  </servlet>
```

3. Save the `web.xml` file.

The configuration now provides component-level high availability, since if the OracleAS Web Cache on APPHOST1 is down, the Parallel Page Engine can loop back to the OracleAS Web Cache on APPHOST2, through the Load Balancing Router, to reach Portal Services.

4. Save the manual configuration changes in the Distributed Configuration Management repository by issuing the following command on APPHOST2 in `ORACLE_HOME/dcm/bin`:

**`dcmctl updateconfig`**

- Restart all components on APPHOST2 by issuing the following command in *ORACLE\_HOME/opmn/bin*:

```
opmnctl stopall
```

```
opmnctl startall
```

#### 7.3.5.4 Modifying the Portal Dependency Settings (iasconfig.xml) File on APPHOST2

The Portal Dependency Settings file *iasconfig.xml* must contain the correct host, port and farm name to enable access to OracleAS Portal and perform OracleAS Web Cache invalidation.

- Copy the *APPHOST1\_ORACLE\_HOME/portal/conf/iasconfig.xml* file to *APPHOST2\_ORACLE\_HOME/portal/conf/*.
- Overwrite the file on APPHOST2 when prompted.

#### 7.3.5.5 Configuring the Portal Tools Providers on APPHOST2

You must propagate the configuration changes made to Portal Tools providers on APPHOST1 to APPHOST2 by following these steps:

- Copy the *APPHOST1\_ORACLE\_HOME/j2ee/OC4J\_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet/provider.xml* file to:  
  
*APPHOST2\_ORACLE\_HOME/j2ee/OC4J\_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet/provider.xml*
- Copy the *APPHOST1\_ORACLE\_HOME/j2ee/OC4J\_Portal/applications/portalTools/webClipping/WEB-INF/providers/webClipping/provider.xml* file to:  
  
*APPHOST2\_ORACLE\_HOME/j2ee/OC4J\_Portal/applications/portalTools/webClipping/WEB-INF/providers/webClipping/provider.xml*
- Copy the *APPHOST1\_ORACLE\_HOME/j2ee/OC4J\_Portal/config/data-sources.xml* file to:  
  
*APPHOST2\_ORACLE\_HOME/j2ee/OC4J\_Portal/config/data-sources.xml*.
- Copy the *APPHOST1\_ORACLE\_HOME/j2ee/OC4J\_Portal/config/jazn-data.xml* file to:  
  
*APPHOST2\_ORACLE\_HOME/j2ee/OC4J\_Portal/config/jazn-data.xml*
- Restart the OC4J\_Portal instance.

#### 7.3.5.6 Re-registering mod\_osso on APPHOST2

- Back up the *APPHOST2\_ORACLE\_HOME/Apache/Apache/conf/osso/osso.conf* file.
- Use FTP binary mode to copy the *APPHOST1\_ORACLE\_HOME/Apache/Apache/conf/osso/osso.conf* file to *APPHOST2\_ORACLE\_HOME/Apache/Apache/conf/osso*.

3. Synchronize the DCM repository with the values in the obfuscated `osso.conf` file by issuing the following command: **`$ORACLE_HOME/Apache/Apache/bin/ssotransfer $ORACLE_HOME/Apache/Apache/conf/osso/osso.conf`**

---

**Note:** This does not create any new partner applications; it enables the partner application **portal.mycompany.com** for APPHOST1 and APPHOST2.

---

4. Issue this command in `ORACLE_HOME/dcm/bin`:  
**`dcmctl updateconfig`**
5. Restart the components on APPHOST2 by issuing these commands in `APPHOST2_ORACLE_HOME/opmn/bin`:  
**`opmnctl stopall`**  
**`opmnctl startall`**
6. Access the following URL:  
**`https://login.mycompany.com/pls/orasso`**
7. Log in to the OracleAS Single Sign-On Administration page as the Administrator, and use the **Administer Partner Applications** page to delete the entry for the partner application **apphost2.mycompany.com**.

### 7.3.6 Configuring OracleAS Web Cache Clusters

To cluster the OracleAS Web Cache instances, you will perform the configuration steps on APPHOST1 and propagate them to APPHOST2.

From the Oracle Enterprise Manager Application Server Control, you can access the Web Cache Manager, the graphical user interface provided for editing the configuration stored in the `webcache.xml` file. Start the Oracle Application Server instance on APPHOST1, then follow these steps to access the Web Cache Manager from the **System Components** page:

1. Access the Web Cache Administrator at:

**`http://apphost1.mycompany.com:9400/webcacheadmin`**

The Web Cache Administrator password dialog appears.

2. For the user name, enter `ias_admin` or `administrator`, and enter the OracleAS Web Cache administrator password.

---

**Note:** At installation time, The OracleAS Web Cache administrator password is set to the same password as the `ias_admin` password. The OracleAS Web Cache administrator password must be identical for all cache cluster members.

---

3. The **Web Cache Manager** page appears. A scrollable frame on the left side of the window contains groups of configuration elements. To access an element, click its link. The content area of the page is then populated with the values for that element.
4. Click **Clustering** in the **Properties** section.  
The **Clustering** page appears.
5. In the **Cluster Members** table, click **Add**.  
The **Add Cache to Cluster** page appears.
6. Enter the following information for APPHOST2:
  - Host Name: **`apphost2.mycompany.com`**
  - Admin. Port: **`9400`**
  - Protocol for Admin. Port: **`HTTP`**
  - Cache Name: **`apphost2.mycompany.com-Webcache`**
  - Capacity: **`20`**
7. Click **Submit**.
8. Click the **Origin Server** link in the **Origin Servers, Sites, and Load Balancing** section.  
The **Origin Server** page appears.
9. Click **Add** under the **Application Web Servers** table.  
The **Add Application Web Server** page appears.

10. Enter the following information:

- Hostname: **apphost2.mycompany.com**
- Port: **7778**
- Routing: **ENABLED**
- Capacity: **30**
- Failover Threshold: **5**
- Ping URL: **/**
- Ping Interval: **10**
- Protocol: **HTTP**

11. Click **Submit**.

12. Click the **Site-to-Server Mapping** link in the **Origin Servers, Sites, and Load Balancing** section.

The **Site-to-Server Mapping** page appears.

13. Select the mapping for the Load Balancing Router site (portal.mycompany.com) from the table and click **Edit Selected**.

The **Edit/Add Site-to-Server Mapping** page appears.

14. In the **Select Application Web Servers** section, select an application Web server specified in the Origin Servers page for **apphost2.mycompany.com** (**apphost1.mycompany.com** is already mapped).

15. Click **Submit**.

16. Click **Apply Changes**.

17. In the **Cache Operations** page, click **Propagate**.

The changes are propagated to apphost2.mycompany.com.

18. Click **Restart**.

OracleAS Web Cache is restarted on APPHOST1 and APPHOST2. OracleAS Web Cache on APPHOST1 begins to balance requests to the Oracle HTTP Server and OC4J\_Portal instances on APPHOST2.

After the clustering operation is completed, OracleAS Web Cache on APPHOST1 will start balancing requests to the Oracle HTTP Server and OC4J\_Portal instances running on APPHOST2. Repeat the steps in [Section 7.3.3.8, "Testing the Configuration on APPHOST1"](#) on page 7-39 to confirm that the Oracle HTTP Server and OC4J\_Portal instances on APPHOST2 were configured properly.

**Tip:** If these tests yield unsatisfactory or unexpected results, revisit the configuration steps performed to identify the cause. If the site is accepting live traffic, you might find it useful to temporarily remove the new OracleAS Web Cache instance from the cluster, revisiting the configuration while the new middle tier is completely off-line. After the problem is resolved, you can redo the clustering operation and perform the validation again.

### Enabling Monitoring of the Load Balancing Router's OracleAS Portal Host and Port Settings

You must first configure a certificate in Oracle Enterprise Manager 10g on APPHOST2 in order to successfully monitor the OracleAS Portal metrics using the Oracle Enterprise Manager 10g Application Server Control Console. Perform the steps in [Section 7.3.3.7, "Enabling Monitoring of the Load Balancing Router's OracleAS Portal Host and Port Settings"](#) on page 7-36 to configure the Application Server Control Console to recognize the Certificate Authority that was used by the Web Site to support HTTPS.

## 7.3.7 Configuring Load Balancing and Monitoring

Follow the steps in [Section 7.3.2, "Configuring Load Balancing and Monitoring"](#) on page 7-23 (substituting APPHOST2) to configure the Load Balancing Router to recognize the second application server instance.

## 7.3.8 Enabling Session Binding on OracleAS Web Cache Clusters

The Session Binding feature in OracleAS Web Cache is used to bind user sessions to a given origin server to maintain state for a period of time. Although almost all components running in a default OracleAS Portal middle tier are stateless, session binding is required for two reasons:

- The Web Clipping Studio, used by both the OracleAS Web Clipping Portlet and the Web Page Data Source of OmniPortlet, uses HTTP session to maintain state, for which session binding must be enabled.
- Enabling session binding forces all the user requests to go to a specific OracleAS Portal middle-tier, resulting in a better cache hit ratio for the portal cache.

Follow these steps on APPHOST1 or APPHOST2 to enable session binding in OracleAS Web Cache:

1. Access the Web Cache Administrator at:

**`http://apphost1.mycompany.com:9400`**

The Web Cache Administrator password dialog appears.

2. Enter the OracleAS Web Cache administrator password.

---

**Note:** At installation time, The OracleAS Web Cache administrator password is set to the same password as the `ias_admin` password. The OracleAS Web Cache administrator password must be identical for all cache cluster members.

---

3. The **Web Cache Manager** page appears. A scrollable frame on the left side of the window contains groups of configuration elements. To access an element, click its link. The content area of the page is then populated with the values for that element.
4. Click the **Session Binding** link in the **Origin Servers, Sites, and Load Balancing** section.

The **Session Binding** page appears.

5. Select the Load Balancing Router site, portal.mycompany.com:443, from the table and click **Edit Selected**.

The **Edit Session Binding** window opens.

6. Select **Any Set-Cookie** from the **Please select a session** drop-down list.
7. Select **Cookie-based** from the **Please select a session binding mechanism** drop-down list.
8. Click **Submit**.
9. Click **Apply Changes**.
10. On the **Cache Options** page, click **Propagate**.

The changes are propagated to the OracleAS Web Cache instance on the other computer.

11. Click **Restart**.

OracleAS Web Cache is restarted on APPHOST1 and APPHOST2.

### 7.3.9 Modifying the Oracle Application Server Welcome Page

The default **Welcome** page for the Oracle Application Server provides a link to the Oracle Application Server Farm page. When you access the secure server's **Welcome** page, access is provided to the **Farm** page (through the link in the **Oracle Application Server Logins** section), which displays the internal server name and port in the URL when the Farm page is accessed. This behavior is contraindicated by the security policy to hide internal server names, and should be modified.

To ensure that internal server names are not exposed by the Oracle Application Server **Welcome** page on the external server, you can do one of the following:

- Substitute a custom index.html page for the external server
- Modify the external server's standard index.html page to eliminate the following content:

#### **Oracle Application Server Logins**

To manage and monitor Oracle Application Server, log on to Oracle Enterprise Manager 10g Application Server Control:

username: ias\_admin

password: specified during install

### 7.3.10 Registering Web Providers or Provider Groups Exposed over SSL (Optional)

To register a Web provider that is exposed over SSL, you must have a copy of the root certificate of the certificate authority used by the Web provider. If the Web provider is using an unknown or uncommon certificate authority, you must add the appropriate root certificate (using Base-64 encoded X.509 format) to the set of trusted certificates recognized by the Oracle database hosting the OracleAS Metadata Repository containing the OracleAS Portal schema.

If the Portal schema is located in an OracleAS Metadata Repository Creation Assistant database, and if the release of that Oracle Database is earlier than 10g (10.1.0.x), then you do not need to perform these steps.



To register Web providers or provider groups, perform these steps:

1. Navigate to the `ORACLE_HOME/javavm/lib/security` directory in the Oracle home containing the Oracle database that hosts the OracleAS Metadata Repository containing the OracleAS Portal schema.
2. Create a backup of the truststore file `cacerts`, for example, `cacerts.bak`.
3. Issue this command to add the required certificate to the trust store:

```
ORACLE_HOME/jdk/bin/keytool -import -alias alias name -file  
root certificate file name -trustcacerts -v -keystore  
$ORACLE_HOME/javavm/lib/security/cacerts
```

4. Provide the trust store password, and type `yes` when prompted for confirmation.

### 7.3.11 Enabling the Federated Portal Adapter for SSL (Optional)

The Federated Portal Adapter uses the Oracle HTTP Server rewrite rules to simplify URLs for registering providers. By default, these rewrite rules are only specified for HTTP communication.

Follow these steps to enable the Federated Portal Adapter for SSL:

1. Edit the Virtual Hosts section of the `ORACLE_HOME/Apache/Apache/conf/ssl.conf` file as follows:

```
## SSL Virtual Host Context
##
#
# NOTE: this value should match the SSL Listen directive set previously in this
# file otherwise your virtual host will not respond to SSL requests.
#
<VirtualHost _default_:3011>
    # General setup for the virtual host
    DocumentRoot /u01/app/oracle/product/as10g/Apache/Apache/htdocs
    ServerName apphost1.mycompany.com
    ServerAdmin you@your.address
    ErrorLog /u01/app/oracle/product/as10g/Apache/Apache/logs/error_log
    TransferLog "/u01/app/oracle/product/as10g/Apache/Apache/logs/access_log"
    Port 3001
    SSLEngine on
    SSLCipherSuite
SSL_RSA_WITH_RC4_128_MD5:SSL_RSA_WITH_RC4_128_SHA:SSL_RSA_WITH_3DES_EDE_CBC_
SHA:SSL_RSA_WITH_DES_CBC_SHA:SSL_RSA_EXPORT_WITH_RC4_40_MD5:S
SL_RSA_EXPORT_WITH_DES40_CBC_SHA
    SSLWallet
file:/u01/app/oracle/product/as10g/Apache/Apache/conf/ssl.wlt/default
    <Files ~ ".(cgi|shtml)$">
        SSLOptions +StdEnvVars
    </Files>
    <Directory /u01/app/oracle/product/as10g/Apache/Apache/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>
        SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
        CustomLog /u01/app/oracle/product/as10g/Apache/Apache/logs/ssl_request_
log "%t %h %{SSL_PROTOCOL}x
%{SSL_CIPHER}x \"%r\" %b"
        RewriteEngine on
        RewriteOptions inherit
</VirtualHost>
```

2. Issue this command in `ORACLE_HOME/dcm/bin` to update the Distributed Configuration Management repository with the changes:

```
dcmctl updateconfig
```

3. Restart the Oracle Application Server instance by issuing these commands in `ORACLE_HOME/opmn/bin`:

```
opmnctl stopall
```

```
opmnctl startall
```

## 7.3.12 Registering OracleAS Portal as an Oracle Ultra Search Content Source (Optional)

If OracleAS Portal was configured using Oracle Enterprise Manager, the Oracle Ultra Search instance is not configured automatically. Therefore, the Ultra Search Administration link in OracleAS Portal will not work. To set this up, you must create an Oracle Ultra Search instance. For instructions, see the *Oracle Ultra Search Administrator's Guide*.

After you create an Oracle Ultra Search instance, perform the steps in this section to enable Oracle Ultra Search access to secure Web sites, and register OracleAS Portal as a content source.

### 7.3.12.1 Enabling Oracle Ultra Search Access

For Oracle Ultra Search to access secure Web sites, you must import certificates into the crawler's trust store and the OC4J JVM's trust store.

By default, the OC4J JVM recognizes certificates of well-known certificate authorities. However, if the secure portal instance uses a self-signed certificate or a certificate signed by an unknown certificate authority, then that certificate must be imported into the OC4J JVM's trust store. The OC4J JVM default trust store is located at `ORACLE_HOME/jdk/jre/lib/security/cacerts`.

To add the required certificate to the trust store, perform the following steps:

1. Navigate to `ORACLE_HOME/jdk/jre/lib/security`.
2. Create a backup of the trust store file `cacerts` (for example, `cacerts.bak`).
3. Issue this command to add the required certificate to the trust store:

```
$ORACLE_HOME/jdk/bin/keytool -import -alias aliasName -file  
root_certificate_file_name -trustcacerts -v -keystore  
$ORACLE_HOME/jdk/jre/lib/security/cacerts
```
4. Provide the trust store password, and type **Yes** when prompted for confirmation.
5. Repeat Steps 1 through 4 on the Oracle Application Server Infrastructure that contains the Oracle Ultra Search crawler.

### 7.3.12.2 Registering OracleAS Portal as an Oracle Ultra Search Content Source

To register OracleAS Portal as an Oracle Ultra Search content source:

1. Access the Ultra Search administration tool by clicking Ultra Search Administration in the Services portlet.

---

**Note:** By default, the **Services** portlet is on the **Portal** sub-tab of the **Administer** tab on the **Portal Builder** page.

---

2. Log in.
3. On the **Instances** tab, select the instance to manage.
4. Click **Apply** to set the instance.
5. On the **Crawler** tab, enter the **Cache Directory Location** and the **Crawler Log File Directory**.

---

**Note:** These directories are on the Oracle Application Server middle tier computer. For example, you could enter `/tmp` for the **Cache Directory Location** and `/tmp` for the **Crawler Log File Directory**.

---

6. On the **Sources** tab, click the **Oracle Sources** sub-tab, choose **Oracle Portal (Crawlable)** from the **Create Source** drop-down list and click **Go**.
7. (Optional) Edit the OracleAS Portal data source and customize the types of documents the Oracle Ultra Search crawler should process. HTML and plain text are the default document types that the crawler will always process, but you can add other document types such as MS Word Doc, MS Excel Doc, PDF, and so on.
8. Enter OracleAS Portal registration details:
9. Enter the **Portal Name**.
10. For the URL base, enter the base URL for the portal. Use the format:  
**`http://host:port/pls/portal DAD/portal schema`**  
 For example:  
**`http://apphost1.mycompany.com:7777/pls/portal/portal`**
11. Click **Register Portal**.
12. Select the page groups that you would like to create data sources for and then click **Create Portal Data Sources**. (Optional: Edit each of the portal data sources to add content types for processing. For example, you can add the MS Word Doc, MS Excel Doc, or PDF Doc types.)

---

**Note:** A page group is available as a crawlable data source when:

- The option Display Page to Public Users is set on its root page (Edit Page:Access tab).
- The View privilege is granted to PUBLIC (Edit Page Group: Access tab).

See the *Oracle Application Server Portal User's Guide* for more information.

---

13. On the **Schedules** tab, schedule the indexing of the portal data sources:
  - a. Click **Create New Schedule** and enter a name for the schedule.
  - b. Click **Proceed to Step 2** and specify synchronization schedule details.
  - c. Click **Proceed to Step 3**, select **Portal** from the drop down list and then click **Get Sources**.

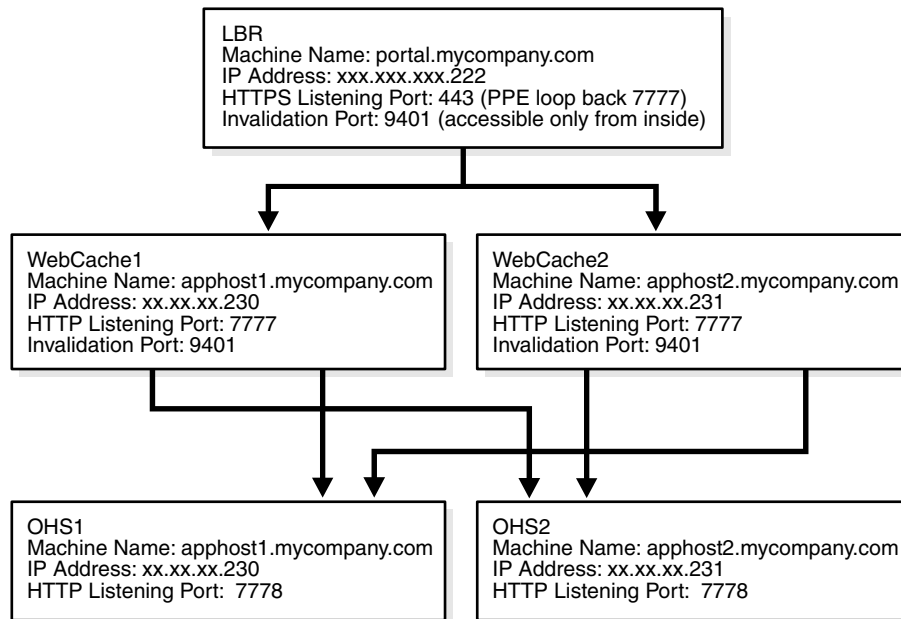
- d. Move the sources over to the Assigned Sources box and click Finish.  
(Optional: Click the **Status** link for the source to run the synchronization immediately.)

After you have registered OracleAS Portal as an Oracle Ultra Search content source, you can register the Oracle Ultra Search provider with OracleAS Portal.

## 7.4 Testing the Application Server Tier

The complete configuration is shown in [Figure 7-10](#).

**Figure 7-10 Final Application Server Configuration: APPHOST1 and APPHOST2**



To ensure that it is working as it should, perform the following tests:

1. Ensure that all components on APPHOST2 are running.
  - a. Issue this command `ORACLE_HOME/opmn/bin` to query the components' status:
 

```
opmnctl status
```
  - b. If necessary, issue this command in `ORACLE_HOME/opmn/bin`:
 

```
opmnctl startall
```
2. Stop all components on APPHOST1 by issuing this command in `ORACLE_HOME/opmn/bin`:
 

```
opmnctl stopall
```
3. Access OracleAS Web Cache and Oracle HTTP Server through the Load Balancing Router with following URL:
 

```
https://portal.mycompany.com
```

4. Test the connection to Oracle Application Server Metadata Repository through the Load Balancing Router, by accessing the following URL:

**`https://portal.mycompany.com/pls/portal/htp.p?cbuf=test`**

The response should be `test`. If this is the result, the Oracle Application Server middle-tier was able to connect to the OracleAS Metadata Repository. If it is not, review these files for information on how to resolve the error:

- `APPHOST2_ORACLE_HOME/Apache/Apache/logs/error_log`
- `APPHOST2_ORACLE_HOME/j2ee/OC4J_Portal/application-deployments/portal/OC4J_Portal_default_island_1/application.log`

5. Test the Oracle AS Portal using following URL (ensure that you can log in):

**`https://portal.mycompany.com/pls/portal`**

6. Verify that content is being cached in OracleAS Web Cache on APPHOST2, using Web Cache Administrator. Under **Monitoring**, click **Popular Requests**. Select **Cached** from the **Filtered Objects** drop-down list, and click **Update**.

If you accessed OracleAS Portal, portal content (for example, URLs that contain `/pls/portal`) will appear.

If there is no portal content, open another browser and log in to OracleAS Portal. Return to the **Popular Requests** page, and click **Update** to refresh the page content.

7. Add a portlet to a page, and then verify that the new content is present. If the new content does not display properly, or if errors occur, then the OracleAS Web Cache invalidation is not configured correctly.
8. Repeat steps 3 through 7, first ensuring that all components on APPHOST1 are running, and all components on APPHOST2 are stopped. (Refer to steps 1 and 2 for the commands to do this.)
9. Repeat steps 3 through 7, first ensuring that all components on APPHOST1 and APPHOST2 are running. (Refer to steps 1 and 2 for the commands to do this.)

## 7.5 Configuring Custom Java Portal Development Kit (JPDK) Providers

There are two types of JPDK providers: custom JPDK providers, which are created by users, and seeded JPDK providers, such as the OracleAS Portal Tools (Web Clipping and OmniPortlet) providers, which are created by the OracleAS Portal installation. This section recommends a deployment scheme, and explains how to configure the custom JPDK providers.

---

**Note:** In multiple middle tier environments that use a Load Balancing Router, all JPDK applications must be re-registered with the Load Balancing Router URL. This URL or port need not be accessible from outside of the firewall; port 7777, which is configured for the Parallel Page Engine loop back, can also be used for the JPDK registration port. You could also designate a separate URL for the JPDK applications on a separate Virtual IP address of the Load Balancing Router.

---

If you are using custom J2EE applications with session APIs, and you need to replicate state between the JPDK instances on multiple middle tiers, you must deploy JPDK and custom J2EE applications on separate OC4J instances. The applications can then use OC4J session state replication, with OC4J islands, to automatically replicate the session state across multiple processes in an application server instance, and in a cluster, across multiple application instances operating on different computers.

### 7.5.1 Deploying Custom JPDK Providers

Follow these steps to deploy custom JPDK providers:

1. Use the Oracle Enterprise Manager 10g Application Server Control Console to create a new OC4J instance named OC4J\_JPDK on APPHOST1 and APPHOST2.
2. Use the Application Server Control Console to deploy the custom providers in the OC4J\_JPDK instances on APPHOST1 and APPHOST2.
3. Use the Application Server Control Console to start the OC4J\_JPDK on APPHOST1 and APPHOST2.
4. Configure your provider registration URL to go through the Load Balancing Router, and verify that the provider works properly through the Load Balancing Router, by accessing the test page at the following URL:

**`http://portal.mycompany.com:7777/<webApp>/providers/<provider name>`**

### 7.5.2 Configuring Manually Managed Oracle Application Server Clusters for Session State Replication in OC4J\_JPDK Applications

A Manually Managed OracleAS Cluster provides the following load balancing and high availability services to a group of Oracle Application Server instances:

- Replication of session state across instances in the cluster
- Load balancing of requests among instances in the cluster
- Transparent failover of requests to a surviving instance in the cluster

A Manually Managed OracleAS Cluster does not provide configuration management services to the cluster (automatic synchronization of instance configurations within the cluster). You must make configuration changes on each instance in the cluster.

---

---

**Note:** See the *Oracle Application Server High Availability Guide* for a complete discussion of types of clusters.

---

---

The following tasks are required to configure a Manually Managed OracleAS Cluster of the Oracle Application Server instances hosting the OC4J\_JPDK instances. Follow the steps in each of the sections listed to configure the Manually Managed OracleAS Cluster:

- [Configuring State Replication in the OC4J Instances](#) on page 7-78
- [Configure the J2EE Applications for Clustering](#) on page 7-80
- [Configure the Oracle HTTP Server for Failover and Load Balancing](#) on page 7-81

### 7.5.2.1 Configuring State Replication in the OC4J Instances

To operate stateful applications, you must replicate the state among OC4J instances in the Manually Managed OracleAS Cluster. This section explains how to configure state replication for Web applications and EJB applications in the OC4J instances.

**7.5.2.1.1 Configuring State Replication for Web Applications** Perform these steps on APPHOST1 and APPHOST2 to configure state replication:

1. Start Oracle Enterprise Manager 10g Application Server Control Console.
2. Click the link for the instance you want to configure.  
The **Application Server** page appears.
3. Click **OC4J\_JPDK** in the System Components table.  
The **OC4J:OC4J\_JPDK** page appears.
4. Click **Administration**.
5. Click **Replication Properties** in the **Instance Properties** column.
6. Select the **Replicate session state** checkbox in the **Web Applications** section.
7. Leave the **Multicast Host** and **Multicast Port** fields blank.

---

**Note:** You may provide a multicast host IP address and port number (if you do not, the default is IP address 230.0.0.1 and port 23791). The IP address must be within the range 224.0.0.2 through 239.255.255.255.

Do not use the same multicast address for HTTP and EJB.

---

8. Click **Apply**.

A confirmation page appears with the message "Replication properties have been applied."

9. Click **OK**.

### 7.5.2.2 Configure the J2EE Applications for Clustering

You must configure J2EE applications to operate in a cluster. To configure the applications, you must edit the `web.xml` and `orion-web.xml` files for each application on APPHOST1 and APPHOST2. Follow the steps in this section to configure the applications for clustering.

1. Start Oracle Enterprise Manager 10g Application Server Control Console.
2. Click the link for the instance you want to configure.  
The **Application Server** page appears.
3. Click **OC4J\_JPDK** in the System Components table.  
The **OC4J:OC4J\_JPDK** page appears.
4. Click **Applications**.
5. Click the name of the web application to configure in the **Deployed Applications** section.  
The **Application** page for the application appears.
6. Edit the `orion-web.xml` file as follows:

- a. Click the module name in the **Web Modules** section.

The Web Module page for the module appears.

- b. Click **Advanced Properties**.

The **Edit orion-web.xml** page appears.

- c. Edit the file to add the `cluster-config` tag within the `orion-web-app` tag. An example of an added `cluster-config` tag is shown in the following lines.

```
<orion-web-app ... >
...
  <cluster-config/>
...
</orion-web-app>
```

- d. Click **Apply**.

A confirmation page appears with the message "Changes have been applied to orion-web.xml."

- e. Click **No** to answer the prompt to restart the server. You will restart the instance after completing all of the required edits.

The **Edit orion-web.xml** page appears.

7. Restart the OC4J\_JPDK instances by issuing this command on APPHOST1 and APPHOST2:

```
opmnctl restartproc ias-component=OC4J
```

### 7.5.2.3 Configure the Oracle HTTP Server for Failover and Load Balancing

The `mod_oc4j` module of the Oracle HTTP Server routes HTTP requests to OC4J instances. `mod_oc4j` identifies requests by their URL prefix, or root context, and routes them to the application associated with that root context. By communicating with OPMN, `mod_oc4j` can determine the status of an OC4J instance (running or stopped), and route requests only to running instances.

Using the `Oc4jMount` directive in the `ORACLE_HOME/Apache/Apache/conf/mod_oc4j.conf` file, you can specify request routing destinations for OC4J applications, and designate APPHOST1 and APPHOST2 as failover candidates for one another.

Follow these steps on APPHOST1 and APPHOST2:

1. On the **Oracle Enterprise Manager 10g** page, select the instance from the **Standalone Instances** section.

The **Application Server** page for the instance appears.

2. Click the `HTTP_Server` link.

The **HTTP Server** page appears.

3. Click **Administration**.

A list of links appears.

4. Click **Advanced Server Properties**.

A list of configuration files appears.

5. Click the `mod_oc4j.conf` link.

The **Edit mod\_oc4j.conf** screen appears.



6. Add an `Oc4JMount` directive to specify the instance to which requests should be load balanced. For example:

```
Oc4jMount path instance://APPHOST1instance:OC4J_Portal,APPHOST2instance:OC4J_Portal
```

In the preceding example, *path* specifies the URI pattern of the request (such as the context root or application directory, that is, `/myapp/*`). `APPHOST1` and `APPHOST2` specify the instance names of the Oracle Application Server instances.

**Tip:** To determine the Oracle Application Server instance names, issue this command in `APPHOST1_ORACLE_HOME/dcm/bin` and `APPHOST2_ORACLE_HOME/dcm/bin`:

```
dcmctl whichinstance
```

7. Click **Apply**.

A confirmation page appears with the message "Configuration changes have been saved. The HTTP Server must be restarted before the changes take effect. Would you like to restart now?"

8. Click **Yes**.

A confirmation page appears with the message "HTTP\_Server has been restarted."

9. Click **OK**.

The **Edit mod\_oc4j.conf** screen appears.

10. Click the **Application Server** link.

The **Application Server** page appears.

#### 7.5.2.4 Disabling the JAZN Session Cache for UDDI Session Replication

Follow these steps to disable the JAZN session cache:

1. Open the `ORACLE_HOME/j2ee/OC4J_Portal/config/jazn.xml` file and locate the `jazn provider` element.
2. Add the property `ldap.cache.session.enable`, set to `false`, as shown in the following example:

```
<jazn provider="LDAP">
    ....
    <property name="ldap.cache.session.enable" value="false" />
</jazn>
```

## 7.6 Setting the OracleAS Single Sign-On Query Path URL for External Applications

This section explains how to set the URL for the OracleAS Single Sign-On query path. You need only perform this task if you are using external applications.

OracleAS Portal maintains the URL prefix of OracleAS Single Sign-On, which accesses certain information through HTTP requests from the database using the `UTL_HTTP` package. These requests must be made over the HTTP protocol (rather than HTTPS). Consequently, even if OracleAS Portal and OracleAS Single Sign-On are configured to use HTTPS, OracleAS Single Sign-On must still have access to an HTTP port, so that it can support these interfaces. The purpose of the requests is to:

- Obtain the list of external applications to allow customization of the External Applications portlet.
- Map OracleAS Single Sign-On user names to external application user names.

Perform these steps to set the URL:

1. Configure the Load Balancing Router (`login.mycompany.com`) with an internal network address translated port `7777`, to receive requests from the OracleAS Portal database and pass them to both OracleAS Single Sign-On Oracle HTTP Servers.
2. Log on to OracleAS Portal as the portal administrator.
3. Click the **Administer** tab.
4. Click the **Portal** tab.
5. Click **Global Settings** in the **Services** portlet.
6. Click the **SSO/OID** tab.
7. Edit the **Query Path URL Prefix** under **SSO Server Settings**. Enter a URL for OracleAS Single Sign-On, for example:

`http://login.mycompany.com:7777/pls/orasso`

8. Allow OracleAS Portal to access the single sign-on server using the HTTP protocol.
  - a. In the `ORACLE_HOME/sso/conf/sso_apache.conf` file, uncomment and modify (as shown in bold) this directive:

```
<Location "/pls/orasso/*[Aa][Pp][Pp][Ss]_[Ll][Ii][Ss][Tt]">
  Order deny,allow
  Deny from all
  Allow from fully qualified OracleAS Portal host name
</Location>
```

- b. Save and close the file.
- c. Issue this command in `ORACLE_HOME/dcm/bin`:

```
dcmctl updateconfig
```

9. Configure the rule for the Load Balancing Router. The example, for the Big IP Load Balancing Router, is presented for illustration only. In practice, you should ensure that any access rule you apply is consistent with the load balancing router in use.

```
if (client_addr != <infrastructure db IP> netmask 255.255.255.0 and
    (http_uri starts_with
      "/pls/orasso/orasso.wwsso_app_admin.external_apps_list" or
      http_uri starts_with
```

```
"/pls/orasso/orasso.wssso_app_admin.validate_user")) {  
discard  
}  
else {  
use pool SSO  
}
```

---

**Note:** In a deployment configuration where the single sign-on server and OracleAS Portal are front-ended by a Load Balancing Router, the rule for limiting access to hosts should be set directly with the Load Balancing Router. Do not attempt to add such a rule in the `ORACLE_HOME/sso/conf/sso_apache.conf` file to allow or deny access to a host for this configuration.

---

### 7.6.1 Firewall Considerations for OracleAS Portal

Connection availability for OracleAS Portal is governed by the pool of connections to the OracleAS Portal database. The default idle time out for the pooled database connections is 15 minutes, and is configurable with the `PlsqlIdleSessionCleanupInterval` parameter. See the *Oracle HTTP Server Administrator's Guide* for information on setting this parameter.

In order to prevent intermittent connection problems when accessing OracleAS Portal, the time out value for this parameter must always be lower than the firewall time out settings between the Oracle Application Server middle tier and the OracleAS Portal database, and the Load Balancing Router time out settings. If it is not, OracleAS Portal will try to use a pooled connection that has already been timed out by the firewall, and errors will occur.



---

# Installing and Configuring the myBIFCompany Application Infrastructure

This chapter provides instructions for creating the Application and Web Server tiers of the myBIFCompany architecture, distributing the software components into the DMZs shown in the Enterprise Deployment architecture depicted in [Figure 2–3](#) on page 2-9.

Before you perform the tasks in this chapter, a two-node Real Application Clusters (RAC) database must be installed. In this chapter, the server names for the database hosts are APPDBHOST1 and APPDBHOST2. Ideally, these are separate physical databases from INFRADBHOST1 and INFRADBHOST2. In addition to isolating the security components, separate application databases provide the flexibility needed to maintain and tune application and security parameters separately.

This chapter contains the following topics:

[Section 8.1, "Installing the Metadata Repository for the Application Infrastructure"](#) on page 8-1

[Section 8.2, "Configuring the Load Balancing Router or Proxy Server"](#) on page 8-1

[Section 8.3, "Installing the Application Tier"](#) on page 8-2

## 8.1 Installing the Metadata Repository for the Application Infrastructure

The procedure for installing the Metadata Repository for the myBIFCompany Application Infrastructure is identical to that for installing it in the myPortal Application Infrastructure. Therefore, follow the instructions in [Section 7.1, "Installing the Metadata Repository for the Application Infrastructure"](#) on page 7-1.

## 8.2 Configuring the Load Balancing Router or Proxy Server

Follow the instructions provided in [Section 7.2](#) on page 7-6.

## 8.3 Installing the Application Tier

Follow these steps to install the Application Tier components (APPHOST1 and APPHOST2) into the Application tier.

### 8.3.1 Installing the First Application Server on APPHOST1

Follow these steps to install an Oracle Application Server middle tier on APPHOST1:

1. Ensure that the system, patch, kernel and other requirements are met as specified in the *Oracle Application Server Installation Guide*. You can find this guide in the Oracle Application Server platform documentation library for the platform and version you are using.
2. Copy the `staticport.ini` file from the `Disk1/stage/Response` directory to a local directory, such as `TMP`.
3. Edit the `staticport.ini` file to assign the following custom ports:

```
Oracle HTTP Server port = 7777
Oracle HTTP Server Listen port = 7778
Web Cache HTTP Listen port = 7777
Web Cache Administration port = 9400
Web Cache Invalidation port = 9401
Web Cache Statistics port = 9402
Application Server Control port = 1810
```

---

**Notes:** Ensure that these ports are not already in use by any other service on the computer. Using the Static Ports feature as described to install the Application Server Tier ensures that the port assignments will be consistent with the documentation in this section, if the ports are correctly specified in the file and the port is not already in use. Otherwise:

- If a port is incorrectly specified, then the Oracle Universal Installer will assign the default port.
- If a port is already in use, then the Oracle Universal Installer will assign the next available port.

See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-2 for more information.

Port 80 is open on the firewall only to accept and redirect requests using the HTTP (non-secure) protocol. Requests using the HTTP protocol (in the form `http://www.mycompany.com`) are redirected to port 443. Requests using the HTTPS, or secure, protocol (in the form `https://www.mycompany.com`) are managed by port 443.

---

4. Start the Oracle Universal Installer as follows:

On UNIX, issue this command: **runInstaller**

On Windows, double-click **setup.exe**

The **Welcome** screen appears.

5. Click **Next**.

On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.

6. Specify the directory you want to be the oraInventory directory and the operating system group that has write permission to it.

7. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.

8. Open a window and run the script, following the prompts in the window.

9. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

- The product files for installation (Source)
- The name and path to the Oracle home (Destination)

10. Specify the path and click **Next**.

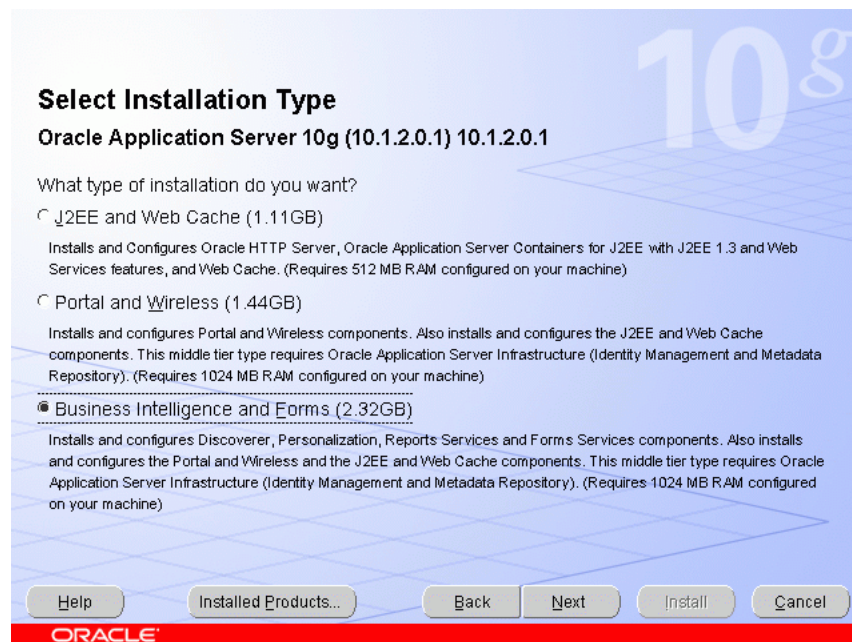
The **Select a Product to Install** screen appears.

**Figure 8–1 Oracle Universal Installer Select a Product to Install Screen**



11. Select **Oracle Application Server 10g**, as shown in [Figure 8–1](#), and click **Next**.

The **Select Installation Type** screen appears.

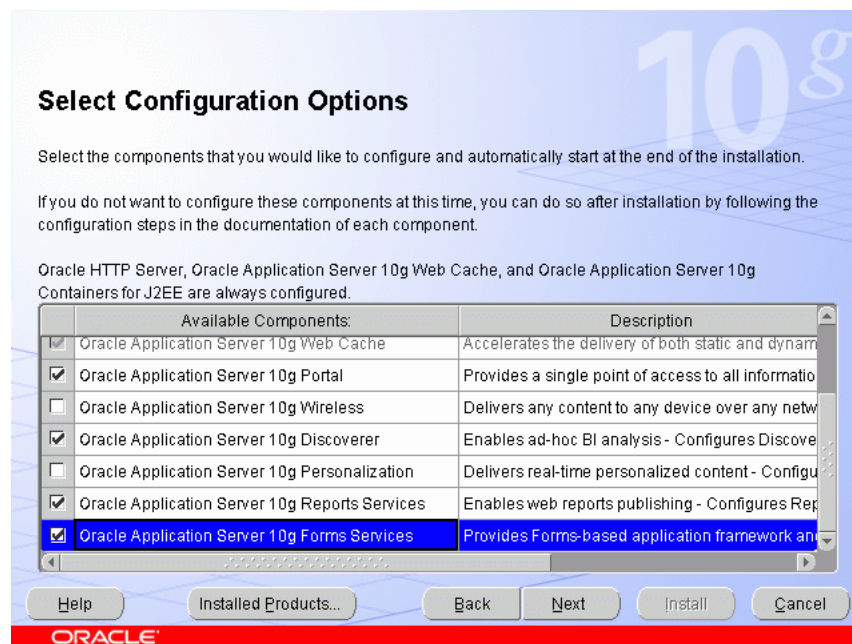
**Figure 8–2 Oracle Universal Installer Select Installation Type Screen**

12. Select **Business Intelligence and Forms**, as shown in Figure 8–2, and click **Next**.

The **Confirm Pre-Installation Requirements** screen appears.

13. Ensure that the requirements are met and click **Next**.

14. The **Select Configuration Options** screen appears.

**Figure 8–3 Oracle Universal Installer Select Configuration Options Screen**



15. Select **Oracle Application Server 10g Portal**, **Oracle Application Server 10g Discoverer**, **Oracle Application Server 10g Reports Services**, and **Oracle Application Server 10g Forms Services**, as shown in [Figure 8–3](#), and click **Next**.

The **Specify Port Configuration Options** screen appears.

16. Select **Manual**, specify the location of the `staticports.ini` file, and click **Next**.
17. The **Register with Oracle Internet Directory** screen appears.

**Figure 8–4 Oracle Universal Installer Register with Oracle Internet Directory Screen**

**Register with Oracle Internet Directory**

To register this instance of Oracle Application Server 10g with an existing Oracle Internet Directory, enter the hostname and port where Oracle Internet Directory is located.

Host:

Port:

☐ Use only SSL connections with this Oracle Internet Directory

Help Installed Products... Back Next Install Cancel

ORACLE

18. Enter the host name and port of the Oracle Internet Directory load balancing router. Do not select the SSL configuration option.

19. Click **Next**.

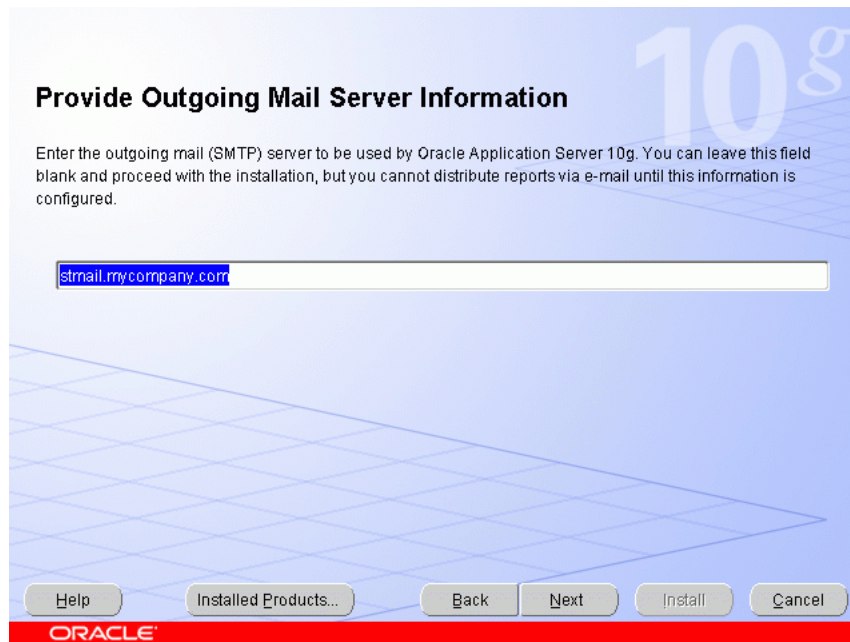
The **Specify OID Login** screen appears.

20. Enter the user name and the password and click **Next**.

The **Select OracleAS 10g Metadata Repository** screen appears, displaying the connect strings for the repository databases that the installer detected.

21. Select the application database (APPDBHOST1 and APPDBHOST2) and click **Next**.

The **Provide Outgoing Mail Server Information** screen appears.

**Figure 8–5 Oracle Universal installer Provide Outgoing Mail Server Information Screen**

22. Complete the field with the outgoing SMTP mail server name and click **Next**.

The **Specify Instance Name and ias\_admin Password** screen appears.

23. Specify an instance name and the Oracle Application Server administrator's password and click **Next**.

The **Summary** screen appears.

24. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `root.sh` script.

25. Open a window and run the script, following the prompts in the window.

26. Return to the Oracle Universal Installer screen and click **Next**.

The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.

27. Click **Exit**, and then confirm your choice to exit.
28. Confirm that the installation was successful by accessing the test pages at the following URLs:
  - For OracleAS Reports Services:  
`http://apphost1.mycompany.com:7777/reports/rwservlet`
  - For OracleAS Forms Services:  
`http://apphost1.mycompany.com:7777/forms/frmservlet`
  - For OracleBI Discoverer:  
`http://apphost1.mycompany.com:7777/discoverer/viewer`

## 8.3.2 Configuring the First Application Server on APPHOST1

The configuration of the Oracle Business Intelligence application server tier on APPHOST1 consists of the following tasks:

- [Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST1](#)
- [Re-registering mod\\_osso on APPHOST1](#)
- [Configuring OracleAS Reports Services Server Targets in Oracle Enterprise Manager 10g Application Server Control Console](#)
- [Testing the Configuration on APPHOST1](#)

### 8.3.2.1 Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST1

This step associates the components on which Oracle Business Intelligence depends with the Load Balancing Router hostname and port: bif.mycompany.com.

1. Access the Oracle Enterprise Manager 10g Application Server Control Console.
2. Click the link for the APPHOST1 installation.
3. Click the **HTTP Server** link.
4. Click the **Administration** link.
5. Click **Advanced Server Properties**.
6. Open the `httpd.conf` file.
7. Perform the following steps:
  - a. Add the `LoadModule certheaders_module` directive for the appropriate platform.

UNIX:

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

Windows:

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

- b. Add the following lines to create a `NameVirtualHost` directive and a `VirtualHost` container for bif.mycompany.com and port 443.

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName bif.mycompany.com
    Port 443
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
    SimulateHttps On
</VirtualHost>
```

---

**Notes:** The `LoadModule` directives (in particular, the `LoadModule rewrite_module` directive) must appear in the `httpd.conf` file at a location preceding the `VirtualHost` directives. The server must load all modules before it can execute the directives in the `VirtualHost` container.

It is a good idea to create the `VirtualHost` directives at the end of the `httpd.conf` file.

---

- c. Create a second `NameVirtualHost` container for `apphost1.mycompany.com` and port `7777`.

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName apphost1.mycompany.com
    Port 7777
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
</VirtualHost>
```

8. Save the `httpd.conf` file, and restart the Oracle HTTP Server when prompted.
9. Restart all of the components by issuing these commands in `ORACLE_HOME/opmn/bin`:

```
opmnctl stopall
opmnctl startall
```

### 8.3.2.2 Re-registering `mod_osso` on `APPHOST1`

1. Set the `ORACLE_HOME` environment variable to the current Oracle home.
2. Edit the SSO registration script `ORACLE_HOME/sso/bin/ssoreg` as shown in [Example 8–1](#), and then execute it. [Example 8–1](#) shows the usage of `ssoreg.sh` on UNIX; on Windows, the script name is `ssoreg.bat`.

---

**Note:** The script shown in [Example 8–1](#) has multiple lines for readability only. When you execute the script, all parameters are on a single continuous line.

---

#### **Example 8–1** *ssoreg Usage on UNIX*

```
ORACLE_HOME/sso/bin/ssoreg.sh
-site_name bif.mycompany.com
-mod_osso_url https://bif.mycompany.com
-config_mod_osso TRUE
-oracle_home_path ORACLE_HOME
-config_file ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
-admin_info cn=orcladmin
-virtualhost
```

A partner application, `bif.mycompany.com`, is created.

3. Restart the Oracle HTTP Server by issuing this command:

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

4. Access the following URL:  
**`https://login.mycompany.com/pls/orasso`**
5. Log in to the OracleAS Single Sign-On Administration page as the Administrator, and use the **Administer Partner Applications** page to delete the entry for the partner application **apphost1.mycompany.com**.

### 8.3.2.3 Configuring OracleAS Reports Services Server Targets in Oracle Enterprise Manager 10g Application Server Control Console

Follow these steps to map the reports server targets to the Load Balancing Router.

1. Open `ORACLE_HOME/sysman/emd/targets.xml` and do the following:
  - a. Locate the target with `TYPE="oracle_repserve"` and `DISPLAY_NAME="Reports Server:server_name"`
  - b. Change the value of the "Servlet" property (the servlet URL) to the URL for the Load Balancing Router (**bif.mycompany.com**).
2. Save and close the `targets.xml` file.
3. Reload the `targets.xml` file in the Application Server Control Console by issuing this command in `ORACLE_HOME/bin`:  
**`emctl reload`**
4. Restart the Application Server Control Console by issuing the following commands in `ORACLE_HOME/bin`:  
**`emctl stop iasconsole`**  
**`emctl start iasconsole`**

### 8.3.2.4 Testing the Configuration on APPHOST1

Verify that the first installation is communicating with the Load Balancing Router by accessing the test pages at these URLs:

- For OracleAS Reports Services:  
**`http://bif.mycompany.com/reports/rwservlet`**
- For OracleAS Forms Services:  
**`http://bif.mycompany.com/forms/frmservlet`**
- For OracleBI Discoverer:  
**`http://bif.mycompany.com/discoverer/viewer`**

### 8.3.3 Installing the Second Application Server on APPHOST2

The installation procedure for the second application server is identical to that for the first. Therefore, on APPHOST2, follow the steps in [Section 8.3.1, "Installing the First Application Server on APPHOST1"](#) on page 8-2.

---

**Note:** The installation instructions are not identical if you plan to configure and use OracleAS Portal in your Business Intelligence and Forms installation. If you plan to configure OracleAS Portal, do not select it as a configuration option in this, the second, installation. Doing so will overwrite previously created configuration entries. For more information, refer to the *Oracle Application Server Portal Configuration Guide*, section titled "Configuring OracleAS Portal During and After Installation".

---

### 8.3.4 Configuring the Second Application Server on APPHOST2

The configuration of the Oracle Business Intelligence application server tier on APPHOST2 consists of the following tasks:

- [Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST2](#)
- [Re-registering mod\\_osso on APPHOST2](#)
- [Configuring OracleAS Reports Services Server Targets in Oracle Enterprise Manager 10g Application Server Control Console](#)

#### 8.3.4.1 Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST2

This step associates the components on which Oracle Business Intelligence depends with the Load Balancing Router, bif.mycompany.com on port 443.

1. Access the Oracle Enterprise Manager 10g Application Server Control Console.
2. Click the link for the APPHOST2 installation.
3. Click the **HTTP Server** link.
4. Click the **Administration** link.
5. Click **Advanced Server Properties**.
6. Open the `httpd.conf` file.
7. Perform the following steps:
  - a. Add the `LoadModule certheaders_module` directive for the appropriate platform.

UNIX:

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

Windows:

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

- b. Add the following lines to create a `NameVirtualHost` directive and a `VirtualHost` container for **bif.mycompany.com** and port 443.

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName bif.mycompany.com
    Port 443
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
    SimulateHttps On
</VirtualHost>
```

---

**Notes:** The `LoadModule` directives (in particular, the `LoadModule rewrite_module` directive) must appear in the `httpd.conf` file at a location preceding the `VirtualHost` directives. The server must load all modules before it can execute the directives in the `VirtualHost` container.

It is a good idea to create the `VirtualHost` directives at the end of the `httpd.conf` file.

---

- c. Create a second `NameVirtualHost` directive and a `VirtualHost` container for **apphost2.mycompany.com** and port 7777.

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName apphost2.mycompany.com
    Port 7777
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
</VirtualHost>
```

8. Save the `httpd.conf` file, and restart the Oracle HTTP Server when prompted.
9. Copy the `APPHOST1_ORACLE_HOME/Apache/modplsql/conf/dads.conf` file to `APPHOST2_ORACLE_HOME/Apache/modplsql/conf/`.
10. Copy the `APPHOST1_ORACLE_HOME/Apache/oradav/conf/oradav.conf` file to `APPHOST2_ORACLE_HOME/Apache/oradav/conf/`.

#### 8.3.4.2 Re-registering mod\_osso on APPHOST2

Follow the steps in [Section 8.3.2.2, "Re-registering mod\\_osso on APPHOST1"](#) on page 8-8, substituting APPHOST2 where appropriate.

#### 8.3.4.3 Configuring OracleAS Reports Services Server Targets in Oracle Enterprise Manager 10g Application Server Control Console

Follow the steps in [Section 8.3.2.3, "Configuring OracleAS Reports Services Server Targets in Oracle Enterprise Manager 10g Application Server Control Console"](#) on page 8-9.

### 8.3.5 Configuring OracleAS Web Cache Clusters

To configure OracleAS Web Cache clusters, follow the steps in [Section 7.3.6, "Configuring OracleAS Web Cache Clusters"](#) on page 7-31.

### 8.3.6 Selecting the Secure Tunneling Protocol for Oracle Business Intelligence Discoverer Plus Deployment

OracleBI Discoverer Plus can communicate over secure tunneling, HTTP tunneling, or JRMP. The method used depends on the connections allowed by the firewall implementation. The myBIFCompany application infrastructure requires use of the secure tunneling protocol, since JRMP makes connections using arbitrary port numbers, and only a few designated ports are open on the firewall in the Enterprise Deployment configuration (as shown in [Figure 2-3](#)).

Follow these steps on each Oracle Business Intelligence instance to select the secure tunneling protocol:

1. In the Oracle Enterprise Manager 10g Application Server Control Console, navigate to **BIHome > Discoverer > DiscovererPlus**.
2. Click **Communication Protocols**.
3. Select **Secure Tunneling**.

The OracleBI Discoverer Plus configuration file is changed to designate the https transport.

### 8.3.7 Completing the Configuration

Follow these steps to configure the Load Balancing Router to recognize the second application server instance. The Load Balancing Router must be configured to:

- Balance requests to `bif.mycompany.com` on port 443 (an HTTPS listening port) to the Application tier OracleAS Web Cache running on APPHOST2 port 7777 (an HTTP listening port).
- Monitor OracleAS Web Cache. The Load Balancing Router must be configured to detect an inoperative computer and stop routing requests to it until it is functioning again. Two OracleAS Web Cache ports must be monitored: the HTTP request port and the invalidation port.

Use this URL in the Load Balancing Router configuration to monitor HTTP request port 7777:

*host name:port/\_oracle\_http\_server\_webcache\_static\_.html*

for example:

`http://apphost2.mycompany.com:7777/_oracle_http_server_webcache_static_.html`

To monitor invalidation port 9401, use this URL:

`http://apphost2.mycompany.com:9401/_oracle_http_server_webcache_static_.html`



### 8.3.8 Managing Connection Availability for OracleAS Reports Services

You must tune the time out settings to manage idle connections for the Load Balancing Router, firewall, Oracle HTTP Server, and OC4J according to their OracleAS Reports Services application requirements. For example, if the Load Balancing Router and firewall connection time out is set to 10 minutes, and the execution time for a report exceeds 10 minutes, then the browser connection is timed out by the Load Balancing Router, and the OracleAS Reports Services output does not reach the browser.

The time out value for the connection between OracleAS Reports Services clients and the Oracle Reports server is governed by the `idleTimeout` attribute of the connection element in the OracleAS Reports Services Server configuration file `server_name.conf`.

For more information, see *Oracle Application Server Reports Services Publishing Reports to the Web*.

### 8.3.9 Configuring Session State Replication in OC4J Instances

To ensure proper session failover, you must add `Oc4jMount` directives and state replication capability to each middle tier for the `OC4J_BI_Forms` instance. To do this, follow the instructions in:

[Section 6.7, "Configuring OC4J Routing"](#), using the configuration shown in [Example 6-3, "OC4JMount Directive to Load Balance Requests to FAQApp on Multiple Instances"](#).

[Section 7.5.2.1, "Configuring State Replication in the OC4J Instances"](#), substituting the `OC4J_BI_Forms` instance name in the instructions.

### 8.3.10 Modifying the Oracle Enterprise Manager 10g Application Server Control Console Welcome Page

You must modify the Oracle Enterprise Manager 10g Application Server Control Console to prevent display of internal server names. Follow the instructions on [Section 7.3.9, "Modifying the Oracle Application Server Welcome Page"](#) on page 7-34.

### 8.3.11 Updating Host and Port Entries in OC4J\_BI\_Forms

Upon installation, the images in OracleBI Discoverer portlet are associated with the local installation host and port. In order for the portlet to display the images when configured with the Load Balancing Router, the portlet configuration must contain the host and port values for the Load Balancing Router. Follow these steps to update the configuration:

1. Open the `ORACLE_HOME/j2ee/OC4J_BI_FORMS/config/oc4j.properties` file.
2. Modify the port and host entries as shown in bold:

```
java.rmi.server.randomIDs=true
oracle.disco.activation.preferencePort=16001
oracle.path=C\:/1012_BIF_050803_
1930/bin;C\:/WINNT/system32;C\:/WINNT;C\:/WINNT/System32/Wbem;
oracle.home=C\:\\1012_BIF_050803_1930
oracle.forms.configFileName=C\:/1012_BIF_050803_1930/forms/server/formsweb.cfg
oracle.discoverer.applications.port=443
oracle.discoverer.applications.host=bif.mycompany.com
```

3. Use the Oracle Enterprise Manager 10g Application Server Control Console to restart the `OC4J_BI_Forms` instance.

## 8.4 Testing the Application Server Tier

To ensure that the configuration is working as it should, perform the following tests:

1. Ensure that all components on APPHOST2 are running.
  - a. Issue this command `ORACLE_HOME/opmn/bin` to query the components' status:  
**`opmnctl status`**
  - b. If necessary, issue this command in `ORACLE_HOME/opmn/bin`:  
**`opmnctl startall`**
2. Stop all components on APPHOST1 by issuing this command in `ORACLE_HOME/opmn/bin`:  
**`opmnctl stopall`**
3. Access OracleAS Web Cache and Oracle HTTP Server through the Load Balancing Router with following URL:  
**`https://bif.mycompany.com`**
4. Test the connection to Oracle Application Server Metadata Repository through the Load Balancing Router, by accessing the following URLs:  
OracleAS Reports Services test page:  
**`https://bif.mycompany.com/reports/rwservlet`**  
OracleAS Forms Services test page:  
**`https://bif.mycompany.com/forms/frmservlet`**  
OracleBI Discoverer test page:  
**`https://bif.mycompany.com/discoverer/viewer`**  
The response should be `test`. If this is the result, the Oracle Application Server middle-tier was able to connect to the OracleAS Metadata Repository. If it is not, review the Oracle HTTP Server `APPHOST2_ORACLE_HOME/Apache/Apache/logs/error_log` file for information about how to resolve the error.

## 8.5 Configuring OracleAS Portal in Business Intelligence and Forms

You may also wish to configure OracleAS Portal as part of the Business Intelligence and Forms installation. In order to do so, follow these instructions:

1. Install Business Intelligence and Forms on APPHOST1 as described in [Section 8.3.1, "Installing the First Application Server on APPHOST1"](#) on page 8-2.
2. Perform the configuration steps in [Section 7.3.3, "Configuring the First Application Server on APPHOST1"](#) on page 7-13.
3. Install Business Intelligence and Forms on APPHOST2 as described in [Section 8.3.3, "Installing the Second Application Server on APPHOST2"](#) on page 8-10.
4. Perform the configuration steps in [Section 7.3.5, "Configuring the Second Application Server on APPHOST2"](#) on page 7-26.
5. Perform the configuration steps in [Section 7.3.6, "Configuring OracleAS Web Cache Clusters"](#) on page 7-31.

6. Perform the configuration steps in [Section 7.3.7, "Configuring Load Balancing and Monitoring"](#) on page 7-33.
7. Perform the configuration steps in [Section 7.3.8, "Enabling Session Binding on OracleAS Web Cache Clusters"](#) on page 7-33.
8. Perform the steps in [Section 7.3.9, "Modifying the Oracle Application Server Welcome Page"](#) on page 7-34.
9. Perform the steps in [Section 8.3.11, "Updating Host and Port Entries in OC4J\\_BI\\_Forms"](#) on page 8-13.
10. Perform the steps in [Section 8.4, "Testing the Application Server Tier"](#) on page 8-14.



---

## Implementing Architecture Variants

This chapter explains how to implement common variants to the Enterprise Deployment configurations described in this guide. This chapter contains the following topics:

[Section 9.1, "Configuring a Dedicated Intranet and Internet for OracleAS Portal"](#)

[Section 9.2, "Configuring a Reverse Proxy for OracleAS Portal and OracleAS Single Sign-On"](#)

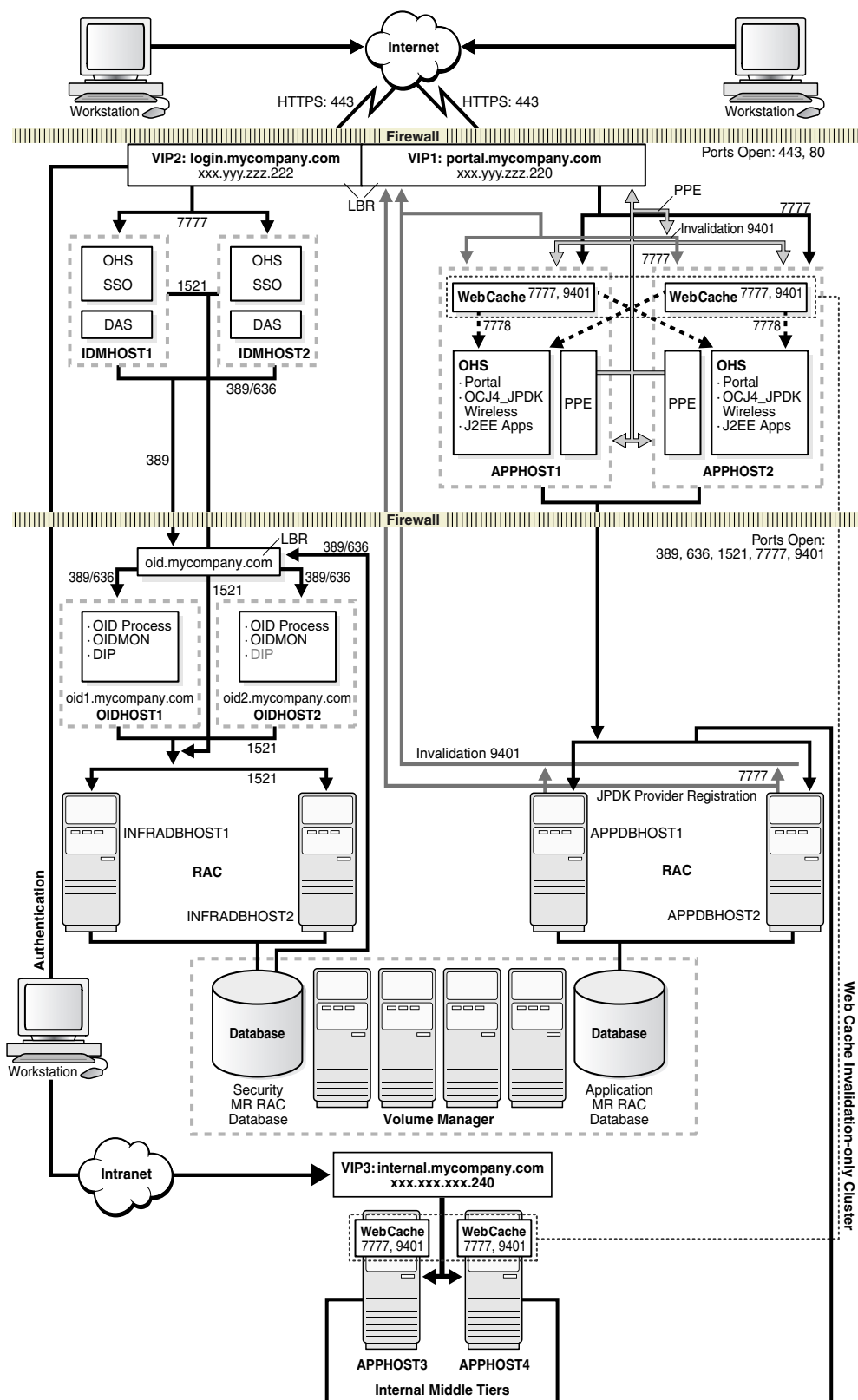
[Section 9.3, "Configuring J2EE and Web Cache on the Web Tier"](#)

### 9.1 Configuring a Dedicated Intranet and Internet for OracleAS Portal

You can configure OracleAS Portal to be accessible from within a company network as well as from external clients. This section describes some important characteristics of this configuration, and provides instructions on how to configure OracleAS Portal for this purpose.

The complete configuration for the dedicated intranet and internet is shown in [Figure 9-1](#) on page 9-2.

**Figure 9–1 Dedicated Intranet and Internet for OracleAS Portal**



The intranet/internet configuration for OracleAS Portal requires two logical middle tiers: `portal.mycompany.com` and `internal.mycompany.com`, each residing on its own computer. This separation of physical middle-tiers helps isolate the content cached for internet and intranet users. This enhances security, and also ensures that users who navigate to one logical middle tier do not access content served by the other logical middle tier. Each logical middle tier then provides access to the same OracleAS Portal schema in the Oracle Application Server Metadata Repository and the same OracleAS Portal data. In this configuration, the external logical middle tier is the primary middle tier used to install, configure, and expose web providers. The internal logical middle tier is designated as a partner application.

The intranet/internet configuration requires that all OracleAS Web Cache instances be configured as an invalidation-only cluster. Invalidation-only clustering ensures that OracleAS Web Cache maintains distinct caches for the two logical sites, while enabling the cluster members to share invalidation messages (thereby ensuring that content edits are visible across the two logical sites).

In this configuration, invalidation messages are sent from the OracleAS Portal schema in the OracleAS Metadata Repository to the internal OracleAS Web Cache instance, and the invalidation message is then sent out to all the cluster members. The invalidation message used in this configuration ensures that it invalidates content regardless of the host and port specified in the cached URL. This type of invalidation ensures that content cached with either logical middle-tier URL is invalidated. For more information on the OracleAS Web Cache invalidation-only cluster, refer to the *Oracle Application Server Portal Configuration Guide*.

To ensure that the internal and external user communities are distinct, two URLs are used to access the OracleAS Portal applications: from the intranet, `http://internal.mycompany.com`; from the Internet, `https://portal.mycompany.com`.

The process of configuring the dedicated intranet and extranet for OracleAS Portal consists of the following tasks:

1. [Installing the Infrastructure and External Middle Tier Instances \(Section 9.1.1\)](#)
2. [Installing the First Internal Middle Tier on APPHOST3 \(Section 9.1.2\)](#)
3. [Installing the Second Internal Middle Tier on APPHOST4 \(Section 9.1.3\)](#)
4. [Configuring an OracleAS Web Cache Invalidation-only Cluster \(Section 9.1.4\)](#)
5. [Configuring the First Internal Middle Tier on APPHOST3 for Load Balancing Router Access \(Section 9.1.5\)](#)
6. [Configuring the Second Internal Middle Tier on APPHOST4 for Load Balancing Router Access \(Section 9.1.6\)](#)
7. [Registering the Internal Middle Tier as a Partner Application \(Section 9.1.7\)](#)
8. [Updating the Default JPKD Instance URL and Seeded Provider Group URLs \(Section 9.1.8\)](#)
9. [Configuring OracleAS Portal Invalidation Messages \(Section 9.1.9\)](#)
10. [Configuring the OracleAS Portal Schema in the OracleAS Metadata Repository \(Section 9.1.10\)](#)
11. [Modifying the Oracle Text Base Search URL \(Section 9.1.11\)](#)
12. [Enabling Session Binding on OracleAS Web Cache \(Section 9.1.12\)](#)
13. [Configuring the Oracle Drive WebDAV Client \(Section 9.1.13\)](#)

## 14. Validating the Completed Configuration (Section 9.1.14)

### 9.1.1 Installing the Infrastructure and External Middle Tier Instances

To install and configure the necessary Infrastructure and Oracle Application Server instances:

1. Perform all steps in [Chapter 4, "Installing and Configuring the Security Infrastructure"](#).
2. Perform all steps in [Chapter 7, "Installing and Configuring the myPortalCompany Application Infrastructure"](#) (Use the URL `portal.mycompany.com` for these instances).

### 9.1.2 Installing the First Internal Middle Tier on APPHOST3

Follow these steps to install the first internal middle tier.

1. Copy the `staticport.ini` file from the `Disk1/stage/Response` directory to a local directory, such as `TMP`.
2. Edit the `staticport.ini` file to assign the following custom ports:

```
Oracle HTTP Server port = 7777
Oracle HTTP Server Listen port = 7778
Web Cache HTTP Listen port = 7777
Web Cache Administration port = 9400
Web Cache Invalidation port = 9401
Web Cache Statistics port = 9402
Application Server Control port = 1810
```

---

**Notes:** Ensure that these ports are not already in use by any other service on the computer. Using the Static Ports feature as described to install the Application Server Tier ensures that the port assignments will be consistent with the documentation in this section, if the ports are correctly specified in the file and the port is not already in use. Otherwise:

- If a port is incorrectly specified, then the Oracle Universal Installer will assign the default port.
- If a port is already in use, then the Oracle Universal Installer will assign the next available port.

See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-2 for more information.

Port 80 is open on the firewall only to accept and redirect requests using the HTTP (non-secure) protocol. Requests using the HTTP protocol (in the form `http://www.mycompany.com`) are redirected to port 443. Requests using the HTTPS, or secure, protocol (in the form `https://www.mycompany.com`) are managed by port 443.

---



3. Install a Portal and Wireless Oracle Application Server 10g middle tier on the first computer. During the installation:
  - Use the Infrastructure installed in Step 1 of [Section 9.1.1](#).
  - Clear the selection for **OracleAS Portal** in the **Select Configuration Options** screen.

---

**Note:** Selecting the Oracle Application Server 10g Portal option in this screen now will overwrite the previously created configuration entries. For more information, refer to the *Oracle Application Server Portal Configuration Guide*, section titled "Configuring OracleAS Portal During and After Installation".

---

4. In the **Select OracleAS 10g Metadata Repository**, select the connect string for the application Metadata Repository database (on APPDBHOST1 and APPDBHOST2) from the drop-down list.
5. Configure **OracleAS Portal**, accessing the Application Server Control Console. To configure OracleAS Portal, perform the following steps:
  - a. On the Oracle Application Server home page, click the **Configure Component** button.
  - b. Select **Portal** from the drop-down list on the **Select Component** page and click **Continue**.

The Login page appears.

- c. Enter the administration password for the Oracle Application Server instance in the **Administration Password** field.
- d. Click **Finish**.

The OracleAS Portal middle-tier components are deployed, but information in the OracleAS Metadata Repository is not overwritten.
- e. When the configuration is complete, click **OK**. The Oracle Application Server home page is displayed.
- f. Verify that the OC4J\_Portal and Portal:portal instances appear in the **System Components** table.
- g. Verify that the **OC4J\_Portal** and **Portal:portal** instances are running:
  - Click **OC4J\_Portal** and verify that the OC4J\_Portal page appears.
  - Click **Portal:portal** and verify that the Portal page appears.
- h. Restart Oracle HTTP Server and OC4J\_Portal.
- i. Verify that the installation is accessible at the following URL:

**`http://apphost3.mycompany.com:7777`**

### 9.1.3 Installing the Second Internal Middle Tier on APPHOST4

Follow the steps in section [Section 9.1.2, "Installing the First Internal Middle Tier on APPHOST3"](#) to install the second internal middle tier, substituting apphost4 where applicable.

---

**Note:** It is recommended that you use the same physical path for installing the second middle tier. This helps when you make configuration changes on one computer and want to transfer the changes to another computer. If the physical path is different on other computers, you must ensure that the path elements are corrected after copying the files.

---

### 9.1.4 Configuring an OracleAS Web Cache Invalidation-only Cluster

You must configure an OracleAS Web Cache invalidation-only cluster that includes the OracleAS Web Cache instances from both the internal and external computers. In this cluster configuration, invalidation requests are propagated across all cache cluster members. However, the OracleAS Web Cache invalidation-only cluster does not forward other requests between the cluster members, so while processing user requests, each cluster member acts as an individual cache and does not request objects from peer cluster members.

This configuration can be used to simplify the administration of many caches, especially in a cluster whose members are separated by a firewall. For example, a cluster can have two caches located on either side of a firewall that separates the intranet from the Internet.

#### 9.1.4.1 Preparing the Network Environment for the OracleAS Web Cache Invalidation-only Cluster

Before configuring the OracleAS Web Cache invalidation-only cluster between the external and internal OracleAS Web Cache instances, perform the following checks:

1. Ensure that all external OracleAS Web Cache instances can resolve and contact all internal OracleAS Web Cache instances and vice versa. This can be done using the `ping` network command.
2. Ensure that the invalidation port (9401) is open in the firewall in one direction only from the internal OracleAS Web Cache instance to the external OracleAS Web Cache instance.
3. Ensure that the administration port (9400) is open in the firewall in both directions.

---

**Note:** After the configuration is complete, the administration port (9400) should be closed to traffic from the external middle tiers to the internal middle tiers.

---

4. Ensure that you can use `telnet` to send network packets from the internal to the external OracleAS Web Cache ports.

### 9.1.4.2 Configuring the Caches

This section explains how to manage the caches as a cluster and segregate cache content, using the OracleAS Web Cache Manager on APPHOST1 to configure settings for a cache cluster.

1. In the navigator frame, select **Properties > Clustering**.

The Clustering page appears. The General Cluster Information section displays the default clusterwide values for failover and invalidation propagation. The Cluster Members table displays the external middle tier caches (APPHOST1 and APPHOST2).

2. In the **General Cluster Information** section of the Clustering page, click **Edit**.

The **Edit General Cluster Information** dialog box appears.

3. In the **Propagate Invalidation** field, select **Yes** to indicate that you want invalidation requests from cache cluster members to be propagated to other cache cluster members.

4. Click **Submit**.

5. In the **Cluster Members** table of the **Clustering** page, default values are displayed for the current cache. Select the APPHOST1 cluster member and click **Edit Selected**.

The **Edit Cluster Member** dialog box appears.

6. In the **Capacity** field, enter 0.

---

**Note:** If you assign a capacity of 0 to *all* cluster members, no requests will be forwarded between cluster members. With this setup, you can propagate the configuration and invalidation across all cache cluster members, simplifying the administration of many caches.

---

7. Click **Submit**.

8. Return to the **Cluster Members** table of the **Clustering** page and select APPHOST2.

9. In the **Capacity** field, enter 0.

10. Click **Submit**.

Before you can add APPHOST3 and APPHOST4 to the cluster, the following conditions must be in effect:

- The cache must be started.
- The administrator password of the cache to be added must be the same as the administrator password of the cache on APPHOST1. If it is different, you must connect to the cache's admin server and modify the administration password. For more information, refer to "Task 2: Modify Security Settings" in Chapter 8, "Setup and Configuration" in Oracle Application Server Web Cache Administrator's Guide.

**9.1.4.2.1 Adding Caches to the Invalidation-Only Cluster** You must now add the APPHOST3 and APPHOST4 caches to the cluster using OracleAS Web Cache Manager on APPHOST1.

To add a cache to the cluster in OracleAS Web Cache Manager:

1. In the navigator frame, select **Properties > Clustering**.  
The **Clustering** page appears.
2. In the **Cluster Members** section of the Clustering page, click **Add**.  
The Add Cache to Cluster dialog box appears.
3. In the **Host Name** field, enter `apphost3.mycompany.com` as the host name of the cache to be added to the cluster.
4. In the **Admin Port** field, enter `9400` for the administration port for the cache to be added to the cluster.
5. In the **Protocol for Admin Port** field, select either **HTTP** to accept HTTP browser requests.
6. In the **Cache Name** field, enter `apphost3.mycompany.com-webcache`.
7. Click **Submit**.  
The cache is now part of the cluster and is listed in the **Cluster Members** table.
8. Repeat Steps 2 through 7, substituting `apphost4` in the **Host Name** and **Cache Name** fields.
9. Click **Apply Changes**.  
OracleAS Web Cache adds the cache-specific information from the new cache cluster members to the cluster configuration.
10. For each cluster member, set the Capacity to 0. To do this, select **Properties**, then **Clustering**. Select a cluster member and click **Edit**. In the **Edit Cluster Member** dialog box, set the Capacity to 0.
11. Propagate the configuration to all cluster members.

When you modify the cluster and apply changes, OracleAS Web Cache adds the cache-specific information from the new cache cluster members to the configuration. For those changes to take affect in all cluster members, you must propagate the configuration and restart the cache server process of the cluster members.

To propagate the configuration to new cluster members in OracleAS Web Cache Manager:

- a. In the navigator frame, select **Operations > Cache Operations**.  
The Cache Operations page appears. The **Operation Needed** column indicates the caches to which the configuration should be propagated.
- b. Propagate the configuration to all cache cluster members:
  - Select **All caches** in the **Operate On** field.
  - Select an **Interval** of **Immediate**. (No other interval is allowed for propagation.)
  - Click **Propagate**.

When the operation completes, the **Operation Needed** column in the Cache Operations page indicates the cluster members that need to be restarted.

- c. Stop and restart all cluster members:
  - Select **All caches** in the **Operate On** field.
  - Select an **Interval** to stagger the time that operation begins on the caches, and then click **Restart**.

When the operation completes, the **Operation Needed** column in the Cache Operations page indicates that no operations are needed. The cache cluster is ready to use.

12. Ensure that the administration and invalidation ports are closed to traffic coming from outside the network.

### 9.1.4.3 Disabling External to Internal Communication Through the Firewall

To disable external to internal communication through the firewall, perform the following steps:

1. Disable the administration port from external middle tier to internal middle tier.
2. Ensure that the network packets cannot be sent from the external to the internal OracleAS Web Cache administration and invalidation ports, using telnet.
3. Ensure that network packets can be sent from the internal to the external OracleAS Web Cache for both the administration and invalidation ports.

The communication paths and ports should now be as listed in [Table 9–1](#):

**Table 9–1 Communication Path and Ports Used by Network Packets**

Communication Path	Ports to be enabled
Internal WebCache 1 to External WebCache 1	Port 9400 and Port 9401
Internal WebCache 2 to External WebCache 1	Port 9400 and Port 9401
Internal WebCache 1 to External WebCache 2	Port 9400 and Port 9401
Internal WebCache 2 to External WebCache 2	Port 9400 and Port 9401

**Note:** For network security reasons, you should perform any additional cluster configuration from a Web Cache instance on one of the internal middle tiers. Any Web Cache instance in the cluster can be used to administer the cluster, but if you want to use an external OracleAS Web Cache instance, you must temporarily open the administration port in the firewall to allow external to internal traffic.

### 9.1.5 Configuring the First Internal Middle Tier on APPHOST3 for Load Balancing Router Access

1. Configure the Load Balancing Router to accept requests on port 80 and forward them to the OracleAS Web Cache port (7777) running on APPHOST3. To do this:
  - a. Set up a group, or pool, on the Load Balancing Router, to which individual servers can be added. See the Load Balancing Router documentation for instructions on how to do this.
  - b. Add the desired servers' IP addresses and port numbers to the group.
  - c. Create a virtual server that listens on port 80, and balances requests among the members of the group. See the Load Balancing Router documentation for instructions on how to do this.

- d. Ensure that the Load Balancing Router translates the port on which it is listening to forward requests to the port on which OracleAS Web Cache is listening.
  2. Configure the OracleAS Portal middle tier on APPHOST3 to allow underlying components to construct URLs based on the Load Balancing Router host name (`internal.mycompany.com`) and Load Balancing Router port number (80), so that self-referential URLs rendered on OracleAS Portal pages are valid for the browser. To do this, define a virtual host as follows:
    - a. Access the Oracle Enterprise Manager 10g Application Server Control Console.

Typically the Application Server Control Console is located at `http://www.xyz.com:1810`.
    - b. Click the link for the middle tier where OracleAS Portal is installed.
    - c. Ensure that the server name, `internal.mycompany.com`, is listed in the table.
    - d. Click the **Administration** link.
    - e. Click **Advanced Server Properties**.
    - f. Select the **httpd.conf** file.
    - g. Add a VirtualHost container, as shown in the following example:

```
NameVirtualHost *:7778

<VirtualHost *:7778>
    ServerName internal.mycompany.com
    Port 80
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
</VirtualHost>
```
    - h. Click **Apply**.
    - i. When prompted to restart Oracle HTTP Server, click **No**.
3. Define a second virtual host, using the same steps as for the first, with the following exceptions:
  - Specify `apphost3.mycompany.com` as the **Server Name**.
  - Specify 7777 for the Port directive in the VirtualHost container.
  - When prompted to restart the Oracle HTTP Server, click **Yes**.
4. Define a site that matches the virtual host entry created in the previous step, using OracleAS Web Cache Manager on APPHOST3, as follows:
  - a. Access the OracleAS Web Cache Manager on APPHOST3, as described in the *Oracle Application Server Web Cache Administrator's Guide*.
  - b. From **Properties**, click **Sites**.
  - c. Click **Create** under **Named Sites Definitions**.
  - d. On the **Create Named Site** page, specify `internal.mycompany.com` for the **Host** and 80 for **Port**. Keep the default values for all other fields.
  - e. Click **OK**. `internal.mycompany.com` now appears in the **Named Sites Definitions** table.

5. Use OracleAS Web Cache Manager on APPHOST1 to add APPHOST3 as an origin server to the OracleAS Web Cache cluster created in [Section 9.1.4.2, "Configuring the Caches"](#) on page 9-7. This will update the routing table. To add APPHOST3, follow these steps:
  - a. Click **Origin Server** under **Origin Servers, Sites, and Load Balancing**.
  - b. In the **Origin Server** page, click **Add** under the **Application Web Servers** table.
  - c. In the **Add Application Web Server** page, provide the following information:

Property	Value
Hostname	apphost3.mycompany.com
Port	7778 (APPHOST3 Oracle HTTP Server listening port)
Routing	ENABLED
Capacity	100
Failover Threshold	5
Ping URL	/
Ping Interval	10
Protocol	HTTP

- d. Click **Submit**.
- e. To verify that the origin server has been added properly, locate `apphost3.mycompany.com` in the **Origin Server** table.

---

**Note:** Refer to the section "Map Sites to Origin Servers" in *Oracle Application Server Web Cache Administrator's Guide*, for more information.

---

6. Use OracleAS Web Cache Manager on APPHOST1 to map the site `internal.mycompany.com` to the middle tier `apphost3.mycompany.com`.
  - a. In the navigation frame, select **Site-to-Server Mapping** under **Origin Servers, Sites, and Load Balancing**.
  - b. In the **Site-to-Server Mapping** page, select the first mapping in the table and click **Insert Above**.
  - c. In the **Edit/Add Site-to-Server Mapping** page, select the **Select from Site definitions** option and then select `internal.mycompany.com`.
  - d. In the **Select Application Web Servers** section, select the application server on APPHOST3 (`apphost3.mycompany.com`) specified in the **Origin Servers** page.
  - e. Click **Submit**.
  - f. Click **Apply Changes** on the top of the page.
  - g. In the **Cache Operations** page, click **Restart** to restart OracleAS Web Cache on APPHOST3.

To verify that the site has been mapped correctly, navigate to the **Site-to-Server Mapping** page, and ensure that APPHOST3 is mapped to the site `internal.mycompany.com`.

7. Configure the `apphost3.mycompany.com` computer so that it can resolve the Load Balancing Router hostname to have the correct IP address. You can use DNS resolution, or create an entry in the `/etc/hosts` file as follows:

```
xxx.xxx.240    internal.mycompany.com
```

---

---

**Note:** Ensure that the `/etc/hosts` file does not have an entry that points the local hostname to `127.0.0.1`. For example:

```
127.0.0.1 apphost3.mycompany.com
```

---

---

8. Configure the Load Balancing Router to perform Network Address Translation (NAT) bounce back for loopback requests coming from the Parallel Page Engine running on `apphost3.mycompany.com`. This ensures that when the Parallel Page Engine makes a loopback request to OracleAS Web Cache, there are no errors.

---

---

**Notes:**

- NAT bounce back is set up differently on individual Load Balancing Routers. See the Load Balancing Router configuration guide on how to set this up.
  - The log files contain the NAT bounce back addresses for all loopback requests from the Parallel Page Engine, that get forwarded to OracleAS Web Cache or Oracle HTTP Server through the Load Balancing Router.
- 
- 

9. Configure the Load Balancing Router (`internal.mycompany.com`) to accept invalidation requests from the OracleAS Metadata Repository on a separate port (9401 in this example), so that it is forwarded to the OracleAS Web Cache running on computer `apphost3.mycompany.com` on port 9401.

---

---

**Notes:** This procedure load balances invalidations sent from the OracleAS Metadata Repository to the internal Web Cache instances. If high availability is a concern, then the Load Balancing Router may be configured to load balance to both the internal and external Web Cache instances. However, the internal Web Cache instances should be given a member priority of 1 and the external Web Cache instances a member priority of 2. The Minimum Active Members should be set to 1. With this configuration, the Load Balancing Router will load balance across the internal Web Cache instances first, and only load balance to an external Web Cache instance if all internal Web Cache instances are unavailable.

The Load Balancing Router does not have to listen on the OracleAS Web Cache invalidation port. On Load Balancing Routers that do not have port mapping ability, the port number must match the OracleAS Web Cache invalidation port.

---

---

- a. Set up a group, or pool on the Load Balancing Router to which individual servers can be added.



- b. Add the desired servers' IP addresses and port numbers to the group.
- c. Create a virtual server that listens on port 9401, and balances requests between the members of the group.

---

**Notes:**

- If the Load Balancing Router's port that is listening for the invalidation requests and the OracleAS Web Cache invalidation port are different, you must ensure that the Load Balancing Router translates the port that it is listening on to forward requests to the port that OracleAS Web Cache is listening on.
  - Use the Load Balancing Router documentation to set up the groups, and virtual server.
  - If the Oracle Application Server Infrastructure is behind another firewall, you must ensure that it can send invalidation messages to the Load Balancing Router.
  - For security reasons, the invalidation port on the Load Balancing Router (port 9401) must not be accessible from outside the network.
- 

10. Copy the configuration settings for OracleAS Portal from the middle tier APPHOST1, to the middle tier APPHOST3. It is a good idea to make backup copies of the files on APPHOST3 first. To do this, perform the following steps:
  - a. Copy `ORACLE_HOME/Apache/modplsql/conf/dads.conf` from APPHOST1 to APPHOST3.
  - b. Copy `ORACLE_HOME/Apache/oradav/conf/oradav.conf` from APPHOST1 to APPHOST3.
  - c. Copy `ORACLE_HOME/Apache/modplsql/conf/cache.conf` from APPHOST1 to APPHOST3.
  - d. Edit `MID_TIER_ORACLE_HOME/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF/web.xml` on both middle tiers to add the `init-param` entries as shown:

```
<servlet>
  <servlet-name>page</servlet-name>
  <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
  <init-param>
    <param-name>useScheme</param-name>
    <param-value>http</param-value>
  </init-param>
  <init-param>
    <param-name>usePort</param-name>
    <param-value>80</param-value>
  </init-param>
</servlet>
```

- e. Copy the `ORACLE_HOME/portal/conf/iasconfig.xml` file from APPHOST1 to APPHOST3.
11. Edit the `ORACLE_HOME/portal/conf/iasconfig.xml` file to specify the correct *farmname*, *hostname*, and *port* used to access OracleAS Portal, and to perform the OracleAS Web Cache invalidation, as shown in [Example 9-1](#) (all changes are shown in bold):

**Example 9–1 iasconfig.xml File Edited to Include Farm Element**

```

<IASConfig XSDVersion="1.0">

    <IASFarm Name="Farm-1.internal.mycompany.com" Host="internal.mycompany.com">
        <WebCacheComponent ListenPort="80" InvalidationPort="9401"
        InvalidationUsername="invalidator" InvalidationPassword="welcome1"
        SSLEnabled="false"/>
    </IASFarm>

    <IASInstance Name="ias.login.mycompany.com" Host="login.mycompany.com">
        <OIDComponent AdminPassword="@BV52KPJEW5a014n81bTxUY="
        PortSSLEnabled="true" LDAPPort="3060" AdminDN="cn=orcladmin"/>
    </IASInstance>

    <IASInstance Name="ias-1.apphost3.mycompany.com"
    Host="apphost3.mycompany.com">
        <EMComponent ConsoleHTTPPort="1810" SSLEnabled="false"/>
    </IASInstance>

    <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
    SchemaPassword="@Beyh8p2bOWELQC5a5zRtuYc="
    ConnectString="cn=iasdb,cn=oraclecontext">
        <WebCacheDependency ContainerType="IASFarm"
        Name="Farm-1.internal.mycompany.com"/>
        <OIDDependency ContainerType="IASInstance"
        Name="ias.login.mycompany.com"/>
        <EMDependency ContainerType="IASInstance"
        Name="ias-1.apphost3.mycompany.com"/>
    </PortalInstance>

</IASConfig>

```

---

**Note:** If OracleAS Web Cache on ias-1.apphost3.mycompany.com (shown in [Example 9–1](#)) is not referenced by any other OracleAS Portal instance, you can remove the entry from iasconfig.xml, as seen in [Example 9–1](#).

---

12. To enable monitoring of the Load Balancing Router's front-end host and port settings for OracleAS Portal, you must edit `ORACLE_HOME/sysman/emd/targets.xml` on APPHOST3, as follows:

- a. Open `targets.xml`, using a text editor.
- b. Search for OracleAS Portal targets, that is, `TYPE="oracle_portal"`.
- c. Edit the `PortalListeningHostPort` property, to point to the Load Balancing Router. For example:

```

<Property NAME="PortalListeningHostPort"
VALUE=http://internal.mycompany.com:80/>

```

- d. Save the changes to `targets.xml`.
- e. Reload the targets in the Application Server Control Console by issuing this command in `ORACLE_HOME/bin/`:

```
emctl reload
```

13. Ensure that the `ORACLE_HOME` environment variable is set.

14. Encrypt any plain text passwords in the `iasconfig.xml` configuration file. To do this, navigate to `MID_TIER_ORACLE_HOME/portal/conf`, and issue this command:

```
ptlconfig -encrypt
```

15. Save the manual configuration changes in the Distributed Configuration Management repository by issuing the following command on APPHOST3 in `ORACLE_HOME/dcm/bin`:

```
dcmctl updateconfig -ct ohs
```

16. Use the Application Server Control Console to access the `mod_plsql` configuration pages.

17. Select the portal DAD and click **Edit**.

18. Click **Apply**.

The required `mod_rewrite` and `mod_oc4j` directives are added.

19. Restart Oracle HTTP Server on APPHOST3 by issuing this command in `ORACLE_HOME/opmn/bin`:

```
opmnctl restartproc type=ohs
```

### 9.1.6 Configuring the Second Internal Middle Tier on APPHOST4 for Load Balancing Router Access

1. Configure the OracleAS Portal middle tier to allow underlying components to construct URLs based on the Load Balancing Router hostname (`internal.mycompany.com`) and Load Balancing Router port number (80). To do this, define a virtual host by performing the following steps:

- a. Access the Oracle Enterprise Manager 10g Application Server Control Console.

Typically the Application Server Control Console is located at `http://www.xyz.com:1810`. Refer to Chapter 7, "Monitoring and Administering OracleAS Portal" in *Oracle Application Server Portal Configuration Guide* for more information about using the Application Server Control Console.

- b. Click the link for the middle tier where OracleAS Portal is installed.
- c. Ensure that the server name `internal.mycompany.com`, is listed in the table.
- d. Click the **Administration** link.
- e. Click **Advanced Server Properties**.
- f. Select the `httpd.conf` file.
- g. Create a VirtualHost container as shown in the following example:

```
NameVirtualHost *:7778

<VirtualHost *:7778>
    ServerName internal.mycompany.com
    Port 80
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
</VirtualHost>
```

- Note:** If APPHOST3 and APPHOST4 are installed using different physical paths, you must ensure that the path elements are corrected after copying the files.

- ```
dcmtcl updateconfig
```

- The required `mod_rewrite` and `mod_oc4j` directives are added.

- ```
opmnctl restartproc type=ohs
```

- ```
L1.L1.L1.L1      internal.mycompany.com
```

---

**Note:** Ensure that the `/etc/hosts` file does not have an entry that points the local hostname to `127.0.0.1`. For example:

```
127.0.0.1 apphost4.mycompany.com
```

---

10. Use OracleAS Web Cache Manager on APPHOST3 to add APPHOST4 to the OracleAS Web Cache cluster created in [Section 9.1.4.2, "Configuring the Caches"](#) on page 9-7. This will update the routing table. To add APPHOST4, follow these steps:
  - a. Click **Origin Server** under **Origin Servers, Sites, and Load Balancing**.
  - b. In the **Origin Server** page, click **Add** under the **Application Web Servers** table.
  - c. In the **Add Application Web Server** page, provide the following information:

| Property           | Value                                             |
|--------------------|---------------------------------------------------|
| Hostname           | apphost4.mycompany.com                            |
| Port               | 7778 (APPHOST4 Oracle HTTP Server listening port) |
| Routing            | ENABLED                                           |
| Capacity           | 100                                               |
| Failover Threshold | 5                                                 |
| Ping URL           | /                                                 |
| Ping Interval      | 10                                                |
| Protocol           | HTTP                                              |

- d. Click **Submit**.
- e. To verify that the origin server has been added properly, locate `apphost4.mycompany.com` in the **Origin Server** table.

---

**Note:** Refer to the section "Map Sites to Origin Servers" in *Oracle Application Server Web Cache Administrator's Guide*, for more information.

---

11. Use OracleAS Web Cache Manager on APPHOST3 to map the Load Balancing Router site `internal.mycompany.com` to the two origin servers `apphost3.mycompany.com` and `apphost4.mycompany.com`, by performing the following steps:
  - a. In the navigation frame, select **Site-to-Server Mapping** under **Origin Servers, Sites, and Load Balancing**.
  - b. On the **Site-to-Server Mapping** page, Select the mapping for the Load Balancing Router site in the table and click **Edit Selected**.
  - c. In the **Select Application Web Servers** section, select an application Web server specified in the **Origin Servers** page for APPHOST4 (`apphost4.mycompany.com`).
  - d. Click **Submit**.

- e. To verify that the site has been mapped correctly, ensure that both APPHOST3 and APPHOST4 are mapped to the site `internal.mycompany.com` in the **Site to Server Mappings** table.

---

**Note:** Refer to the section "Map Sites to Origin Servers" in the *Oracle Application Server Web Cache Administrator's Guide*, for more information.

---

- 12. Click **Apply Changes** on the top of the page. Perform the following steps in the **Cache Operations** page:
  - a. Click **Propagate** to propagate changes to APPHOST4.
  - b. Click **Restart** to restart Web Caches on APPHOST3 and APPHOST4.
- 13. Configure the Load Balancing Router (`internal.mycompany.com`) to forward requests on the invalidation port to OracleAS Web Cache on the second middle tier `apphost4.mycompany.com` on port 9401 (similar to the previous configuration on the first middle tier `apphost3.mycompany.com`).

---

**Notes:** This procedure load balances invalidations sent from the OracleAS Metadata Repository to the internal Web Cache instances. If high availability is a concern, then the Load Balancing Router may be configured to load balance to both the internal and external Web Cache instances. However, the internal Web Cache instances should be given a member priority of 1 and the external Web Cache instances a member priority of 2. The Minimum Active Members should be set to 1. With this configuration, the Load Balancing Router will load balance across the internal Web Cache instances first, and only load balance to an external Web Cache instance if all internal Web Cache instances are unavailable.

The Load Balancing Router does not have to listen on the OracleAS Web Cache invalidation port. On Load Balancing Routers that do not have port mapping ability, the port number must match the OracleAS Web Cache invalidation port.

---

- 14. Configure the Load Balancing Router (`internal.mycompany.com`) to forward requests on port 80 to OracleAS Web Cache on `apphost2.mycompany.com` on port 7777 (similar to the previous configuration on the first middle tier `apphost3.mycompany.com`).

---

**Note:** Use the Load Balancing Router documentation to complete this step.

---

- 15. Configure the Load Balancing Router to perform Network Address Translation (NAT) bounce back for loopback requests coming from Oracle HTTP Server on `apphost4.mycompany.com`. Refer to Step 6 in "[Configuring the First Internal Middle Tier on APPHOST3 for Load Balancing Router Access](#)" for more information.

---

**Note:** You can add more middle tiers by repeating the procedures in ["Installing the Second Internal Middle Tier on APPHOST4"](#) on page 9-6 and ["Configuring the Second Internal Middle Tier on APPHOST4 for Load Balancing Router Access"](#) on page 9-15, for each additional middle tier.

---

16. To enable monitoring of the Load Balancing Router's front-end host and port settings for OracleAS Portal, you must edit the `ORACLE_HOME/sysman/emd/targets.xml` file on APPHOST4, as follows:
- Open the `targets.xml` file, using a text editor.
  - Locate the OracleAS Portal targets (that is, `TYPE="oracle_portal"`).
  - Edit the `PortalListeningHostPort` property, to point to the Load Balancing Router. For example:

```
<Property NAME="PortalListeningHostPort"
VALUE=http://internal.mycompany.com:80/>
```

- Save the changes to `targets.xml`.
- Reload the targets in the Application Server Control Console by issuing this command in `ORACLE_HOME/bin/`:

```
emctl reload
```

---

**Note:** This procedure load balances invalidations sent from the OracleAS Metadata Repository to the internal Web Cache instances. If high availability is a concern, then the Load Balancing Router may be configured to load balance to both the internal and external Web Cache instances. However, the internal Web Cache instances should be given a member priority of 1 and the external Web Cache instances a member priority of 2. The Minimum Active Members should be set to 1. With this configuration, the Load Balancing Router will load balance across the internal Web Cache instances first, and only load balance to an external Web Cache instance if all internal Web Cache instances are unavailable.

---

17. To verify if the internal middle tiers have been configured to work with the internal Load Balancing Router, you must perform the following steps:
- Ensure that you can telnet to the listen and invalidation ports on the internal Load Balancing Router. To do this, leave APPHOST3 running and stop APPHOST4. Verify that you are able to contact port 80 on `internal.mycompany.com` from the intranet and specifically, from the infrastructure database computer(s), `APPDBHOST*.mycompany.com`. The command to check this is:

```
telnet internal.mycompany.com 80
```

Ensure that no connection failure message is returned. Verify that the invalidation port can be reached in the same manner from the APPDB computer(s). The command used to check this is:

```
telnet internal.mycompany.com 9401
```

Perform the following tests:

- Access OracleAS Web Cache and Oracle HTTP Server through the LBR using the following URL:

<http://internal.mycompany.com>

- Test the connection to the OracleAS Metadata Repository through the LBR, by accessing the following URL:

<http://internal.mycompany.com/pls/portal/http.p?cbuf=test>

The response should be test. If this succeeds, then the Oracle Application Server middle tier can connect to the OracleAS Metadata Repository. If this test fails, then examine the Oracle HTTP Server `ORACLE_HOME/Apache/Apache/logs/error_log` file to determine the cause.

- Repeat the preceding steps with APPHOST3 stopped and APPHOST4 running.

### 9.1.7 Registering the Internal Middle Tier as a Partner Application

For the single sign-on component to work properly, it must always be referenced by a partner application with the same host name in the URL. This is because cookies are sent back only to the host that generated them. For example the Oracle Application Server Single Sign-On components must always be referenced as <http://login.mycompany.com>.

You must register `internal.mycompany.com` as a partner application. To do this, perform the following steps from an external middle tier, APPHOST1:

1. Add a partner application entry for `internal.mycompany.com` by executing the script `OH/portal/conf/ptlconfig`, as follows:

```
ptlconfig -dad portal -sso -host internal.mycompany.com -port 80
```

2. Edit the SSO registration script `ORACLE_HOME/sso/bin/ssoreg` as shown in [Example 9-2](#), and then execute it. [Example 9-2](#) shows the usage of `ssoreg.sh` on UNIX; on Windows, the script name is `ssoreg.bat`.

---

**Note:** The script shown in [Example 9-2](#) has multiple lines for readability only. When you execute the script, all parameters are on a single continuous line.

---

#### **Example 9-2 ssoreg Usage on UNIX**

```
ORACLE_HOME/sso/bin/ssoreg.sh
-site_name internal.mycompany.com
-mod_osso_url http://internal.mycompany.com
-config_mod_osso TRUE
-oracle_home_path ORACLE_HOME
-config_file ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
-admin_info cn=orcladmin
```

To verify whether the internal middle tier has been registered as a partner application, ensure that you can log in to the Portal at <http://internal.mycompany.com>.



### 9.1.8 Updating the Default JPDK Instance URL and Seeded Provider Group URLs

The Default JPDK Instance URL determines the middle tier on which a web provider is created. In this configuration, web providers must be created in the external middle tier.

Follow these steps to set the URL:

1. Log in to OracleAS Portal as the administrator (for example, PORTAL).
2. Click the **Administer** tab.
3. Click the **Portal** tab.
4. Click **Global Settings** in the **Services** portlet.
5. Click the **Configuration** tab.
6. For the **Default JPDK Instance** setting, change the URL to the external URL:  
`https://portal.mycompany.com/jpdk/servlet/soaprouter/`
7. Click **OK**.
8. Click the **Administer** tab.
9. Click the **Portlets** tab.
10. In the **Remote Provider Group** portlet, enter `oracle.ias.providers` in the **Name** field.
11. Click **Edit**.
12. Click the **Connection** tab.
13. Change the URL to:  
`http://portal.mycompany.com/jpdk/soaprouter/`
14. Click **OK**.
15. Repeat Steps 8 through 14, substituting `oracle.sample.providers` for the Remote Provider Group portlet name.

### 9.1.9 Configuring OracleAS Portal Invalidation Messages

You must configure OracleAS Web Cache invalidation messages to be sent from the OracleAS Portal schema in the OracleAS Metadata Repository to the internal middle tier. To do this, perform the following steps from an external middle tier, APHOST1:

1. Update the `iasconfig.xml` file. Add the `InvalidationHost` property to the `WebCacheDependency` element with the following parameters:  

```
<WebCacheDependency ContainerType="IASInstance"
Name="Farm1.portal.mycompany.com"
InvalidationHost="internal.mycompany.com"/>
```
2. Execute the `ptlconfig` script, located in the directory `MID_TIER_ORACLE_HOME/portal/conf`, in site mode, as shown in the following example:  

```
ptlconfig -dad portal -site -wc -em
```

---

**Note:** The administration URL, which is created using the OracleAS Web Cache host name that is on the Web Cache **Global Settings** tab, now points to the internal Web Cache instance. Any OracleAS Web Cache instance in the cluster can be used to administer the cluster, but if you want to use an external OracleAS Web Cache instance, you must temporarily open up the administration port in the firewall to allow external to internal traffic.

---

#### 9.1.9.1 Verifying the OracleAS Web Cache Invalidation Messages Configuration

To verify that OracleAS Web Cache invalidation messages have been configured to be sent from the OracleAS Portal schema in the OracleAS Metadata Repository to the internal middle tier, perform the following steps:

1. Log in to Portal as the administrator.
2. Click the **Administer** tab.
3. Click the **Portal** tab.
4. Click **Global Settings**.
5. Click the **Cache** tab. In the **Web Cache Host Settings** section, the information displayed should be the same as the one in the `iasconfig.xml` file.

To verify if OracleAS Web Cache invalidation messages have been configured, perform the following steps:

1. Ensure that `internal.mycompany.com` and `portal.mycompany.com` are being used here.
2. Ensure that you can log in to internal site and add a portlet to a page.
3. Ensure that the updated page shows new content when accessed from the internal site.
4. Ensure that you can log in to the external site and add a portlet to a page.
5. Ensure that the updated page shows new content when accessed from the external and internal sites.

You must perform these steps with one middle tier stopped in each of the external and internal parts of the Web Cache cluster. Then, repeat the steps with the other middle tier stopped.

#### 9.1.10 Configuring the OracleAS Portal Schema in the OracleAS Metadata Repository

Configure the OracleAS Portal schema in the OracleAS Metadata Repository to send host-independent invalidations. To do this, perform the following steps:

1. Log in to APPHost1 and run the script  
`OH/portal/admin/plsql/wwc/cachhii.sql` using SQL\*Plus.
2. Specify `on` at the prompt to enable host-independent invalidations.

### 9.1.11 Modifying the Oracle Text Base Search URL

When creating a URL item and specifying a relative URL, then that URL will be resolved relative to the `basehref` of the page on which the URL link is being rendered. For OracleAS Portal pages, the `basehref` is always `http://host:port/portal/pls/dad` where the host, port, and DAD reflect the middle tier host from which that page was accessed. Therefore, when these URLs are rendered on an OracleAS Portal page, they are resolved relative to the base URL, `http://host:port/portal/pls/dad`.

The URL content is indexed by Oracle Text so that it can be searched within OracleAS Portal and the content referenced by the URL can be retrieved. Indexing is done outside the context of a page request, so there is no incoming request from which to determine the `basehref` to use when resolving relative URLs. Therefore, it is necessary to specify the `basehref` to use during indexing. This `basehref` should be to a host that is accessible from the database server. In most cases, this is the internal host.

To set the Oracle Text Base search URL to the internal host URL, perform the following steps:

1. In the **Services** portlet, click **Global Settings** page.  
By default, the **Services** portlet is on the **Portal** sub-tab of the **Administer** tab on the **Portal Builder** page.
2. Select the **Search** tab.
3. Set the Base URL field under Oracle Text Base Search to `http://host:port/portal/pls/dad`, where host, port, and DAD are the host, port, and DAD for the internal host.

### 9.1.12 Enabling Session Binding on OracleAS Web Cache

The session binding feature in OracleAS Web Cache is used to bind user sessions to a given origin server to maintain state for a period of time. Although almost all components running in a default OracleAS Portal middle tier are stateless, session binding is required for two reasons:

- The Web Clipping Studio, used by both the OracleAS Web Clipping portlet and the Web Page Data Source of OmniPortlet, uses HTTP session to maintain state, for which session binding must be enabled.
- Enabling session binding forces all the user requests to go to a given OracleAS Portal middle tier, resulting in a better cache hit ratio for the portal cache.

---

**Note:** Regardless of whether you have configured an LBR in your topology, you must enable session binding in OracleAS Web Cache, if you have more than one middle tier. In this configuration OracleAS Portal does not require session binding to be set up on the LBR.

---

To make OracleAS Web Cache bind the portal user session to the OracleAS Portal middle tier, perform the following steps:

1. In OracleAS Web Cache Manager on **APPHOST3**, click **Session Binding** under **Origin Servers, Sites, and Load Balancing**.

2. In the **Session Binding** page, select the LBR site name (`internal.mycompany.com:80`) in the table, and then click **Edit Selected**.
3. From the **Please select a session** drop-down list, change the session value to **Any Set-Cookie**.
4. From the **Please select a session binding mechanism** drop-down list, select **Cookie-based**.
5. Click **Submit** to apply the new settings to the site `internal.mycompany.com:80`.
6. To save your configuration changes, click **Apply Changes** at the top of the page.
7. On the **Cache Operations** page, click **Propagate** to propagate the changes.
8. Click **Restart** to restart OracleAS Web Cache on **APPHOST3** and **APPHOST4**.

### 9.1.13 Configuring the Oracle Drive WebDAV Client

To configure the Oracle Drive WebDAV client, add the *DAVParam*, *ORAPORTALUIURL*, in the `oradav.conf` file. The `oradav.conf` file must be configured for both internal middle tiers. The syntax to add the parameter is as follows:

```
DavParam ORAPORTALUIURL
```

Here is an example of the *ORAPORTALUIURL* parameter setting in the `oradav.conf` file:

```
Options Indexes
```

```
DAV oracle
```

```
DAVDepthInfinity On
```

```
DavParam ORACONTAINERNAME wwdav
```

```
DavParam ORACookieMaxAge 28800
```

```
DavParam ORASERVICE cn=iasdb,cn=oraclecontext
```

```
DavParam ORAUSER portal
```

```
DavParam ORACRYPTPASSWORD BUpsf8IQI6Ow==
```

```
DavParam ORAPACKAGENAME portal.wwdav_api_driver
```

```
DavParam ORAPORTALUIURL
```

```
http://internal.mycompany.com/pls/portal/
```

The addition of this parameter enables the Oracle Drive WebDAV client to have the extra OracleAS Portal menu options for the correct middle tier.

The extra Oracle Drive menu options are visible when you right-click a folder or file in the Oracle Drive user interface. This displays a set of menu options such as Set Properties, Change Access Control, Preview Content, View Versions, and View Page. These OracleAS Portal menu options require a URL to be in context with the selected folder or file.

### 9.1.14 Validating the Completed Configuration

To verify that your complete configuration is working as expected, perform the following steps:

1. To clear the contents stored in OracleAS Web Cache, restart APPHOST3 and APPHOST4, as follows:
  - a. Access the Application Server Control Console, typically located at `http://apphost3.mycompany.com:1810`.
  - b. Click the APPHOST3 instance.
  - c. Click **Restart All**.
  - d. Repeat the steps for APPHOST4.
2. Test access to OracleAS Portal through the Load Balancing Router by completing the following steps:
  - a. Access the OracleAS Portal home page at `http://internal.mycompany.com/pls/portal`.
  - b. Click the portal login link.
  - c. Click some links in the portal.
  - d. Confirm that content is getting cached in OracleAS Web Cache. To do this, access the OracleAS Web Cache Manager on **APPHOST3** as described in *Oracle Application Server Web Cache Administrator's Guide*.

Under **Monitoring**, click **Popular Requests**. select **Cached** from the **Filter Objects** drop-down list, and click **Update**. If you accessed OracleAS Portal, you will see portal content (For example, URLs that contain `/pls/portal`).

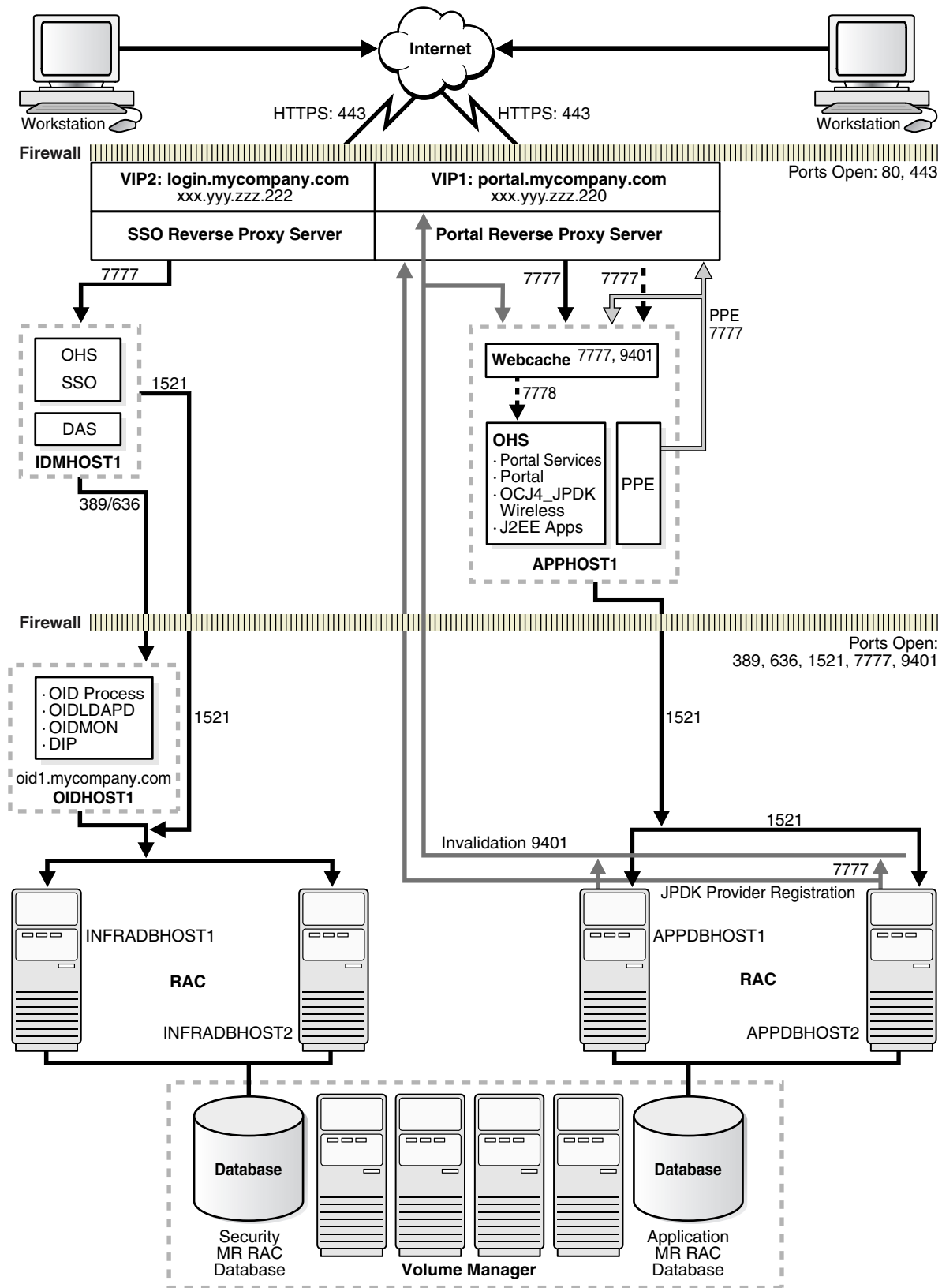
Perform some basic page edits in OracleAS Portal, such as adding a portlet to a page, and verify that the new content shows up. If the new content does not display properly, or errors occur, OracleAS Web Cache invalidation is misconfigured.

## 9.2 Configuring a Reverse Proxy for OracleAS Portal and OracleAS Single Sign-On

A reverse proxy is a server that appears to outside clients to be the content server. It relays requests from outside the firewall to servers behind the firewall, and delivers retrieved content back to the client. A firewall rule allows access only to the proxy server, so that the content servers are protected. The proxy server changes URLs listed in the headers of any messages generated by the content servers, so that external clients are given no information about the servers behind the firewall.

Figure 9-2 depicts the reverse proxy configuration described in this section. Only port 443 is designated as open on the firewall because in certain cases, port 80 may have to be closed. For example, if the IIS server is listening on both 443 and 80, and OracleAS Portal is using proxy:80 for Parallel Page Engine loopback, then port 443 must be open, and port 80 closed.

**Figure 9-2 Reverse Proxy Server Configuration**



This section explains how to configure a reverse proxy server for OracleAS Portal with OracleAS Single Sign-On. It contains these subsections:

[Section 9.2.1, "Install and Configure the Proxy Server" on page 9-27](#)

[Section 9.2.2, "Testing the OracleAS Single Sign-On Connection" on page 9-39](#)

[Section 9.2.3, "Configuring OracleAS Single Sign-On to Use a Reverse Proxy" on page 9-39](#)

[Section 9.2.4, "Validating the OracleAS Single Sign-On Configuration" on page 9-43](#)

[Section 9.2.5, "Testing the OracleAS Portal Connection" on page 9-43](#)

[Section 9.2.6, "Configuring OracleAS Portal for a Reverse Proxy" on page 9-44](#)

[Section 9.2.7, "Validating the OracleAS Portal Configuration" on page 9-52](#)

## 9.2.1 Install and Configure the Proxy Server

This section explains how to configure your choice of proxy server for the reverse proxy: OracleAS Web Cache, the Oracle HTTP Server, or the Internet Information Services (IIS) listener. (If necessary, you may also enable SSL communication on the Oracle HTTP Server, by following the instructions in the *Oracle HTTP Server Administrator's Guide*.)

Follow the instructions in one of these sections for the proxy server of your choice:

- [Section 9.2.1.1, "Configuring OracleAS Web Cache as a Reverse Proxy" on page 9-27](#)
- [Section 9.2.1.2, "Configuring the Oracle HTTP Server as a Reverse Proxy" on page 9-32](#)
- [Section 9.2.1.3, "Configuring Internet Information Services as a Reverse Proxy" on page 9-36](#)

### 9.2.1.1 Configuring OracleAS Web Cache as a Reverse Proxy

Configuring OracleAS Web Cache as a reverse proxy may be done using two proxy host computers, as depicted in [Figure 9-2, "Reverse Proxy Server Configuration"](#). The topology elements used on two proxy host computers for OracleAS Web Cache are listed in [Table 9-2](#). The instructions in this section explain how to define the login.mycompany.com site and map it to the idmhost.mycompany.com origin server, and define the portal.mycompany.com site and map it to the apphost1.mycompany.com origin server.

---

**Note:** Alternatively, if you have a proxy host computer with two IP addresses, you may use one OracleAS Web Cache instance that listens on two ports (one for OracleAS Single Sign-On and the other for OracleAS Portal). See the *Oracle Application Server Web Cache Administrator's Guide* for instructions. This is one of the advantages of using OracleAS Web Cache as a reverse proxy server instead of the Oracle HTTP Server or IIS.

---

**Table 9-2 Topology Elements for OracleAS Web Cache Reverse Proxy**

| Description                                                   | Value               |
|---------------------------------------------------------------|---------------------|
| OracleAS Web Cache proxy computer for OracleAS Single Sign-On | login.mycompany.com |

**Table 9–2 (Cont.) Topology Elements for OracleAS Web Cache Reverse Proxy**

| Description                                                     | Value                  |
|-----------------------------------------------------------------|------------------------|
| OracleAS Web Cache proxy computer for OracleAS Portal           | portal.mycompany.com   |
| OracleAS Single Sign-On computer                                | idmhost1.mycompany.com |
| OracleAS Single Sign-On computer Oracle HTTP Server listen port | 7777                   |
| OracleAS Portal computer                                        | apphost1.mycompany.com |
| OracleAS Portal OracleAS Web Cache listen port                  | 7777                   |
| OracleAS Portal Oracle HTTP Server listen port                  | 7778                   |

**9.2.1.1.1 Installing OracleAS Web Cache on the Proxy Computer** Install the standalone version of OracleAS Web Cache on login.mycompany.com and portal.mycompany.com, specifying 7777 as the listening port on each installation. OracleAS Web Cache is now listening on:

http://login.mycompany.com:7777

http://portal.mycompany.com:7777

**9.2.1.1.2 Applying the OracleAS Web Cache Patch** OracleAS Web Cache can function solely as a software load balancer (no caching functions are performed). You must configure this mode to use it as a reverse proxy server for OracleAS Portal.

See "OracleAS Web Cache as a Software Load Balancer" in the *Oracle Application Server Web Cache Administrator's Guide* for instructions on configuring OracleAS Web Cache in this manner.

**9.2.1.1.3 Creating Wallets for the Reverse Proxy Servers** The reverse proxy servers login.mycompany.com and portal.mycompany.com must have SSL credentials to serve requests. Follow these steps to create a wallet and obtain an SSL certificate:

1. Log in to the proxy host computer as the oracle user (the user that installed OracleAS Web Cache).
2. Start Oracle Wallet Manager by doing one of the following:  
(UNIX) Issue this command in the *WC\_ORACLE\_HOME/bin* directory:  
**./owm**  
(Windows) Select these options:  
**Start > Programs > Oracle OracleAS Web Cache installation instance name > Integrated Management Tools > Wallet Manger**
3. Click **Wallet**, then click **New**.  
A dialog box appears with the message:  
Your default wallet directory does not exist. Do you want to create it?
4. Click **Yes**.  
A dialog box may appear with the following message:  
Unable to create the system default wallet directory. Please contact your Oracle System Administrator for help. You can continue and create a wallet but you must save it in another location. Do you want to continue anyway?



5. Click **Yes**.
6. Provide a wallet password and confirm it. Leave the wallet type as **Standard**.  
A dialog box appears with the message:  
A new empty wallet has been created. Do you want to create a certificate request at this time?
7. Click **Yes**.
8. Type the following:  
Common Name: **login.mycompany.com**  
Organizational Unit: *organizational unit, such as IT*  
Organization: *organization name*  
Locality/City: *city*  
State/Province: *state or province*  
Country: *country*  
Key Size: **2048**
9. Click **OK**.  
A message appears, notifying you that the certificate was successfully created.
10. Copy the certificate request text from the body of this dialog and paste it into an e-mail message to send to a certificate authority, such as Verisign or Thawte.
11. Click **OK**.  
The main Wallet Manager window reappears; the certificate status is now **Requested**.
12. When the certificate authority sends you the certificate, you must import it into the wallet. Select and copy all of the lines in the certificate, including `---BEGIN NEW CERTIFICATE REQUEST---` and `---END NEW CERTIFICATE REQUEST---`.
13. Click **Operations**, then **Import User Certificate**.  
The **Import User Certificate** dialog box appears.
14. Click **Paste the Certificate**, then click **OK**.  
Another **Import User Certificate** dialog box appears with the following message:  
Please provide a base64 format certificate and paste it below.
15. Paste the certificate into the dialog box, and click **OK**.  
One of the following occurs:
  - If the certificate received is in PKCS#7 format, it is installed, and all the other certificates included with the PKCS#7 data are placed in the Trusted Certificate list.
  - If the certificate received is *not* in PKCS#7 format, and the certificate of its CA is not already in the Trusted Certificates list, then more must be done. Oracle Wallet Manager will ask you to import the certificate of the CA that issued your certificate. This CA certificate will be placed in the Trusted Certificates list. (If the CA certificate was already in the Trusted Certificates list, your certificate is imported without additional steps.)

A message at the bottom of the window confirms that the certificate was successfully installed. The Oracle Wallet Manager main window reappears, and the status of the wallet changes to **Ready**.

16. Click **Wallet**, then **Save as**. Specify a location that is accessible to you (for example, `WC_ORACLE_HOME/mycompany/wallet`. If the directory does not exist, you can create it now.
17. Click **Wallet**, then check the **Auto Login** checkbox, then click **Save As**.  
The following message appears:  
A wallet already exists in the selected location. Do you want to overwrite it?
18. Click **Yes**.
19. Repeat all of the preceding steps on the `portal.mycompany.com` computer, specifying **portal.mycompany.com** as the Common Name in Step 8.
20. Click **Wallet**, then **Exit**.

**9.2.1.1.4 Configuring the OracleAS Web Cache Listen Port** Follow these steps to configure OracleAS Web Cache to listen on port 443:

1. Access the Web Cache Administrator at:  
**`http://login.mycompany.com:9400/webcacheadmin`**  
The Web Cache Administrator password dialog appears.
2. Enter the OracleAS Web Cache administrator password.

---

**Note:** At installation time, The OracleAS Web Cache administrator password is set to the same password as the `ias_admin` password. The OracleAS Web Cache administrator password must be identical for all cache cluster members.

---

The **Web Cache Cache Operations** page appears. A scrollable frame on the left side of the window contains groups of configuration elements. To access an element, click its link. The content area of the page is then populated with the values for that element.

3. Under **Ports**, click **Listen Ports**.
4. Click **Add**.
5. Enter the following:  
IP Address: **ANY**  
Port Number: **443**  
Protocol: **HTTPS**  
Wallet: **`WC_ORACLE_HOME/mycompany/wallet`**
6. Click **Apply Changes**.
7. Exit the Web Cache Administrator.
8. Stop OracleAS Web Cache by issuing this command in `ORACLE_HOME/opmn/bin`:  
**`opmnctl stopproc ias-component=WebCache`**

9. Start OracleAS Web Cache as the root user by issuing this command in *ORACLE\_HOME/webcache/bin*:

```
webcachesetuser.sh setroot oracle user
```

In the preceding command, *oracle user* is the user that performed the OracleAS Web Cache installation.

10. Repeat all of the preceding steps on the portal.mycompany.com computer.

**9.2.1.1.5 Configuring Sites, the Origin Server, and Site-to-Server Mappings** Follow these steps to configure the necessary site definitions and mappings:

1. Access the Web Cache Administrator at:

```
http://login.mycompany.com:9400/webcacheadmin
```

The Web Cache Administrator password dialog appears.

2. Enter the OracleAS Web Cache administrator password.

---

**Note:** At installation time, The OracleAS Web Cache administrator password is set to the same password as the ias\_admin password. The OracleAS Web Cache administrator password must be identical for all cache cluster members.

---

The **Web Cache Cache Operations** page appears. A scrollable frame on the left side of the window contains groups of configuration elements. To access an element, click its link. The content area of the page is then populated with the values for that element.

3. Click the **Site Definitions** link in the **Origin Servers, Sites and Load Balancing** section.

The **Site Definitions** window opens.

4. Click **Add Site**.

5. Enter the following information (leave other fields blank):

- Host name: **login.mycompany.com**
- Port: **443**
- Client-side Certificate: **Not required**
- Default Site: **Yes**
- Create Alias from Site Name with/without www: **No**

6. Click **Add Alias**.

The **Add Alias for Site** window opens.

7. Enter the following information:

- Host name: **login.mycompany.com**
- Port: **7777**

8. Click **Submit**.

9. Click the **Site-to-Server Mapping** link in the **Origin Servers, Sites, and Load Balancing** section.

The **Site-to-Server Mapping** page appears, in which you map the site and site alias to an origin server.

10. Select the first mapping in the table and click **Insert Above**.

The **Edit/Add Site-to-Server Mapping** page appears.

11. Select the **Select From Site Definitions** option.
12. Select **login.mycompany.com**.
13. Select **idmhost1.mycompany.com** in the **Select Application Web Servers** section.
14. Click **Submit**.
15. Remove unused mappings or entries containing the wild card character **\***.
16. Click **Apply Changes**.
17. Click **Restart**.
18. Repeat all of the preceding steps on the portal.mycompany.com computer, substituting portal.mycompany.com as the site and apphost1.mycompany.com as the origin server (**Select Application Web Servers** section).

### 9.2.1.2 Configuring the Oracle HTTP Server as a Reverse Proxy

To use the Oracle HTTP Server as a reverse proxy, first install the standalone version of the Oracle HTTP Server on the reverse proxy server computer. This section explains how to configure the Oracle HTTP Server to pass incoming requests to Oracle Application Server Single Sign-On, Oracle Delegated Administration Services and OracleAS Portal, and modify all HTTP headers so that only the identity of the reverse proxy server computer is visible to clients.

There are two ways to configure the Oracle HTTP Server as a reverse proxy: using the ProxyPass and ProxyPassReverse directives, or the RewriteRule directive with the [P] (force proxy) flag. The RewriteRule directive is a more flexible and powerful implementation of the proxy functionality.

**9.2.1.2.1 Using the ProxyPass, ProxyPassReverse, and ProxyPreserveHost Directives** These directives, used together, configure an external server to function as a reverse proxy. This section describes each directive, and gives an example of using them in the Enterprise Deployment configuration, in which the reverse proxy server computer is the local server, and APPHOST1 is the remote server.

The ProxyPass, ProxyPassReverse, and ProxyPreserveHost directives are implemented by mod\_proxy, so they must appear in the httpd.conf file at a location following the LoadModule directive that loads mod\_proxy. Similarly, the RewriteRule directive must appear at a location following the LoadModule directive that loads mod\_rewrite.

In Apache 2.0, the ftp and http protocols are handled by separate modules, so you must add this directive to the httpd.conf file:

```
LoadModule proxy_http_module modules/mod_proxy_http.so
```

**ProxyPass Directive**

This directive causes the reverse proxy server computer to appear to be a mirror of APPHOST1.

The syntax of the ProxyPass directive is:

```
ProxyPass path url
```

*path* is the name of a local virtual path.

*url* is a partial URL for the remote server.

The **Apache HTTP Server documentation** describes the ProxyPass directive in detail.

[Example 9-3](#) shows the modifications to make to the `ORACLE_HOME/Apache/Apache/conf/httpd.conf` file on the reverse proxy server computer to configure reverse proxy functionality. Note that the ProxyRequests directive must be set to `off` when you use the ProxyPass directive.

Use the ProxyPassReverse directive in conjunction with the ProxyPass directive to achieve reverse proxy functionality.

**ProxyPassReverse Directive**

This directive causes the Oracle HTTP Server to adjust the URL in the Location header on HTTP redirect responses. This is necessary in a reverse proxy configuration, so that the reverse proxy is not bypassed on HTTP redirects from the servers behind the reverse proxy.

The syntax of the ProxyPassReverse directive is:

```
ProxyPassReverse path url
```

*path* is the name of a local virtual path.

*url* is a partial URL for the remote server.

The **Apache HTTP Server documentation** describes the ProxyPassReverse directive in detail.

**ProxyPreserveHost Directive (Apache 2.0.31 and later)**

This directive, when set to `On`, directs the server to pass the `Host :` line from an incoming request to the proxied host instead of to the hostname specified by the ProxyPass directive.

This directive is normally set to `Off`. It is useful in configurations that require the original host header to be evaluated by the back end server.

**Example 9-3 *httpd.conf* file on the Reverse Proxy Server Computer with ProxyPass, ProxyPassReverse, and ProxyPreserveHost Directives**

```
# (Windows) Ensure that mod_proxy is loaded for these directives
LoadModule proxy_module modules/ApacheModuleProxy.dll

# (UNIX) Ensure that mod_proxy is loaded for these directives
LoadModule proxy_module libexec/mod_proxy.so
.
.
.
# (Apache 2.0)
LoadModule proxy_http_module modules/mod_proxy_http.so

ProxyRequests off
ProxyPass / http://apphost1.mycompany.com
ProxyPassReverse / http://apphost1.mycompany.com
# (Apache 2.0)
ProxyPreserveHost On
```

**9.2.1.2.2 Using the RewriteRule Directive** You can use the RewriteRule directive to define URL rewriting rules and to pass requests through `mod_proxy`, using the `[P]` (force proxy) flag. The combined abilities enable a more powerful, flexible implementation of the functionality provided by the ProxyPass directive.

The RewriteRule directive relies on `mod_proxy` and `mod_rewrite` for its implementation, so the directive must appear in the `httpd.conf` file at a location following the LoadModule directives that load `mod_proxy` and `mod_rewrite`.

The syntax of the RewriteRule directive used for reverse proxy functionality is:

```
RewriteRule pattern substitution [P]
```

*pattern* is a regular expression that is applied to the current URL (the current URL is the value of the URL when the rule is applied).

*substitution* is the string that replaces the original URL that *pattern* matched

The `[P]` (force proxy) flag is used as a third argument to invoke the proxy functionality.

---

---

**Note:** If multiple flags are used in a rewriting rule, ensure that:

- There are no spaces between the brackets (that the flags are separated only by commas within the brackets). For example:

```
[I,P]
```

```
not
```

```
[I, P]
```

- The `P` flag appears last, since it forces the substituted URL through the proxy module without processing any more rewriting rules.
- 
- 

This directive can be used multiple times, so that each occurrence of the directive defines a rewriting rule. At run time, the rewriting rules are applied in the order in which they are defined.

The **Apache HTTP Server documentation** describes the RewriteRule directive in detail.

[Example 9–4](#) shows the modifications to make to the `ORACLE_HOME/Apache/Apache/conf/httpd.conf` file on the reverse proxy server computer to configure reverse proxy functionality.

**Example 9–4 *httpd.conf* file on the Reverse Proxy Server Computer with RewriteRule Directive**

```
# (Windows) Ensure that mod_proxy and mod_rewrite are loaded for this directive
LoadModule proxy_module modules/ApacheModuleProxy.dll
LoadModule rewrite_module modules/ApacheModuleRewrite.dll

# (UNIX) Ensure that mod_proxy and mod_rewrite are loaded for this directive
LoadModule proxy_module libexec/mod_proxy.so
LoadModule rewrite_module libexec/mod_rewrite.so
.
.
.
ProxyPassReverse / http://apphost1.mycompany.com

RewriteEngine On
RewriteRule ^/(.*) http://apphost1.mycompany.com/$1 [P]

# (Apache 2.0)
ProxyPreserveHost On
```

**9.2.1.2.3 Using the X-FORWARDED-HOST Value with Apache v. 1.3 mod\_proxy** To ensure that the correct host name and port combination is passed to mod\_osso, you must place a mod\_perl script on the infrastructure instance to replace the header. (In Apache 2.0, this is not necessary, as the hostname header behavior is configurable with the ProxyPreserveHost directive.)

The Apache 1.3 version of mod\_proxy always replaces the host header that was sent by the browser with a host header that contains the host and port information from the back end server. This is unacceptable to mod\_osso, as it expects a match between the definition of the partner application and the host header.

When mod\_proxy replaces the host header, it preserves the original host header information in another header called X-FORWARDED-HOST. Therefore, you can replace the value of the host header actually sent with the desired value from X-FORWARDED-HOST.

Follow these steps on the computer hosting the Infrastructure instance:

1. Create a `mod_perl` script file, `ORACLE_HOME/perl/lib/site_perl/5.6.1/Apache/ForwardedHostReplace.pm`, with these contents:

```
package Apache::ForwardedHostReplace;

use Apache::Constants qw(OK DECLINED);
use Apache::URI ();
use strict;

sub handler {
    my $r = shift;
    my $forwardedhost = $r->header_in("x-forwarded-host");

    if ( $forwardedhost )
    {
        $r->header_in("host", $forwardedhost);
    }

    return DECLINED;
}

1;
```

2. Add these directives to `ORACLE_HOME/Apache/Apache/conf/httpd.conf`:

```
PerlModule Apache::ForwardedHostReplace
PerlHeaderParserHandler Apache::ForwardedHostReplace
```

3. Restart the Oracle HTTP Server.

### 9.2.1.3 Configuring Internet Information Services as a Reverse Proxy

To use the Internet Information Services (IIS) listener as a reverse proxy for an Oracle Application Server instance, you must configure it using the Oracle Application Server Proxy Plug-in for Oracle HTTP Server. The Oracle Application Server Proxy Plug-in enables you to use a third-party HTTP listener, such as IIS, to send requests to Oracle Application Server. The OracleAS Proxy Plug-in is a reverse HTTP proxy; it forwards incoming HTTP requests to an Oracle Application Server instance.

The proxy logic is provided as a plug-in, a shared library that is loaded by IIS. The plug-in uses APIs provided with IIS to directly handle HTTP requests, similar to the way in which modules are plugged into Oracle HTTP Server.

The Oracle HTTP Server can mimic the address and port that IIS is using. That is, when sending a request to Oracle HTTP Server, the proxy can be configured to send a different Host: HTTP header than the actual hostname and port that the request is being sent to, so that downstream applications are shielded from the introduction of the reverse proxy. Appendix A of the *Oracle HTTP Server Administrator's Guide* describes the proxy plug-in in detail.

This section explains how to configure the IIS listener for use as a reverse proxy server with Oracle Application Server components, using the Oracle Application Server Proxy Plug-in.

---

**Note:** Example host names are supplied in these instructions for clarity only; you may substitute names of your choice.

---



The instructions in this section assume existence of this computing environment:

- A computer behind a firewall with an Oracle Application Server Infrastructure installed, with host name `idmhost1.mycompany.com`
- A computer behind a firewall with an OracleAS Portal middle tier installed, with host name `apphost1.mycompany.com`
- A computer in front of a firewall, with host name `login.mycompany.com`, with the IIS server installed
- A computer in front of a firewall, with host name `portal.mycompany.com`, with the IIS server installed

**9.2.1.3.1 Installing the Oracle Application Server Proxy Plug-in** This section explains how to obtain and install the plug-in.

---

**Note:** Example directory and file names are supplied in these instructions for clarity only; you may substitute names of your choice.

---

1. Obtain the OracleAS Proxy Plug-in from the Oracle Application Server 10g Companion CD, in the `/plugins/iis/` directory.
2. Create a subdirectory, `oracleproxy`, in the configuration directory of the IIS listener on `login.mycompany.com` and `portal.mycompany.com`.
3. Place the OracleAS Proxy configuration files and shared libraries in the `oracleproxy` directory, and ensure that the IIS process has read and execute privileges to the `oracleproxy` directory.

**9.2.1.3.2 Configuring the Oracle Application Server Proxy Plug-in** A single configuration file controls the functionality of the OracleAS Proxy Plug-in. The presence of the configuration file in the IIS file system activates the functionality. This section explains how to create the proxy configuration files in the `oracleproxy` directory on each computer.

---

**Note:** Example file and parameter names are supplied in these instructions for clarity only; you may substitute names of your choice (some parameters contain host names, so ensure proper syntax). The example files include comments to describe the parameters.

---

1. On `login.mycompany.com`, use a text editor to create the proxy server definition file `oproxydef`, with the following parameters and values:

```
# Server names that the proxy plug-in will recognize.
oproxy.serverlist=ssoserver

# Hostname to use when communicating with a specific server.
oproxy.ssoserver.hostname=idmhost1.mycompany.com

# Port to use when communicating with a specific server.
oproxy.ssoserver.port=7777

# Hostname and port that clients use to access the third-party HTTP listener.
# This value will be passed as the Host: HTTP header.
oproxy.ssoserver.alias=login.mycompany.com
```

```
# Description of URL(s) that will be redirected to this server.
oproxy.ssoserver.urlrule=/*
```

2. On `portal.mycompany.com`, use a text editor to create the proxy server definition file `oproxydef`, with the following parameters and values:

```
# Server names that the proxy plug-in will recognize.
oproxy.serverlist=portal

# Hostname to use when communicating with a specific server.
oproxy.portal.hostname=apphost1.mycompany.com

# Port to use when communicating with a specific server.
oproxy.portal.port=7777

# Hostname and port that clients use to access the third-party HTTP listener.
# This value will be passed as the Host: HTTP header.
oproxy.portal.alias=portal.mycompany.com

# Description of URL(s) that will be redirected to this server.
oproxy.portal.urlrule=/*
```

**9.2.1.3.3 Configuring the IIS Listener to Use the Oracle Application Server Proxy Plug-in** This section provides proxy plug-in configuration instructions for the IIS listener on Windows systems. The process involves creating Windows registry entries and using the IIS management console to add directories and filters. You must restart the listener after configuring the plug-in. To configure the plug-in, perform the following steps:

1. Disable the Oracle HTTP Server in the Oracle Application Server instance by performing these steps:
  - a. Navigate to the page for the Oracle Application Server instance in Oracle Enterprise Manager 10g Application Server Control Console.
  - b. Select the Oracle HTTP Server in the System Components list.
  - c. Click **Enable/Disable Components**.
2. Select **Start > Run**.
3. In the **Run** dialog box, type **regedit**.  
The **Registry Editor** window opens.
4. Expand the **HKEY\_LOCAL\_MACHINE** folder (click the + preceding its name).
5. Expand the **SOFTWARE** folder (click the + preceding its name).
6. Click the **ORACLE** folder.
7. Select **Edit > New > Key**.  
A new folder is added under the **ORACLE** folder with the name **New Key #1**.
8. Type **IIS Proxy Adapter** for the key name.
9. Select **Edit > New > String Value**.  
A new value is added in the right window pane with the name **New Value #1**.
10. Type `server_defs` for the value name.
11. Select **Edit > Modify**.  
The **Edit String** dialog box appears.

12. In the **Value** field, type the full path of the proxy server definition file and click **OK**.
13. (Optional) Specify a log file and logging level using the **Edit > New > String Value** procedure in steps 8-11.
  - a. Add a string value with the name `log_file` and the desired location of the log file (for example, `d:\proxy\proxy.log`)
  - b. Add a string value with the name `log_level` and a value for the desired log level. Valid values are `debug`, `inform`, `error` and `emerg`.
14. Using the IIS management console, add a new virtual directory to the IIS Web site with the same physical path as that of `oracle_proxy.dll`. Name the directory `oproxy` and give it execute access.
15. Using the IIS management console, right-click **IIS Web Site**, then choose **Properties** from the menu. Click the **ISAPI Filter Tab** and add `oracle_proxy.dll` as a filter in the IIS Web site. The name of the filter should be `oproxy` and its executable must point to the directory containing `oracle_proxy.dll` (for example, `d:\proxy\oracle_proxy.dll`).
16. Stop, and then start the IIS Server, ensuring that the `oproxy` filter is marked with a green upward arrow.

For usage notes and troubleshooting information about the Oracle Application Server Proxy Plug-In, see Appendix A of the *Oracle HTTP Server Administrator's Guide*.

## 9.2.2 Testing the OracleAS Single Sign-On Connection

At this point, you can test the connection to OracleAS Single Sign-On by accessing the following URL:

**`http://login.mycompany.com:7777/pls/orasso`**

The **Access Partner Applications** page appears. However, the login link still points to `idmhost1.mycompany.com`.

## 9.2.3 Configuring OracleAS Single Sign-On to Use a Reverse Proxy

The procedure for configuring OracleAS Single Sign-On to use a reverse proxy server comprises these tasks:

1. [Ensuring that IP Checking is Off](#)
2. [Executing the `ssocfg` Script](#)
3. [Updating the `targets.xml` File](#)
4. [Updating the `httpd.conf` File](#)
5. [Registering `mod\_osso` to Use the Proxy Host Name](#)
6. [Updating the Single Sign-On Configuration](#)

### 9.2.3.1 Ensuring that IP Checking is Off

1. Access the following URL:

**`http://idmhost.mycompany.com:7777/pls/orasso`**

The **Access Partner Applications** page appears.

2. Click **Login**.

The **Login** page appears.

3. Enter the administrator's user name and password and click **Login**.

The **Home** page appears.

4. Click **SSO Server Administration**.

The **SSO Server Administration** page appears.

5. Click **Edit SSO Server Configuration**.

The **Edit SSO Server** page appears.

6. Ensure that the **Verify IP addresses for requests made to the SSO server** box is deselected.

### 9.2.3.2 Executing the ssocfg Script

Issue this command in *ORACLE\_HOME/sso/bin*:

**ssocfg.sh https proxyName proxyHttpsPort** (UNIX)

**ssocfg.bat https proxyName proxyHTTPsPort** (Windows)

### 9.2.3.3 Updating the targets.xml File

1. Open the *ORACLE\_HOME/sysman/emd/targets.xml* file and locate the target type *oracle\_sso\_server*.
2. Update the *HTTPMachine* and *HTTPPort* attributes with the proxy server host and port attributes that were passed to *ssocfg*. For example:

```
<Property NAME="HTTPMachine" VALUE="proxyName" />
<Property NAME="HTTPPort" VALUE="proxyHttpsPort" />
<Property NAME="HTTPProtocol" VALUE="https" />
```

3. Save and close the file.
4. Reload the Application Server Control Console by issuing this command:

**ORACLE\_HOME/bin/emctl reload**

### 9.2.3.4 Updating the httpd.conf File

1. Access the Oracle Enterprise Manager 10g Application Server Control Console.
2. Click the link for the instance to configure.
3. Click the **HTTP Server** link.
4. Click the **Administration** link.
5. Click **Advanced Server Properties**.
6. Select **httpd.conf**.

An editor window opens for the *httpd.conf* file.

7. Change the *ServerName* and *Port* directive values to the proxy server host and port values that were passed to *ssocfg*. For example:

```
KeepAlive off
ServerName proxyName
Port proxyPort
```

8. Navigate to the end of the file.

9. Create a virtual host container. This causes the single sign-on login module in OC4J\_SECURITY to be invoked when a user logs into the proxy server.

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

# Windows
LoadModule certheaders_module modules/ApacheModuleCertHeaders.dll

NameVirtualHost idmhost1.mycompany.com:7777

<VirtualHost idmhost1.mycompany.com:7777>
    ServerName login.mycompany.com
    Port 443
    RewriteEngine On
    RewriteOptions inherit
    SimulateHttps On
</VirtualHost>
```

### 9.2.3.5 Updating Oracle Internet Directory with the Operation URL

You must modify the Delegated Administration Services URL in Oracle Internet Directory. To do this, follow these steps on IDMHOST1:

1. Use a text editor to create a file named `setdasurl.ldif` with these contents:
 

```
dn:cn=OperationURLs,cn=DAS,cn=Products,cn=OracleContext
changetype: modify
replace: orcldasurlbase
orcldasurlbase: https://login.mycompany.com/
```
2. Set the ORACLE\_HOME environment variable to specify the Oracle home in which ldapmodify is available.
3. Issue this command:
 

```
ldapmodify -D "cn=orcladmin" -w password -v -f setdasurl.ldif
```
4. Access OracleAS Portal at this URL:
 

```
http://APPHOST1.mycompany.com:7777/pls/portal
```
5. Log in as the portal user and provide the password defined during installation.
6. Click **Administer**, then **Global Settings**, then **SSO/OID**.
7. Check the **Refresh Cache for OID Parameters** box.
8. Click **Apply**.
9. Click **OK**.

### 9.2.3.6 Registering mod\_osso to Use the Proxy Host Name

1. Set the ORACLE\_HOME environment variable to the directory in which Oracle Application Server is installed, using the applicable command, substituting *oraHome* with the Oracle home directory:

UNIX (csh):

```
setenv ORACLE_HOME oraHome
```

UNIX (Bourne and ksh):

```
ORACLE_HOME=oraHome; export ORACLE_HOME
```

Windows:

```
set ORACLE_HOME oraHome
```

2. Execute the `ssoreg` script by issuing one of the following commands in `ORACLE_HOME/sso/bin`. The command and parameters are shown on separate lines for readability; optional parameters are bracketed. Descriptions of all parameters are given in [Table 9–3](#).

UNIX:

```
ssoreg.sh
```

```
-oracle_home_path oracleHome
```

```
-site_name siteName
```

```
-config_mod_osso TRUE
```

```
-mod_osso_url https://login.mycompany.com
```

```
[-virtualhost]
```

```
[-update_mode CREATE | DELETE | MODIFY]
```

```
[-config_file configFilePath]
```

```
[-admin_info adminInfo]
```

```
[-admin_id adminId]
```

Windows:

```
ssoreg.bat
```

(Parameters are the same as for UNIX.)

3. Restart the Oracle HTTP Server by issuing this command:

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

**Table 9–3 ssoreg Parameters**

Parameter	Description
<code>oracle_home_path</code>	Absolute path to the Oracle home.
<code>site_name</code>	The host name and port of the partner application.
<code>config_mod_osso</code>	Indicates that <code>mod_osso</code> is the application being registered. This parameter must be included in order for the <code>osso.conf</code> file to be generated.
<code>mod_osso_url</code>	The URL used to access the partner application. This parameter must be specified in the format <code>http://host.domain</code> . For example: <code>https://www.mycompany.com</code>
<code>virtualhost</code>	(Optional) Indicates that a virtual host is being registered with the single sign-on server.
<code>update_mode</code>	(Optional) Creates, deletes, or modifies the partner registration record.
<code>config_file</code>	(Optional) Location of the <code>osso.conf</code> file for the virtual host, if one is being configured. For example: <code>ORACLE_HOME/Apache/Apache/conf/osso/virtual_host_name/osso.conf</code> .
<code>admin_info</code>	(Optional) User name of the <code>mod_osso</code> administrator.
<code>admin_id</code>	(Optional) E-mail address of the <code>mod_osso</code> administrator.

4. Log in to the OracleAS Single Sign-On Administration page as the Administrator, and use the **Administer Partner Applications** page to delete the entry for the partner application **apphost1.mycompany.com**.

### 9.2.3.7 Updating the Single Sign-On Configuration

1. Update the DCM metadata repository with the changes to the local files by issuing this command in *ORACLE\_HOME/dcm/bin*:

```
dcmctl upateconfig
```

2. Restart the single sign-on middle tier by issuing these commands in *ORACLE\_HOME/bin*:

```
opmnctl restartproc process-type=HTTP_Server
```

```
opmnctl restartproc process-type=OC4J_SECURITY
```

## 9.2.4 Validating the OracleAS Single Sign-On Configuration

Follow these steps to ensure that the OracleAS Single Sign-On configuration is functioning as it should.

1. Access the following URL:

```
https://login.mycompany.com/pls/orasso
```

The **Access Partner Applications** page appears. The login link now points to [login.mycompany.com](https://login.mycompany.com).

2. Click **Login**.

The **Login** page appears.

3. Enter the administrator's user name and password and click **Login**.

If the **Home** page appears (the login is successful), the proxy is configured correctly for OracleAS Single Sign-On.

## 9.2.5 Testing the OracleAS Portal Connection

At this point, you can test the connection to OracleAS Portal.

1. Access the following URL:

```
http://portal.mycompany.com/pls/portal
```

The **OracleAS Portal** page appears. However, the login link still points to [idmhost1.mycompany.com](https://idmhost1.mycompany.com). The OracleAS Portal welcome page URL points to [apphost1.mycompany.com](https://apphost1.mycompany.com). After you complete all of the steps in [Section 9.2.6, "Configuring OracleAS Portal for a Reverse Proxy"](#), it will point to [portal.mycompany.com](https://portal.mycompany.com).

## 9.2.6 Configuring OracleAS Portal for a Reverse Proxy

After installing and configuring the reverse proxy server and configuring OracleAS Single Sign-On to use it, configure OracleAS Portal as described in this section.

The procedure for configuring OracleAS Portal to use a proxy server comprises these tasks:

1. [Ensuring Validity of Self-Referential URLs Rendered on OracleAS Portal Pages](#)
2. [Configuring Loopback Communication to the Internal Server](#)
3. [Specifying the OracleAS Portal Published Address and Protocol](#)
4. [Configuring Seeded Providers and Locally Hosted Web Providers](#)
5. [Registering the Domain Name](#)
6. [Augmenting the Parallel Page Engine x509certfile for Web Providers Exposed Over SSL \(Optional\)](#)
7. [Registering Web Providers or Provider Groups Exposed over SSL \(Optional\)](#)
8. [Enabling the Federated Portal Adapter for SSL \(Optional\)](#)
9. [Registering OracleAS Portal as an Oracle Ultra Search Content Source \(Optional\)](#)

### 9.2.6.1 Ensuring Validity of Self-Referential URLs Rendered on OracleAS Portal Pages

Follow these steps to define a virtual host on the middle tier for the proxy server.

1. Use the Oracle Enterprise Manager 10g Application Server Control Console to access the OracleAS Portal middle tier page on APPHOST1. Start a browser and access **http://hostname:1810** and click the link for the application server instance hosting OracleAS Portal.
2. Click the **HTTP Server** link.
3. Click the **Administration** link.
4. Click **Advanced Server Properties**.
5. Select **httpd.conf**.

An editor window opens for the `httpd.conf` file.

6. Navigate to the end of the file.
7. Create a virtual host container for the proxy server, as shown in the following example:

```
# UNIX
LoadModule certheaders_module libexec/mod_certheaders.so
# Windows
LoadModule certheaders_module modules/ApacheModuleCertHeaders.dll

NameVirtualHost apphost1.mycompany.com:7778

<VirtualHost apphost1.mycompany.com:7778>
    ServerName portal.mycompany.com
    Port 443
    RewriteEngine On
    RewriteOptions inherit
    SimulateHttps On
</VirtualHost>
```



```
<VirtualHost apphost1.mycompany.com:7778>
  ServerName apphost1.mycompany.com
  Port 7777
  RewriteEngine On
  RewriteOptions inherit
</VirtualHost>
```

8. Click **Apply**.
9. Restart the Oracle HTTP Server when prompted.

### 9.2.6.2 Configuring Loopback Communication to the Internal Server

In order for the computer that hosts login.mycompany.com to be able to resolve the proxy name within the firewall, you must create entries in the hosts file on that computer. The hosts file is stored on the computer's filesystem, and is used to look up the Internet Protocol address of a device connected to a computer network. The hosts file describes a many-to-one mapping of device names to IP addresses. To access a device by name, the networking system tries locate the name within the hosts file, if it exists. This is usually the first method used to locate the address of a system, since the hosts file is stored on the computer itself and does not require network access. A DNS lookup requires access to an external system, which is typically slower.

The hosts file is usually named `hosts`. Its location varies according to operating system, as follows:

UNIX and like operating systems: `/etc`

Windows 95/Windows 98/Windows Me: `%windir%\`

Windows 2000/Windows XP: `%SystemRoot%\system32\drivers\etc\`

Windows NT: `C:\winnt\system32\drivers\etc\`

Entries in the host file resemble the following:

```
# Sample HOSTS file
127.0.0.1 localhost
123.45.67.8 www.proxyName.com
```

If you do not provide the entries in the hosts file as specified in the sample hosts file example, then you need to configure the Oracle Application Server computer to recognize a proxy server that would take the request out to the Internet and back in through the reverse proxy server (`www.proxyName.com`), or configure the reverse proxy server's internal interface to respond to `www.proxyName.com`.

### 9.2.6.3 Specifying the OracleAS Portal Published Address and Protocol

The host name and port number by which OracleAS Portal is addressed is usually the OracleAS Web Cache host and port (since OracleAS Web Cache is usually configured as the first listener). However, in a configuration with a reverse proxy server in front of OracleAS Web Cache, the published host name and port will be that of the reverse proxy server.

In this configuration, the OracleAS Web Cache invalidation messages must be sent directly to the OracleAS Web Cache computer, as opposed to the reverse proxy server. When the published host name is different from the host name used for OracleAS Web Cache invalidation, you can use the Portal Dependency Settings file, `iasconfig.xml`, to specify these settings.

The `iasconfig.xml` file must be updated to provide the correct farm name, host name, and port information to access OracleAS Portal through the reverse proxy

server, and to perform the OracleAS Web Cache invalidation. The invalidation password used is always the invalidation password for the OracleAS Web Cache that is caching the OracleAS Portal content (not the OracleAS Web Cache that is acting as a reverse proxy).

Follow these steps to configure these settings:

1. Edit the `ORACLE_HOME/portal/conf/iasconfig.xml` file to include the entries shown in bold in [Example 9-5](#).

**Example 9-5 `iasconfig.xml` File Edited to Include Farm Element**

```
<IASConfig XSDVersion="1.0">

  <IASFarm Name="Farm-1.portal.mycompany.com" Host="portal.mycompany.com">
    <WebCacheComponent ListenPort="443" InvalidationPort="9401"
      InvalidationUsername="invalidator" InvalidationPassword="welcome1"
      SSLEnabled="true"/>
  </IASFarm>

  <IASInstance Name="ias-1.oid.mycompany.com" Host="oid.mycompany.com">
    <OIDComponent AdminPassword="@BV5s2KPJEWc5a0l4n8lbTxUY="
      PortSSLEnabled="false" LDAPPort="389" AdminDN="cn=orcladmin"/>
  </IASInstance>

  <IASInstance Name="ias-1.apphost1.mycompany.com" Host="apphost1.mycompany.com">
    <EMComponent ConsoleHTTTPort="1810" SSLEnabled="false"/>
  </IASInstance>

  <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
    SchemaPassword="@Beyh8p2bOWELQC5a5zRtuYc="
    ConnectString="cn=iasdb,cn=oraclecontext">
    <WebCacheDependency ContainerType="IASFarm"
      Name="Farm-1.portal.mycompany.com"/>
    <OIDDependency ContainerType="IASInstance" Name="ias-1.oid.mycompany.com"/>
    <EMDependency ContainerType="IASInstance" Name="ias-1.apphost1.mycompany.com"/>
  </PortalInstance>
</IASConfig>
```

2. Encrypt any plain text passwords in `iasconfig.xml` by issuing this command in `ORACLE_HOME/portal/conf`:
 

```
ptlconfig -encrypt
```
3. Update the `WebCacheDependency` element in the `iasconfig.xml` file to include the `InvalidationHost` property as shown in this example:
 

```
<WebCacheDependency ContainerType="IASFarm" Name=" Farm-1.portal.mycompany.com
  " InvalidationHost="internal.mycompany.com" />
```
4. Execute the `ptlconfig` script by issuing this command in `ORACLE_HOME/portal/conf`:
 

```
ptlconfig -dad portal -site -wc -em
```

### 9.2.6.4 Configuring the Parallel Page Engine Loop-Back with the Load Balancing Router on APPHOST1

In this step, you enable (non-SSL) loop-back communication between the proxy server and the Parallel Page Engine on APPHOST1.

Follow these steps to create the loop-back configuration:

1. Open the `APPHOST1_ORACLE_HOME/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF/web.xml` file.
2. Locate the Page servlet section.
3. Add the lines shown in bold:

```
<servlet>
<servlet-name>page</servlet-name>
  <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
    <init-param>
      <param-name>useScheme</param-name>
      <param-value>http</param-value>
    </init-param>
    <init-param>
      <param-name>usePort</param-name>
      <param-value>7777</param-value>
    </init-param>
    <init-param>
      <param-name>httpsports</param-name>
      <param-value>443</param-value>
    </init-param>
  </servlet>
```

4. Save the `web.xml` file.
5. Save the manual configuration changes in the Distributed Configuration Management repository by issuing the following command on APPHOST1 in `ORACLE_HOME/dcm/bin`:

```
dcmctl updateconfig
```

6. Restart all components on APPHOST1 by issuing the following command in `ORACLE_HOME/opmn/bin`:

```
opmnctl stopall
```

```
opmnctl startall
```

### 9.2.6.5 Configuring OracleAS Web Cache with the Reverse Proxy Server on APPHOST1

You must configure a site definition, site alias, and a site-to-server mapping to make OracleAS Web Cache function correctly with the reverse proxy server. Use the Web Cache Manager, the graphical user interface provided for editing the configuration stored in the `webcache.xml` file.

1. Access the Web Cache Administrator at:

```
http://apphost1.mycompany.com:9400/webcacheadmin
```

The Web Cache Administrator password dialog appears.

2. Enter the OracleAS Web Cache administrator password.

---

**Note:** At installation time, The OracleAS Web Cache administrator password is set to the same password as the `ias_admin` password. The OracleAS Web Cache administrator password must be identical for all cache cluster members.

---

The **Web Cache Cache Operations** page appears. A scrollable frame on the left side of the window contains groups of configuration elements. To access an element, click its link. The content area of the page is then populated with the values for that element.

3. Click the **Site Definitions** link in the **Origin Servers, Sites and Load Balancing** section.

The **Site Definitions** window opens.

4. Click **Add Site**.
5. Enter the following information (leave other fields blank):
  - Host name: **portal.mycompany.com**
  - Port: **443**
  - Client-side Certificate: **Not required**
  - Default Site: **Yes**
  - Create Alias from Site Name with/without www: **No**
6. Click **Submit**.
7. Select the radio button for the site for which the alias will be added (portal.mycompany.com).
8. Click **Add Alias**.

The **Add Alias for Site** window opens.

9. Enter **portal.mycompany.com** for the host name and **7777** for the port. (7777 is the value for the `usePort` parameter in the `web.xml` file in the Parallel Page Engine configuration.)
10. Click **Submit**.

The alias is added. An alias is needed in the configuration because Portal sends invalidation messages with the value of the `HOST` attribute in the invalidation message the same as the site name (in this case, `portal.mycompany.com:443`), but OracleAS Web Cache caches the portal content keyed on a host:port combination such as `portal.mycompany.com:7777`; thus, the invalidation is not executed. Therefore, it is necessary to define an alias, so that OracleAS Web Cache manages the content caching recognizing `portal.mycompany.com:443` and `portal.mycompany.com:7777` as one and the same, and thereby correctly invalidating OracleAS Portal content, although the content is keyed on a different host:port combination than the site name.

11. Click **Add Alias**.

A window with host name and port fields opens.

12. Enter **portal.mycompany.com** for the host name and **80** for the port.
13. Click **Submit**.

The alias is added.

---

**Note:** An alias for port 80 is needed because the HOST header sent by the browser will be portal.mycompany.com (without a port number appended to it). Since OracleAS Web Cache is listening on the HTTP port, it will assume that the port number is 80 and use this to determine the site-to-server mapping, and for any cache key creation.

---

14. Click **Apply Changes**.
15. Click the **Site-to-Server Mapping** link in the **Origin Servers, Sites, and Load Balancing** section.  
The **Site-to-Server Mapping** page appears, in which you map the site and site alias to an origin server.
16. Select the first mapping in the table and click **Insert Above**.  
The **Edit/Add Site-to-Server Mapping** page appears.
17. Select the **Select From Site Definitions** option.
18. Select **portal.mycompany.com**.
19. Select **apphost1.mycompany.com** in the **Select Application Web Servers** section.
20. Click **Submit**.
21. Remove unused mappings or entries containing the wild card character \*.
22. Click **Apply Changes**.
23. Click **Restart**.

#### 9.2.6.6 Configuring Seeded Providers and Locally Hosted Web Providers

1. Edit the `ORACLE_HOME/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/web.xml` file to include these parameters:
 

```
<context-param>
  <param-name>useScheme</param-name>
  <param-value>http</param-value>
</context-param>
<context-param>
  <param-name>usePort</param-name>
  <param-value>7777</param-value>
</context-param>
```
2. Issue this command in `ORACLE_HOME/opmn/bin`:
 

```
opmnctl restartproc ias-component=OC4J process-type=OC4J_Portal
```
3. Log in to OracleAS Portal as the administrator (for example, PORTAL).
4. Click **Administer**.
5. Click **Portlets**.
6. In the **Remote Providers** portlet, **Name** field, enter **WEBCLIPPING**.
7. Click **Edit**.
8. Click **Connection**.

9. In the URL field, change the protocol to **http** and the port in the URL to **7777**.
10. Click **Apply**.
11. Click **OK**.
12. Repeat steps 4 through 11, substituting **OMNIPORTLET** for **WEBCLIPPING**.
13. Add the proxy server to the hosts file.

When you register locally hosted Web providers (that is, Web providers operating on the same computer as OracleAS Portal), such as the JPDK sample provider, you must register them with the HTTP protocol, *www.proxyName.com* as the host name, and **7777** as the port number.

### 9.2.6.7 Registering the Domain Name

You must register the proxy server host name on a domain name server on the Internet.

### 9.2.6.8 Re-registering mod\_osso on APPHOST1

1. Set the `ORACLE_HOME` environment variable to the current Oracle home.
2. Edit the SSO registration script `ORACLE_HOME/sso/bin/ssoreg` as shown in [Example 9-6](#), and then execute it. [Example 9-6](#) shows the usage of `ssoreg.sh` on UNIX; on Windows, the script name is `ssoreg.bat`.

---

**Note:** The script shown in [Example 9-6](#) has multiple lines for readability only. When you execute the script, all parameters are on a single continuous line.

---

#### **Example 9-6 ssoreg Usage on UNIX**

```
ORACLE_HOME/sso/bin/ssoreg.sh
-site_name portal.mycompany.com
-mod_osso_url https://portal.mycompany.com
-config_mod_osso TRUE
-oracle_home_path ORACLE_HOME
-config_file ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
-admin_info cn=orcladmin
-virtualhost
```

A partner application, **portal.mycompany.com**, is created.

3. Restart the Oracle HTTP Server by issuing this command:  
**ORACLE\_HOME/opmn/bin/opmnctl restartproc process-type=HTTP\_Server**
4. Access the following URL:  
**https://login.mycompany.com/pls/orasso**
5. Log in to the OracleAS Single Sign-On Administration page as the Administrator, and use the **Administer Partner Applications** page to delete the entry for the partner application **apphost1.mycompany.com**.

### 9.2.6.9 Augmenting the Parallel Page Engine x509certfile for Web Providers Exposed Over SSL (Optional)

If you use the IIS listener and registering Web provider exposed over HTTPS, you must augment the provider certificate to the x509certfile defined in the Parallel Page Engine's *ORACLE\_HOME/j2ee/OC4J\_Portal/applications/portal/portal/WEB-INF/web.xml* file. Add the x509certfile parameter as shown:

```
<servlet-name>page</servlet-name>
<servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
<init-param>
  <param-name>x509certfile</param-name>
  <param-value>C:\mySSLconfig\trustedCerts.txt</param-value>
</init-param>
```

### 9.2.6.10 Registering Web Providers or Provider Groups Exposed over SSL (Optional)

See [Section 7.3.10](#) for instructions.

### 9.2.6.11 Enabling the Federated Portal Adapter for SSL (Optional)

See [Section 7.3.11](#) for instructions.

### 9.2.6.12 Registering OracleAS Portal as an Oracle Ultra Search Content Source (Optional)

See [Section 7.3.12](#) for instructions.

### 9.2.6.13 Using Oracle HTTP Server 1.3 as a Reverse Proxy for OracleAS Portal

When you use Oracle HTTP Server version 1.3 as a reverse proxy for OracleAS Portal, the configuration steps in this section are required in order to resolve OracleAS Portal OracleAS Web Cache invalidation.

On the OracleAS Portal middle tier:

1. Perform the steps specified in [Section 9.2.1.2.3, "Using the X-FORWARDED-HOST Value with Apache v. 1.3 mod\\_proxy"](#) on page 9-35.
2. Access the Web Cache Administrator at:  
**http://apphost1.mycompany.com:9400/webcacheadmin**  
The Web Cache Administrator password dialog appears.
3. For the user name, enter *ias\_admin* or *administrator*, and enter the OracleAS Web Cache administrator password.

---

**Note:** At installation time, The OracleAS Web Cache administrator password is set to the same password as the *ias\_admin* password. The OracleAS Web Cache administrator password must be identical for all cache cluster members.

---

4. The **Web Cache Manager** page appears. A scrollable frame on the left side of the window contains groups of configuration elements. To access an element, click its link. The content area of the page is then populated with the values for that element.

5. Click the **Site Definition** link in the **Origin Servers, Sites, and Load Balancing** section.

The **Site Definition** page appears.

6. Select the radio button for the site for which the alias will be added (**apphost1.mycompany.com**).

The **Add Alias for Site** window opens.

7. Enter **portal.mycompany.com** for the host name and **443** for the port.
8. Click **Submit**.
9. Click **Apply Changes**.
10. Click **Restart**.

### 9.2.7 Validating the OracleAS Portal Configuration

Perform this step to ensure that the OracleAS Portal configuration is functioning as it should.

1. Access the following URL:

**`http://myportal.mycompany.com/pls/portal`**

The **OracleAS Portal** page appears. The login link now points to `login.mycompany.com`.

## 9.3 Configuring J2EE and Web Cache on the Web Tier

This section explains how to install and configure myJ2EE using a J2EE and Web Cache installation on the Web Server Tier, instead of a standalone Oracle HTTP Server. Advantages to this configuration include the ability to use Oracle Enterprise Manager 10g Application Server Control Console to manage the Oracle HTTP Server, and the ability to manage all of the Oracle Application Server instances in the configuration as part of an Oracle Application Server File-Based Farm.

### 9.3.1 Installing and Configuring the Security Infrastructure

Follow the instructions in [Section 6.1, "Installing and Configuring the Security Infrastructure"](#) on page 6-1.

### 9.3.2 Installing and Configuring the Application Tier

The application tier consists of multiple computers hosting middle tier Oracle Application Server instances in an Oracle Application Server File-Based Farm. Each instance contains multiple Oracle Application Server Containers for J2EE instances, hosting deployed applications. In the complete configuration, requests are balanced among the OC4J instances on the application tier computers to create a performant and fault tolerant application environment. [Figure 2-1, "Enterprise Deployment Architecture for myJ2EECompany.com"](#) on page 2-4, shows the application tier (APPHOST1 and APPHOST2).



### 9.3.2.1 A Note About Port Assignments for the Oracle Application Server File-Based Farm

Before you begin installing and configuring the OracleAS File-Based Farm for myJ2EECompany, you should understand the implications of the default port assignments for Distributed Configuration Management, in the case of environments that require inter-instance communication across a firewall.

The Oracle Universal Installer assigns the ports described in [Table 9–4](#) by default when the instance is installed.

**Table 9–4 Oracle Universal Installer Default Port Assignments**

Quantity	Purpose/Description
1	DCM Discovery Port. The first instance installed on a computer is assigned port 7100 for this; the second instance installed on a computer is assigned 7101, and so on. This is defined in the <code>ORACLE_HOME/dcm/config/dcmCache.xml</code> file, in the <code>discoverer</code> element (for example, <code>&lt;discoverer discovery-port = "7100" original-"true" xmlns=" " /&gt;</code>
50	<p>Range of ports for inter-instance communication: 7120 to 7179. These are defined in the <code>ORACLE_HOME/dcm/config/dcmCache.xml</code> file, in the <code>port</code> element (for example, <code>&lt;port lower="7120" upper="7179"&gt;</code>.)</p> <p>After installation, you will probably want to limit the number of ports open on the firewall. The actual port needs for inter-instance communication are:</p> <ul style="list-style-type: none"> <li>■ 1 for the Oracle Enterprise Manager 10g Application Server Control Console on each instance</li> <li>■ 1 for the DCM daemon on each instance</li> <li>■ 1 for each <code>dcmctl</code> client operating on each instance</li> </ul>

If the ports in the range 7100 to 7179 were open on the firewall before installation, the instances in the farm will be able to communicate immediately after installation. Note that:

- If you want the port assignments to be of a different numeric range from these, then, before installation, you must assign a DCM Discovery Port using the `staticports.ini` file, and select the **Manual** option during installation. (See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-2 for more information.) The range of ports will then be assigned accordingly, as specified in [Table 9–4](#).
- After installation of all instances, you should configure the firewall to close the unused ports within the assigned range on each instance.

The procedure for installing and configuring the Application Tier is the same as that for the primary configuration of myJ2EE. Perform all steps in these sections:

- [Section 6.3.1, "Installing the First Application Tier Application Server Instance on APPHOST1"](#) on page 6-2
- [Section 6.3.2, "Installing the Second Application Tier Application Server Instance on APPHOST2"](#) on page 6-6
- [Section 6.3.3, "Creating OC4J Instances on the Application Tier"](#) on page 6-10
- [Section 6.3.4, "Deploying J2EE Applications"](#) on page 6-11

- [Section 6.3.5, "Creating a DCM-Managed Oracle Application Server Cluster on the Application Tier"](#) on page 6-12
- [Section 6.3.5.1, "Creating the DCM-Managed OracleAS Cluster"](#)

### 9.3.3 Installing and Configuring the Web Tier

The Web Tier consists of multiple middle tier Oracle Application Server J2EE and Web Cache instances, with only Oracle HTTP Server configured. The Oracle HTTP Servers which route the requests to the OC4J instances on the application tier computers.

#### 9.3.3.1 Installing the Web Tier Application Servers on WEBHOST1 and WEBHOST2

Follow these steps to install an Oracle Application Server middle tier on WEBHOST1 and WEBHOST2:

1. Ensure that the system, patch, kernel and other requirements are met as specified in the *Oracle Application Server Installation Guide*. You can find this guide in the Oracle Application Server platform documentation library for the platform and version you are using.
2. Copy the `staticports.ini` file from the `Disk1/stage/Response` directory to a local directory, such as `TMP`. You will provide the path to this file during installation.
3. Edit the `staticport.ini` file to assign the following custom ports:

```
Oracle HTTP Server port = 7777
Oracle HTTP Server Listen port = 7778
Application Server Control port = 1810
```

---

**Notes:** Ensure that these ports are not already in use by any other service on the computer. Using the Static Ports feature to install the Application Server Tier ensures that the port assignments will be consistent, if the ports are correctly specified in the file and the port is not already in use. If a port is incorrectly specified, the Oracle Universal Installer will assign the default port. If a port is already in use, the Oracle Universal Installer will select the next available port.

See [Section B.3, "Using the Static Ports Feature with Oracle Universal Installer"](#) on page B-2 for more information.

---

4. Start the Oracle Universal Installer as follows:  
On UNIX, issue this command: **runInstaller**  
On Windows, double-click **setup.exe**  
The **Welcome** screen appears.
5. Click **Next**.  
On UNIX systems, the **Specify Inventory Directory and Credentials** screen appears.
6. Specify the directory you want to be the `oraInventory` directory and the operating system group that has write permission to it.
7. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `oraInstRoot.sh` script.

8. Open a window and run the script, following the prompts in the window.
9. Return to the Oracle Universal Installer screen and click **Next**.

The **Specify File Locations** screen appears with default locations for:

- The product files for installation (Source)
- The name and path to the Oracle home (Destination)

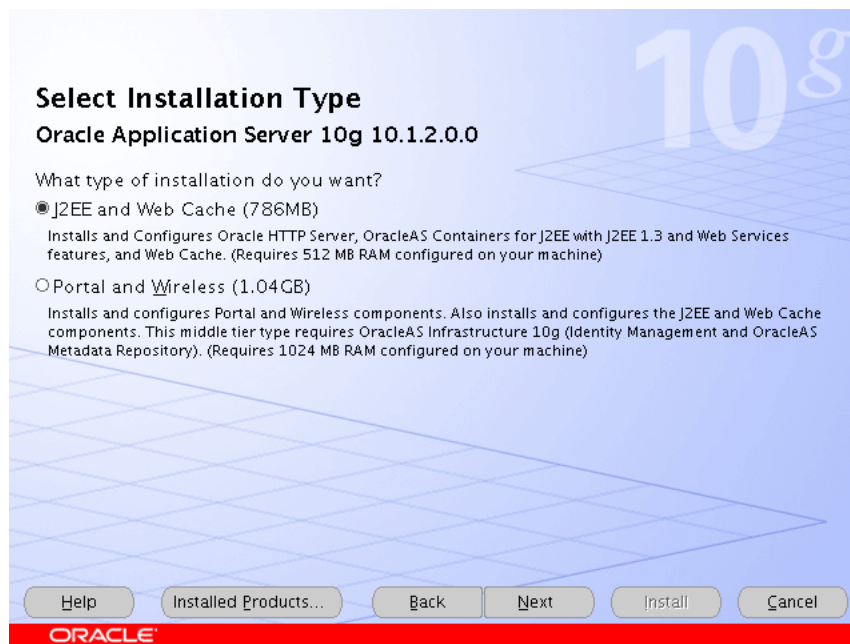
10. Click **Next**.

The **Select a Product to Install** screen appears.

**Figure 9–3 Oracle Universal Installer Select a Product to Install Screen**



11. Select **Oracle Application Server 10g**, as shown in [Figure 9–3](#), and click **Next**.  
The **Select Installation Type** screen appears.

**Figure 9–4 Oracle Universal Installer Select Installation Type Screen**

12. Select **J2EE and Web Cache**, as shown in Figure 9–4, and click **Next**.

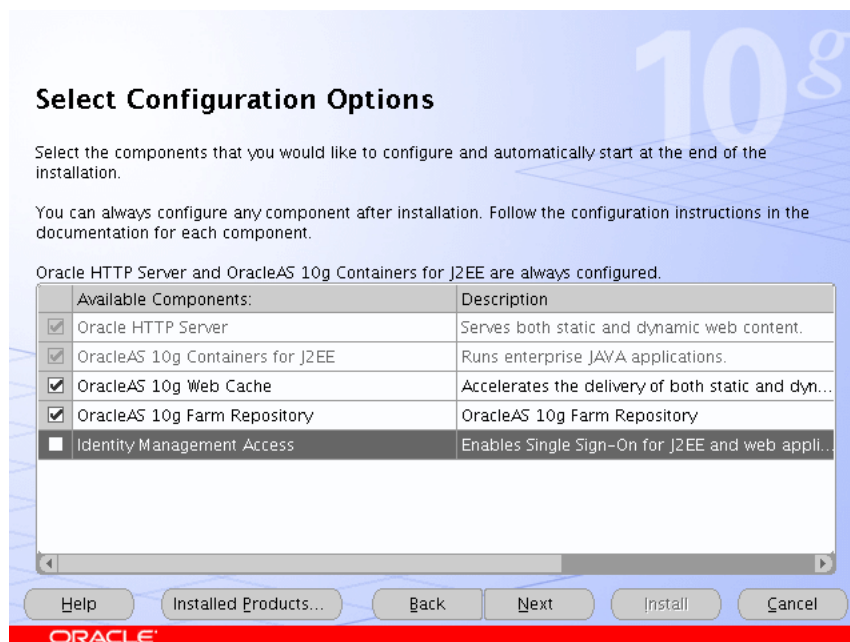
The **Product-Specific Prerequisite Checks** screen appears.

13. Click **Next**.

The **Confirm Pre-Installation Requirements** screen appears.

14. Ensure that the requirements are met and click **Next**.

The **Select Configuration Options** screen appears.

**Figure 9–5 Oracle Universal Installer Select Configuration Options Screen**

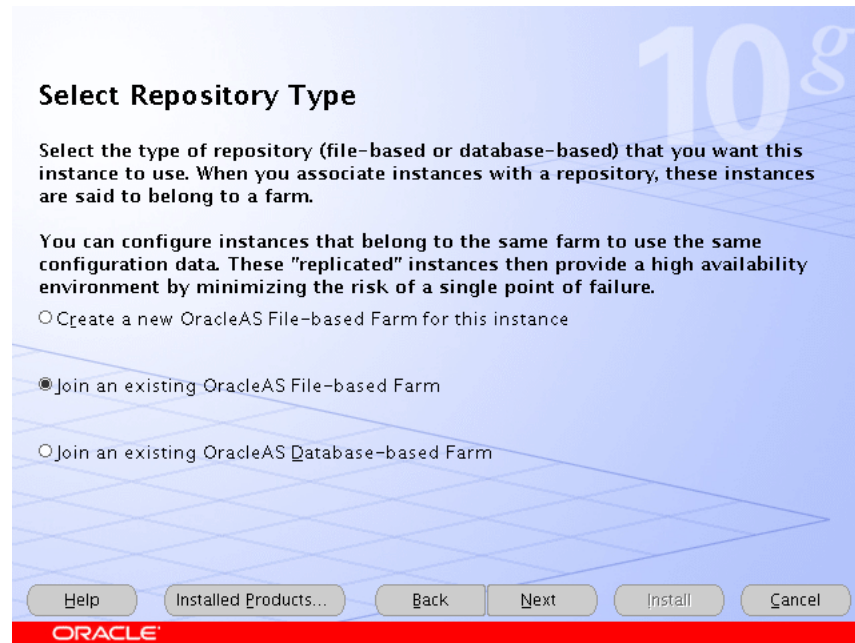
15. Select **OracleAS Web Cache and OracleAS 10g Farm Repository**, as shown in [Figure 9-5](#), and click **Next**.

The **Specify Port Configuration Options** screen appears.

16. Select **Manual**, specify the location of the `staticports.ini` file, and click **Next**.

The **Select Repository Type** screen appears.

**Figure 9-6 Oracle Universal Installer Select Repository Type Screen**



17. Select **Join an existing OracleAS File-based Farm**, as shown in [Figure 9-6](#), and click **Next**.

The **Specify File-based Farm Repository** screen appears.

18. Specify the host name of APPHOST1, and the DCM Discovery Port on which the OracleAS File-based Farm Repository listens, and click **Next**.

---

**Note:** The port range 7100-7179 is used for communication between DCM instances. The first installed instance of an OracleAS File-Based Farm on a computer has port 7100 assigned as its DCM Discovery Port. A subsequently installed instance will use port 7101, and so on. See [Section 9.3.2.1, "A Note About Port Assignments for the Oracle Application Server File-Based Farm"](#) on page 9-53 for more information.

---

The **Specify Instance Name and ias\_admin Password** screen appears.

19. Specify an instance name and the Oracle Application Server administrator's password and click **Next**.

The **Summary** screen appears.

20. Click **Next**.

On UNIX systems, a dialog appears, prompting you to run the `root.sh` script.

21. Open a window and run the script, following the prompts in the window.
22. Return to the Oracle Universal Installer screen and click **Next**.  
The **Configuration Assistants** screen appears. Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, the **End of Installation** screen appears.
23. Click **Exit**, and then confirm your choice to exit.
24. Verify that the installation was successful by viewing the application server instance in Oracle Enterprise Manager 10g. Start a browser and access:  
**`http://hostname:1810`**

---

## Sample Configurations for Load Balancers

This appendix provides sample configurations for commonly used load balancers. It contains these sections:

[Section A.1, "Test Network Configuration"](#)

[Section A.2, "F5 Big IP Application Switch \(Software Version 4.5 PTF.5\)"](#)

[Section A.3, "Cisco CSM 3.1\(2\)"](#)

[Section A.4, "Foundry Server Iron v08.1.00cT24"](#)

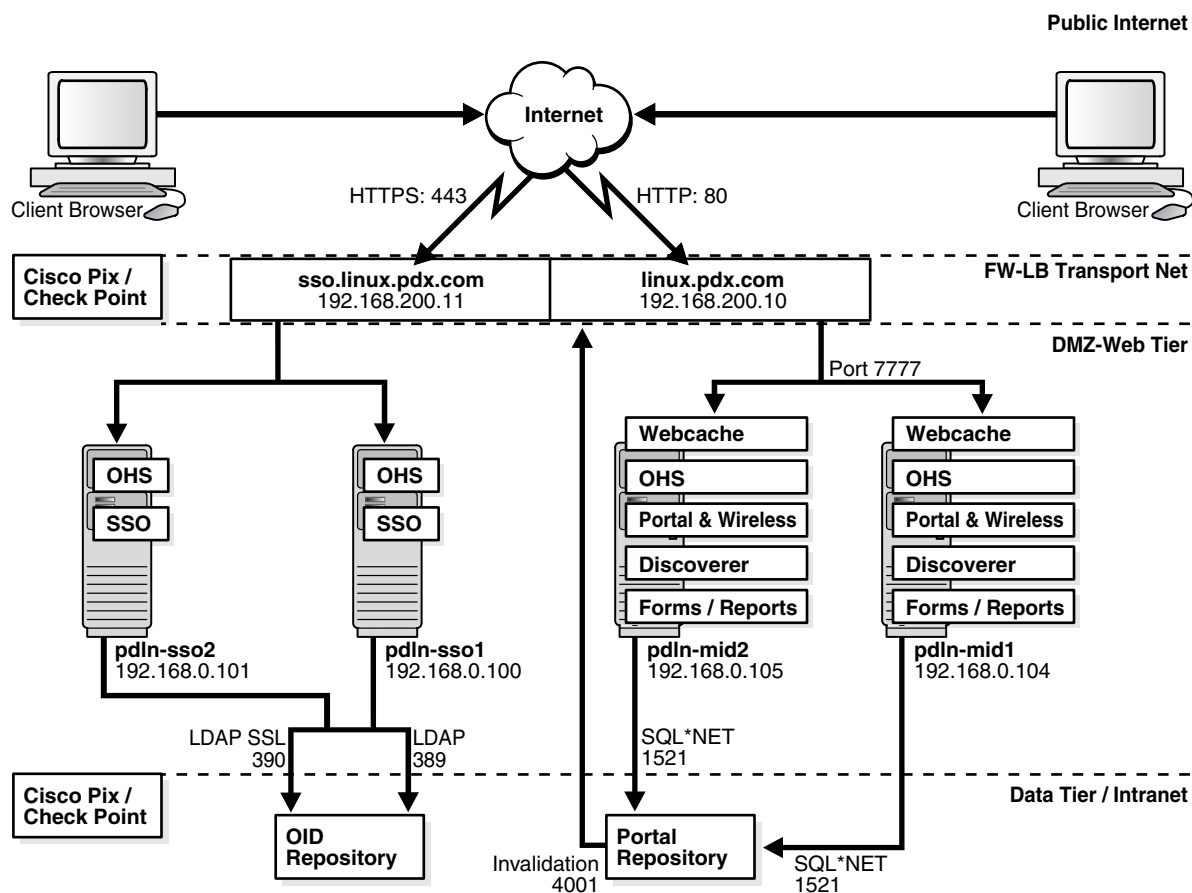
[Section A.5, "Nortel Alteon 2424 SSL \(Software Version 20.2.2.1\)"](#)

[Section A.6, "Radware Web Server Director NP with SynApps 7.50.05"](#)

### A.1 Test Network Configuration

This section identifies the elements of the network configuration and considerations for the operation of Oracle Application Server components. [Figure A-1](#) shows the configuration, its subnets, and the placement of the Oracle Application Server components in it.

Figure A-1 Test Network Configuration



### A.1.1 Network Subnets in the Test Configuration

The test network consists of several subnets for deployment of the hardware and Oracle Application Server components:

- **Internet**  
Simulated public network
- **Firewall-Load Balancer Transport Net**  
Network between the border firewall and load balancer external interface
- **DMZ or Web Tier**  
The OracleAS Single Sign-On middle tiers are installed on this tier. This subnet has two gateways:
  - Internal interface of the load balancer
  - Firewall interface to the data tier
- **Data Tier**  
The Oracle Application Server Infrastructure instance are installed on this tier. This is a protected network.



## A.1.2 Hardware in the Test Configuration

The test configuration contains the following hardware:

- Cisco Pix border or gateway firewall
- Check Point Firewall-1 NG internal firewall (DMZ to the Intranet)
- One of the following load balancers (F5 Big IP was used in Oracle tests):

[F5 Big IP Application Switch \(Software Version 4.5 PTF.5\)](#)

[Cisco CSM 3.1\(2\)](#)

[Foundry Server Iron v08.1.00cT24](#)

[Nortel Alteon 2424 SSL \(Software Version 20.2.2.1\)](#)

[Radware Web Server Director NP with SynApps 7.50.05](#)

## A.1.3 Configuration of Load Balancers and Firewalls for Oracle Application Server Component High Availability

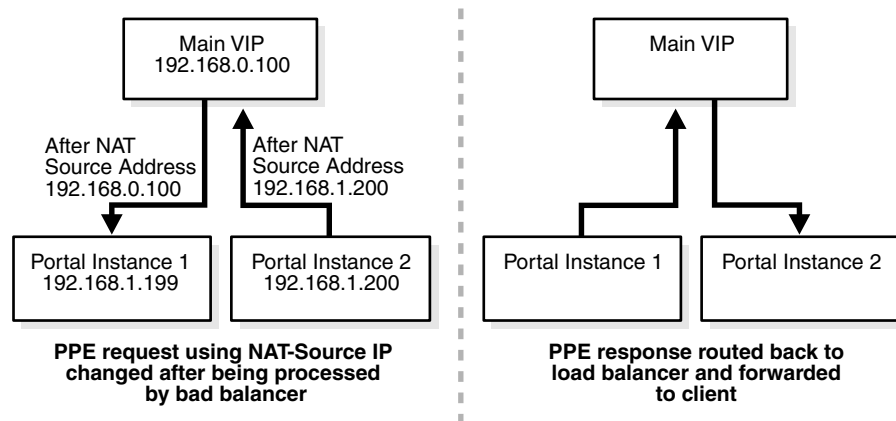
OracleAS Portal and OracleAS Wireless use server-to-server communication. This means that an OracleAS Portal or OracleAS Wireless instance must be able to make HTTP or HTTPS requests to a virtual IP address (VIP), and have the requests routed back to itself or another instance of its kind on the Web tier. The invalidation requests that OracleAS Portal makes to OracleAS Web Cache must be handled in a similar manner.

This section describes the communication in general terms and identifies the network configuration that enables it. For specific instructions on configuring a particular load balancer, refer to the section for that load balancer.

### A.1.3.1 OracleAS Portal Communication

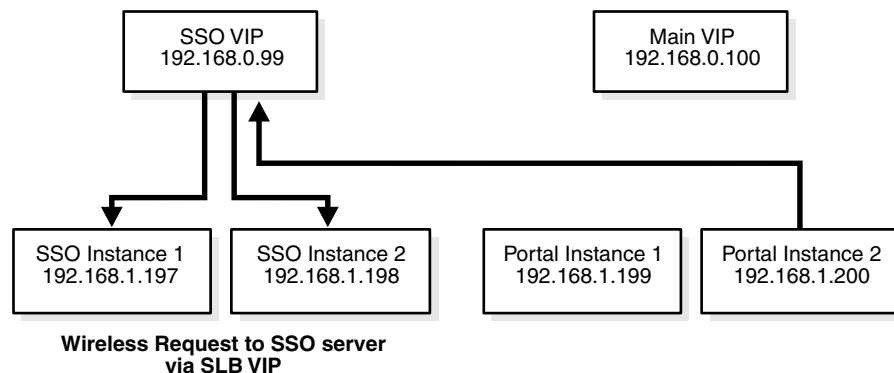
The Parallel Page Engine in OracleAS Portal makes loop-back (server-to-server) requests from the middle tier Oracle Application Server instance and back to that instance. In order to make OracleAS Portal highly available, these loop-back requests must be received by the load balancer, rather than individual Oracle Application Server middle tier instances.

After the Parallel Page Engine requests are routed to the VIP on the load balancer, the source address for the Parallel Page Engine requests must use Network Address Translation (NAT) to ensure correct routing. Without NAT on the source IP address of Parallel Page Engine requests, the host will respond directly to the client, which will break the session, since the client was expecting the response from the VIP. [Figure A-2](#) shows how an address is translated after the request is processed by the load balancer.

**Figure A–2 OracleAS Portal Parallel Page Engine Network Address Translation**

OracleAS Wireless makes requests to OracleAS Single Sign-On (which should be located with OracleAS Wireless on the Web tier). In order to make OracleAS Wireless highly available, these requests must be received by the load balancer. These requests must also be processed by NAT, as the OracleAS Single Sign-On and OracleAS Portal instances reside on the same subnet.

Figure A–3 shows the request from the OracleAS Portal instance to the OracleAS Single Sign-On load balancer.

**Figure A–3 Request Routing to the OracleAS Single Sign-On Server Load Balancer**

OracleAS Portal also makes invalidation requests to OracleAS Web Cache. In order for the invalidation to function correctly, you must enable communication on port 9401 from the OracleAS Portal repository to a VIP that can communicate with the OracleAS Web Cache instances on the Web tier. Depending on how routing is configured in the network, you may also need to use NAT for these requests, and open outbound ports as needed on the data tier.

## A.2 F5 Big IP Application Switch (Software Version 4.5 PTF.5)

This section describes the network configuration necessary to test the Big IP Application Switch load balancer with the Oracle Application Server 10g Release 2 (10.1.2) application server.

### A.2.1 Subnets for the Big IP Configuration

The following subnets were used in the Big IP configuration:

- External: 192.168.200.0/24 (DMZ2)
- Internal: 192.168.0.0/24 (DMZ1)

Two interfaces were created:

- 1.1 192.168.200.5/24 (External)
- 1.2 192.168.0.1/24 (Internal)

---

**Note:** In the configuration for port 1.2, Secure Network Address Translation (SNAT) automap was also enabled.

---

## A.2.2 Servers/Nodes for the Big IP Configuration

As shown in [Figure A-1, "Test Network Configuration"](#), the following servers were used for the middle tier installations and OracleAS Single Sign-On servers:

- pdln-mid1.pdx.com
- pdln-mid2.pdx.com
- pdln-sso1.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)
- pdln-sso2.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)

## A.2.3 Pools for the Big IP Configuration

The following pools were created:

### Pool 1: HTTP

- pdln-mid1.pdx.com (Port 7777)
- pdln-mid2.pdx.com (Port 7777)
- Enable SNAT

### Pool 2: OracleAS Single Sign-On

- pdln-sso1.pdx.com (Port 7777)
- pdln-sso2.pdx.com (Port 7777)
- Enable SNAT
- Persistent rebalance

### Pool 3: OracleAS Web Cache Invalidation

- pdln-mid1.pdx.com (Port 9401)
- pdln-mid2.pdx.com (Port 9401)
- Enable SNAT

## A.2.4 Virtual Servers (VIPs) for the Big IP Configuration

The following virtual servers were used:

**Table A-1 Virtual Servers for the Big IP Configuration**

Name	IP Address	Port	Pool
VIP1	192.168.200.10	80	1
VIP2	192.168.200.11	80	2

**Table A–1 (Cont.) Virtual Servers for the Big IP Configuration**

Name	IP Address	Port	Pool
VIP3	192.168.200.10	9401	3

## A.2.5 Load Balancing Method for the Big IP Configuration

The following load balancing methods were used:

- Middle tiers: Round Robin with basic HTTP health check
- Identity Management: Least Connections with OracleAS Single Sign-On health check (in-house)

## A.2.6 Health Monitors for the Big IP Configuration

You can create health monitors for Oracle Application Server components as described in this section.

### A.2.6.1 OracleAS Single Sign-On

Send String: GET /sso/status

Receive Rule: The OC4J\_SECURITY instance is running

### A.2.6.2 Middle Tier Components

Since there are multiple components running on the middle tiers, the best way to monitor this is with an HTTP GET /. You can also create customized health checks using OracleAS Portal and OracleAS Wireless status pages.

### A.2.6.3 OracleAS Web Cache Invalidation

A health monitor is needed for OracleAS Web Cache invalidation messages. Use HTTP LOGIN to monitor these messages.

### A.2.6.4 Oracle Internet Directory LDAP

Monitor Oracle Internet Directory LDAP communication using LDAP LOGIN.

### A.2.6.5 SSL Configuration

Because two different hosts (sso-linux and linux) were used, two proxies, each with its own certificate, were created:

#### ■ Proxy 1

Type: SSL

IP:Port: 192.168.200.10:443 (linux.pdx.com)

Destination Host: 192.168.200.10:80 (linux.pdx.com)

(Certificate information here)

#### ■ Proxy 2

Type: SSL

IP:Port: 192.168.200.11:443 (sso-linux.pdx.com)

Destination Host: 192.168.200.11:80 (sso-linux.pdx.com)

(Certificate information here)

These proxies decrypt the HTTPS session in Big IP's internal SSL accelerator and forward the HTTP traffic back to the VIP.

## A.2.7 OracleAS Portal Configuration Notes for Big IP

In order to use the load balancer to handle the Parallel Page Engine requests from the middle tiers, you must set up Secure Network Address Translation (SNAT) on the VLAN's self IP address and the middle tier pools. To do this, follow the instructions in this section.

1. In the network configuration, check SNAT Automap for the self IP of the internal interface.
2. In the middle tier pool configuration, ensure that SNAT is enabled and NAT is disabled.
3. Issue the following command:

```
b vlan internal snat automap enable
```

In the preceding command, *internal* is the IP address of the internal interface.

4. Test the configuration with a telnet command from one of the middle tiers to the VIP address on port 80, with a HEAD request, for example:

```
telnet 192.168.200.10 80
```

```
HEAD
```

A response similar to the following should be returned:

```
Date: Wed, 02 Jun 2004 15:08:25 GMT
```

```
Allow: GET, HEAD
```

```
Server: OracleAS-Web-Cache-10g/10.1.2.0.0
```

```
Content-Type: text/html
```

```
Content-Length: 100
```

```
Cache-Control: public
```

5. Ensure that SNAT is enabled on the pool that was created for invalidation requests. You may also need to create a static route on the firewall to ensure that invalidation requests are routed properly. (This is required, since the middle tier may have a different route to the database.)
6. If you are using SSL and routing Parallel Page Engine and Invalidation requests through the load balancer/SSL accelerator, you must import the trusted site certificate. To do this, follow the instructions in the *Oracle Application Server Portal Configuration Guide*, section titled "Adding Certificates for Trusted Sites".

## A.2.8 OracleAS Wireless Configuration Notes for Big IP

The configuration described in the preceding sections can also be applied to OracleAS Wireless. The only difference is that the middle tiers must know the IP address of the OracleAS Single Sign-On pool, and be able to route requests to that pool to authenticate clients. If you are using SSL, you must also import CA and Site certificates into the OracleAS Wireless configuration. See the *Oracle Application Server Wireless Administrator's Guide* for instructions.

## A.2.9 OracleAS Web Cache Configuration Notes for Big IP

If you are using OracleAS Web Cache with Big IP, ensure that the Big IP version is at least 4.5 PTF5, with the fix described in the F5 document 28154. Without this version and the fix, severe performance problems will occur. (In versions later than 4.5 PTF5, the problems have been fixed.)

## A.3 Cisco CSM 3.1(2)

This section describes the network configuration necessary to test the Cisco CSM 3.1(2) load balancer with the Oracle Application Server 10g Release 2 (10.1.2) application server.

### A.3.1 Subnets for the CSM 3.1(2) Configuration

The following subnets were used in the Cisco CSM 3.1(2) configuration:

- External: 192.168.200.0/24 (DMZ2)
- Internal: 192.168.0.0/24 (DMZ1)

### A.3.2 Servers/Nodes for the Cisco CSM 3.1(2) Configuration

As shown in [Figure A-1, "Test Network Configuration"](#), the following servers were used for the middle tier installations and OracleAS Single Sign-On servers:

- pdln-mid1.pdx.com
- pdln-mid2.pdx.com
- pdln-sso1.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)
- pdln-sso2.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)

### A.3.3 VLANs for the Cisco CSM 3.1(2) Configuration

The following VLANs were created:

- VLAN 2: Client
- VLAN 200: Server (Web tier)
- VLAN 400: Server (SSL)

### A.3.4 Server Farms for the Cisco CSM 3.1(2) Configuration

The following server farms were created:

- HTTPS\_POOL (Redirection to SSL Accelerator)
  - NAT server
  - No NAT client
  - Real 192.168.100.10
- LINUX\_FARM
  - NAT server
  - No NAT client
  - Real 192.168.0.104 7777
  - Real 192.168.0.105 7777

- LINUX\_FARM2
  - NAT server
  - NAT client SOURCENAT (for Parallel Page Engine requests)
  - Real 192.168.0.104 7777
  - Real 192.168.0.105 7777
- SSO\_FARM
  - NAT server
  - No NAT client
  - Real 192.168.0.101 7777
- SSO\_FARM2
  - NAT server
  - NAT client SOURCENAT
  - Real 192.168.0.101
- SSO\_SSL-A (Redirection to SSL Accelerator)
  - NAT server
  - No NAT client
  - Real 192.168.100.11
- WC\_INVALID (Web Cache Invalidation)
  - NAT server
  - NAT client WEBCACHE (for NAT of invalidation requests)
  - Real 192.168.0.101 9401
  - Real 192.168.0.105 9401

### A.3.5 Virtual Servers (VIPs) for the Cisco CSM 3.1(2) Configuration

This section describes the virtual servers in the Cisco CSM 3.1(2) configuration.

#### A.3.5.1 Virtual Servers for Outside Traffic Access to Server Farms

- HTTPS\_POOL (Redirect to SSL Accelerator)
  - Virtual 192.168.200.10 tcp https
  - Serverfarm HTTPS\_POOL
  - Sticky 120 group 4
  - No persistent rebalance
- HTTP\_POOL (HTTP direct to servers)
  - Virtual 192.168.200.11 tcp https
  - VLAN 2
  - Serverfarm LINUX\_FARM
  - Sticky 120 group 2
  - Idle 7200

Persistent rebalance

- SSO3 (SSL redirection to the SSL Accelerator)

Virtual 192.168.200.11 tcp https

VLAN 2

Serverfarm SSO\_SSL-A

Persistent rebalance

#### **A.3.5.2 Sticky Configuration**

sticky 2 netmask 255.255.255.255 timeout 120

sticky 3 ssl timeout 120

sticky 4 netmask 255.255.255.255 timeout 120

#### **A.3.5.3 Virtual Servers for HTTP Request Forwarding From the SSL Accelerator**

- HTTP\_POOL3 (Accept requests from the SSL Accelerator VLAN to the middle tiers)

Virtual 192.168.200.10 tcp www

VLAN 400

Serverfarm LINUX\_FARM

Persistent rebalance

- SSO (Accepts HTTP requests from the SSL Accelerator VLAN to the SSO servers)

Virtual 192.168.200.11 tcp https

VLAN 400

Serverfarm SSO\_FARM

Idle 7200

Persistent rebalance

#### **A.3.5.4 Virtual Servers for Traffic from VLAN for Parallel Page Engine Requests**

- HTTP-2 (Accept requests from the server VLAN for Parallel Page Engine loop-back)

Virtual 192.168.200.10 tcp www

VLAN 200

Serverfarm LINUX\_FARM2

Persistent rebalance

In order to allow the wireless authentication using OracleAS Single Sign-On, the following virtual server must be created on the middle tier VLAN to allow communication from the OracleAS Portal middle tier to the OracleAS Single Sign-On server's VIP:

- SSO2

Virtual 192.168.200.11 tcp https

VLAN 200

Serverfarm SSO\_FARM2



Persistent rebalance

The following virtual server is required for OracleAS Web Cache invalidation:

WEBCACHE\_INVAL

Virtual 192.168.200.10 tcp 9401

VLAN 200

Serverfarm WC\_INVAL

Persistent rebalance

To verify the Parallel Page Engine communication from the middle tiers, follow these steps:

1. Test the configuration with a telnet command from one of the middle tiers to the VIP address on port 80, with a HEAD request, for example:

```
telnet 192.168.200.10 80
```

```
HEAD
```

A response similar to the following should be returned:

```
Date: Wed, 02 Jun 2004 15:08:25 GMT
```

```
Allow: GET, HEAD
```

```
Server: OracleAS-Web-Cache-10g/10.1.2.0.0
```

```
Content-Type: text/html
```

```
Content-Length: 100
```

```
Cache-Control: public
```

---

**Note:** You can perform the same test for the invalidation communication from the Infrastructure database. Syntax errors may occur with these requests, but if the response contains the preceding information, the communication is functioning properly.

---

### A.3.6 Test Configuration: Cisco CSM 3.1(2)

```
Current configuration : 8198 bytes
!
! Last configuration change at 01:03:50 PDT Tue May 18 2004
! NVRAM config last updated at 01:03:52 PDT Tue May 18 2004
!
version 12.1
service timestamps debug datetime show-timezone
service timestamps log datetime show-timezone
no service password-encryption
!
hostname pd-cat6k
!
boot buffersize 522200
boot system slot0:c6sup22-jsv-mz.121-8a.EX

boot bootldr bootflash:c6msfc2-boot-mz.121-8a.E5.bin
enable secret 5 $1$u2be$MClIIqnBVnmCaNTtAMxLI/
!
clock timezone PST -8
clock summer-time PDT recurring
```

```
clock calendar-valid
redundancy
main-cpu
  auto-sync standard
diagnostic level complete
ip subnet-zero
!
!
no ip domain-lookup
!
no mls ip multicast aggregate
no mls ip multicast non-rpf cef
mls qos statistics-export interval 300
mls qos statistics-export delimiter |
module ContentSwitchingModule 3
  vlan 2 client
    ip address 192.168.200.5 255.255.255.0
    gateway 192.168.200.1
  !
  vlan 200 server
    ip address 192.168.0.1 255.255.255.0
  !
  vlan 400 server
    ip address 192.168.100.1 255.255.255.0
  !!
  natpool WEBCACHE 192.168.200.125 192.168.200.125 netmask 255.255.255.0
  natpool SOURCENAT 192.168.200.100 192.168.200.100 netmask 255.255.255.0
  !
  serverfarm HTTPS_POOL
    nat server
    no nat client
    real 192.168.100.10
    inservice
  !
  serverfarm LINUX_FARM
    nat server
    no nat client
    real 192.168.0.104 7777
    inservice
    real 192.168.0.105 7777
    inservice
  !
  serverfarm LINUX_FARM2
    nat server
    nat client SOURCENAT
    real 192.168.0.104 7777
    inservice
    real 192.168.0.105 7777
    inservice
  !
  serverfarm SSO_FARM
    nat server
    no nat client
    real 192.168.0.100 7777
    no inservice
    real 192.168.0.101 7777
    inservice
  !
  serverfarm SSO_FARM2
    nat server
```

```

    nat client SOURCENAT
    real 192.168.0.101 7777
    inservice
!
serverfarm SSO_SSL-A
    nat server
    no nat client
    real 192.168.100.11
    inservice
!
serverfarm WC_INVALID
    nat server
    nat client WEBCACHE
    real 192.168.0.104 9401
    inservice
    real 192.168.0.105 9401
    inservice
!
sticky 2 netmask 255.255.255.255 timeout 120
sticky 3 ssl timeout 120
sticky 4 netmask 255.255.255.255 timeout 120
!
vserver HTTP-2
    virtual 192.168.200.10 tcp www
    vlan 200
    serverfarm LINUX_FARM2
    persistent rebalance
    inservice
!
vserver HTTPS_POOL
    virtual 192.168.200.10 tcp https
    serverfarm HTTPS_POOL
    sticky 120 group 4
    idle 7200
    no persistent rebalance
    inservice
!
vserver HTTP_POOL
    virtual 192.168.200.10 tcp www
    vlan 2
    serverfarm LINUX_FARM
    sticky 120 group 4
    idle 7200
    persistent rebalance
    inservice
!
vserver HTTP_POOL3
    virtual 192.168.200.10 tcp www
    vlan 400
    serverfarm LINUX_FARM
    persistent rebalance
    inservice
!
vserver SSO
    virtual 192.168.200.11 tcp www
    vlan 400
    serverfarm SSO_FARM
    idle 7200
    persistent rebalance
    inservice

```

```
!  
vserver SS02  
  virtual 192.168.200.11 tcp https  
  vlan 200  
  serverfarm SSO_FARM2  
  persistent rebalance  
  inservice  
!  
vserver SS03  
  virtual 192.168.200.11 tcp https  
  vlan 2  
  serverfarm SSO_SSL-A  
  persistent rebalance  
  inservice  
!  
vserver WEBCACHE_INVALID  
  virtual 192.168.200.10 tcp 9401  
  vlan 200  
  serverfarm WC_INVALID  
  persistent rebalance  
  inservice  
!  
!  
!  
!  
interface GigabitEthernet1/1  
  no ip address  
  shutdown  
!  
interface GigabitEthernet1/2  
  no ip address  
  shutdown  
!  
interface FastEthernet2/1 (Management Interface)  
  ip address 138.1.33.105 255.255.255.128  
  duplex full  
  speed 100  
!  
interface FastEthernet2/2  
  no ip address  
  duplex full  
  speed 100  
  switchport  
  switchport access vlan 2  
  switchport mode access  
!  
interface FastEthernet2/3  
  no ip address  
  duplex full  
  speed 100  
  switchport  
  switchport access vlan 200  
  switchport mode access  
!  
interface FastEthernet2/4  
  no ip address  
  duplex full  
  speed 100  
  switchport  
  switchport access vlan 400
```

```
        switchport mode access
    !
interface FastEthernet2/5
    no ip address
    duplex full
    speed 100
    switchport
    switchport access vlan 400
    switchport mode access
    !
interface FastEthernet2/6
    no ip address
    duplex full
    speed 100
    switchport
    switchport access vlan 400
    switchport mode access
    !
interface FastEthernet2/7
    no ip address
    duplex full
    speed 100

    switchport
    switchport access vlan 400
    switchport mode access
    !
interface FastEthernet2/8
    no ip address
    duplex full
    speed 100
    switchport
    switchport access vlan 400
    switchport mode access
    !
interface FastEthernet2/9
    no ip address
    duplex full
    speed 100
    switchport
    switchport access vlan 400
    switchport mode access
    !
interface FastEthernet2/10
    no ip address
    duplex full
    speed 100
    switchport
    switchport access vlan 400
    switchport mode access
    !
interface FastEthernet2/11
    no ip address
    duplex full
    speed 100
    switchport
    switchport access vlan 200
    switchport mode access
    !
interface FastEthernet2/12
```

```

no ip address
duplex full
speed 100
switchport
switchport access vlan 200
switchport mode access
!
interface FastEthernet2/13
no ip address
duplex full
speed 100
switchport
switchport access vlan 200
switchport mode access
!
interface FastEthernet2/14
no ip address
duplex full
speed 100
switchport
switchport access vlan 200
switchport mode access
!
interface Vlan1
no ip address
shutdown
!
!
interface Vlan200
no ip address
!
ip default-gateway 138.1.34.229
ip classless
no ip http server
!
!
tftp-server slot0:c6slb-apc.2-1-1.bin
!
line con 0
line vty 0 4
password welcome
login
transport input lat pad mop telnet rlogin udptn nasi
!
end
pd-cat6k#

```

## A.4 Foundry Server Iron v08.1.00cT24

This section describes the network configuration necessary to test the Foundry Server Iron v08.1.00cT24 load balancer with the Oracle Application Server 10g Release 2 (10.1.2) application server.

### A.4.1 Subnets for the Foundry Server Iron v08.1.00cT24 Configuration

The following subnets were used in the Foundry Server Iron v08.1.00cT24 configuration:

- External: 192.168.200.0/24 (DMZ2)
- Internal: 192.168.0.0/24 (DMZ1)

## A.4.2 Servers/Nodes for the Foundry Server Iron v08.1.00cT24 Configuration

As shown in [Figure A-1, "Test Network Configuration"](#), the following servers were used for the middle tier installations and OracleAS Single Sign-On servers:

- pdln-mid1.pdx.com
- pdln-mid2.pdx.com
- pdln-cache1.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)
- pdln-cache2.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)

## A.4.3 Real Servers for the Foundry Server Iron v08.1.00cT24 Configuration

- Server103 192.168.0.105 (OracleAS Portal on pdln.mid1)  
Source-NAT  
Port 7777  
Port 9401
- Server102 192.168.0.104 (OracleAS Portal on pdln.mid2)  
Source-NAT  
Port 7777  
Port 9401
- Server101 192.168.200.101 (Identity Management and OracleAS Single Sign-On middle tier on pdln-cache1)  
Port 7777

To verify the Parallel Page Engine communication from the middle tiers, follow these steps:

1. Test the configuration with a telnet command from one of the middle tiers to the VIP address on port 80, with a HEAD request, for example:

```
telnet 192.168.200.10 80
```

```
HEAD
```

A response similar to the following should be returned:

```
Date: Wed, 02 Jun 2004 15:08:25 GMT
```

```
Allow: GET, HEAD
```

```
Server: OracleAS-Web-Cache-10g/10.1.2.0.0
```

```
Content-Type: text/html
```

```
Content-Length: 100
```

```
Cache-Control: public
```

---

---

**Note:** You can perform the same test for the invalidation communication from the Infrastructure database. Syntax errors may occur with these requests, but if the response contains the preceding information, the communication is functioning properly.

---

---

#### A.4.4 OracleAS Portal Configuration Notes for Foundry Server Iron v08.1.00cT24

In order for invalidation to work correctly, you must ensure that client NAT is enabled on each of the real servers on which OracleAS Web Cache is installed. You may also need to create a static route on the firewall to ensure that invalidation requests are routed properly.

If you are using SSL and routing Parallel Page Engine and Invalidation requests through the load balancer/SSL accelerator, you must import the trusted site certificate. To do this, follow the instructions in the *Oracle Application Server Portal Configuration Guide*, section titled "Adding Certificates for Trusted Sites".

#### A.4.5 OracleAS Wireless Configuration Notes for Foundry Server Iron v08.1.00cT24

The configuration described in the preceding sections can also be applied to OracleAS Wireless. The only difference is that the middle tiers must know the IP address of the OracleAS Single Sign-On pool, and be able to route requests to that pool to authenticate clients. If you are using SSL, you must also import CA and Site certificates into the OracleAS Wireless configuration. See the *Oracle Application Server Wireless Administrator's Guide* for instructions.

#### A.4.6 Test Configuration: Foundry Server Iron v08.1.00cT24

```

ver 08.1.00cT24
!
module 1 bi-0-port-wsm-management-module
module 2 bi-8-port-gig-copper-module
module 4 bi-24-port-copper-module
!
global-protocol-vlan
!
!
!
!
!
server real server103 192.168.0.105
source-nat
port 7777
port 9401
!
server real server102 192.168.0.104
source-nat
port 7777
port 9401
port 7778
!
server real server101 192.168.0.101
source-nat
port 7777
!
server cache-name ssl_10 192.168.100.10
port http

```



```

port http no-health-check
port http url "HEAD /"
port ssl
port ssl no-health-check
!
server cache-name ssl_11 192.168.100.11
port http
port http no-health-check
port http url "HEAD /"
port ssl
port ssl no-health-check
!
server real server100 192.168.0.100
source-nat
port 7777
!
!
server virtual 200_10 192.168.200.10
sym-priority 254
port http
port http spoofing
port 9401
port 7778
port ssl sticky
bind http server102 7777 server103 7777
bind 9401 server102 9401 server103 9401
bind ssl ssl_10 ssl
!
server virtual 200_11 192.168.200.11
sym-priority 254
port http
port http spoofing
port ssl sticky
bind http server100 7777
bind ssl ssl_11 ssl
!
server vip-group 1
vip 192.168.200.10
vip 192.168.200.11
server cache-group 1
cache-name ssl_10
cache-name ssl_11
!
!
vlan 1 name DEFAULT-VLAN by port
!
vlan 4092 name internal by port
untagged ethe 2/5 to 2/8 ethe 4/13 to 4/18 ethe 4/23 to 4/24
router-interface ve 1
!
vlan 4093 name external by port
untagged ethe 2/1 to 2/4 ethe 4/1 to 4/12
router-interface ve 2
!
vlan 4095 name SSL by port
untagged ethe 4/19 to 4/21
router-interface ve 3
!
!
hostname ServerIron_1

```

```

ip default-network 192.168.200.1/24
ip l4-policy 1 cache tcp 0 global
ip l4-policy 2 cache tcp ssl global
ip route 0.0.0.0 0.0.0.0 192.168.200.1
ip route 192.168.2.0 255.255.255.0 192.168.0.200
!
username twillard password .....
router vrrp
snmp-server community ..... rw
!
interface ethernet 2/1
confirm-port-up 6
!
interface ethernet 2/2
confirm-port-up 6
!
interface ethernet 2/3
confirm-port-up 6
!
interface ethernet 2/4
confirm-port-up 6
!
interface ethernet 2/5
confirm-port-up 6
!
interface ethernet 2/6
confirm-port-up 6
!
interface ethernet 2/7
confirm-port-up 6
!
interface ethernet 2/8
confirm-port-up 6
!
interface ethernet 4/1
speed-duplex 100-full
!
interface ethernet 4/13
speed-duplex 100-full
!
interface ve 1
ip address 192.168.0.1 255.255.255.0
ip vrrp vrid 1
owner
advertise backup
ip-address 192.168.0.1
vip-group 1
track-port ve 2
activate
!
interface ve 2
ip address 192.168.200.5 255.255.255.0
ip vrrp vrid 2
owner
advertise backup
ip-address 192.168.200.5
track-port ve 1
activate
!
interface ve 3

```

```

ip address 192.168.100.1 255.255.255.0
ip vrrp vrid 3
  owner
  advertise backup
  ip-address 192.168.100.1
  track-port ve 1
  activate
!
!
!
!
end

```

## A.5 Nortel Alteon 2424 SSL (Software Version 20.2.2.1)

This section describes the network configuration necessary to test the Nortel Alteon 2424 SSL (Software Version 20.2.2.1) load balancer with the Oracle Application Server 10g Release 2 (10.1.2) application server.

### A.5.1 Subnets for the Nortel Alteon 2424 SSL (Software Version 20.2.2.1) Configuration

The following subnets were used in the Foundry Server Iron v08.1.00cT24 configuration:

- External: 192.168.200.0/24 (DMZ2)
- Internal: 192.168.0.0/24 (DMZ1)

### A.5.2 Servers/Nodes for the Nortel Alteon 2424 SSL (Software Version 20.2.2.1) Configuration

As shown in [Figure A-1, "Test Network Configuration"](#), the following servers were used for the middle tier installations and OracleAS Single Sign-On servers:

- pdln-mid1.pdx.com
- pdln-mid2.pdx.com
- pdln-sso1.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)
- pdln-sso2.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)

### A.5.3 Real Servers for the Nortel Alteon 2424 SSL (Software Version 20.2.2.1) Configuration

You must create Real Server entries for each middle tier balanced by the load balancer. [Table A-2](#) lists the servers used in the test configuration.

**Table A-2 Real Servers**

Real	Real IP	Name
1	192.168.0.104	pdln-mid1
2	192.168.0.105	pdln-mid2
3	192.168.0.100	pdln-sso1
4	192.168.0.101	pdln-sso2
5	192.168.100.10	SSL Accelerator linux.pdx.com

## A.5.4 Groups for the Nortel Alteon 2424 SSL (Software Version 20.2.2.1) Configuration

The servers listed in [Table A–2](#) must belong to groups, as listed in [Table A–3](#). Note that the groups contain like instances, for example, Group 1 contains OracleAS Portal instances, Group 4 contains the Identity Management instances, and Group 5 has only the SSL accelerator.

**Table A–3 Groups**

Group	Servers	Metric
1	1, 2	Round robin
4	3, 4	Round robin
5	5	Round robin

## A.5.5 Virtual IP Addresses for Nortel Alteon 2424 SSL (Software Version 20.2.2.1)

This section describes the virtual IP addresses used in this configuration.

Virtual #1 is set up to listen on port 80 (HTTP) using the address 192.168.200.10, which is on the external subnet interface. Group 1 is bound to this virtual address, and the remote port 7777 (the OracleAS Web Cache listen port) has also been set. Pbind is for client stickiness; since we are using an OracleAS Web Cache cluster in this scenario, no real session binding is needed on the load balancer.

Virtual #4 is for OracleAS Single Sign-On, and is also configured on port 80 (can be set to 443 for SSL communication), using the address 192.168.200.11, which is on the external subnet interface. Group 4 is bound to this virtual server and the remote port 7777. No session binding is needed for the OracleAS Single Sign-On requests, but for his instance client IP has been selected.

**Table A–4 Virtual IP Addresses**

Number	Service	VIP	Dname	Group	Pbind	Rport
1	HTTP	192.168.200.10	linux.pdx.com	1	Clientip	7777
1	9401	192.168.200.10	N/A	1		
4	HTTP	192.168.200.11	sso-linux.pdx.com	4	Clientip	7777

## A.5.6 Additional Server Configuration for Nortel Alteon 2424 SSL (Software Version 20.2.2.1)

To make the OracleAS Portal Parallel Page Engine and invalidation to work correctly, you must enable a proxy on the internal or server ports of the load balancer. This causes NAT (with PIP addresses) on any requests that are generated by the internal servers.

**PIP Configuration:** Configure PIP addresses that the proxy will use: For example:

```
/c/slb/pip<#>xxx.xxx.xxx.xxx
```

Replace the xs in the preceding example with the PIP address. The PIP addresses must be on the same subnet as the servers.

**Port Configuration:**

Port 1 (External): client enable, proxy enable

Port 2 (Internal server): client enable, proxy enable, server enable

Ports 3-8: client enable

### A.5.7 OracleAS Portal Configuration Notes for Nortel Alteon 2424 SSL (Software Version 20.2.2.1)

In order for invalidation to work correctly, you must ensure that client NAT is enabled on each of the real servers on which OracleAS Web Cache is installed. You may also need to create a static route on the firewall to ensure that invalidation requests are routed properly.

If you are using SSL and routing Parallel Page Engine and Invalidation requests through the load balancer/SSL accelerator, you must import the trusted site certificate. To do this, follow the instructions in the *Oracle Application Server Portal Configuration Guide*, section titled "Adding Certificates for Trusted Sites".

### A.5.8 OracleAS Wireless Configuration Notes for Nortel Alteon 2424 SSL (Software Version 20.2.2.1)

The configuration described in the preceding sections can also be applied to OracleAS Wireless. The only difference is that the middle tiers must know the IP address of the OracleAS Single Sign-On pool, and be able to route requests to that pool to authenticate clients. If you are using SSL, you must also import CA and Site certificates into the OracleAS Wireless configuration. See the *Oracle Application Server Wireless Administrator's Guide* for instructions.

### A.5.9 Test Configuration: Nortel Alteon 2424 SSL (Software Version 20.2.2.1)

```
script start "Alteon Application Switch 2424-SSL" 4 /**** DO NOT EDIT THIS LINE!
/* Configuration dump taken 10:47:15 Thu Jun  3, 2004
/* Version 20.2.2.1, Base MAC address 00:01:81:2e:b8:50
/c/sys
http ena
/c/sslproc/
mip 192.168.100.15
rts ena
/c/port 1
pvid 2
/c/port 1/fast
speed 100
fctl none
mode full
auto off
/c/port 2
pvid 3
/c/port 2/fast
speed 100
fctl none
mode full
auto off
/c/port 3
pvid 2
/c/port 3/fast
speed 100
fctl both
mode full
auto on
/c/port 4
pvid 4
/c/port 4/fast
speed 100
fctl both
```

```
mode full
auto on
/c/port 5
pvid 4
/c/port 5/fast
speed 100
fctl both
mode full
auto on
/c/port 6
pvid 4
/c/port 6/fast
speed 100
fctl both
mode full
auto on
/c/port 7
pvid 4
/c/port 7/fast
speed 100
fctl both
mode full
auto on
/c/port 8
pvid 4
/c/port 8/fast
speed 100
fctl both
mode full
auto on
/c/port 9
tag ena
pvid 4
/c/port 9/fast
speed any
fctl both
mode full
auto on
/c/vlan 1
def 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
/c/vlan 2
ena
name "Outside-Virtual"
def 1 3
/c/vlan 3
ena
name "DMZ"
def 2
/c/vlan 4
ena
name "SSL"
def 4 5 6 7 8 9
/c/vlan 99
ena
name "VLAN 99"
def 0
/c/stp 1/off
/c/stp 1/clear
/c/stp 1/add 1 2 3 4 99
/c/ip/if 1
```

```
ena
addr 192.168.200.5
vlan 2
/c/ip/if 2
ena
addr 192.168.0.1
vlan 3
/c/ip/if 3
ena
addr 192.168.100.1
vlan 4090
/c/ip/gw 1
ena
addr 192.168.200.1
retry 1
/c/ip/route
add 192.168.2.0 255.255.255.0 192.168.0.200 2
/c/slb
on
/c/slb/adv
direct ena
/c/slb/real 1
ena
rip 192.168.0.104
inter 15
retry 6
/c/slb/real 2
ena
rip 192.168.0.105
inter 15
retry 6
/c/slb/real 3
ena
rip 192.168.0.100
inter 15
retry 6
/c/slb/real 4
dis
rip 192.168.0.101
inter 15
retry 6
/c/slb/real 5
ena
rip 192.168.100.10
/c/slb/group 1
metric roundrobin
add 1
add 2
/c/slb/group 2
metric roundrobin
/c/slb/group 4
metric roundrobin
add 3
add 4
/c/slb/group 5
health sslh
add 5
/c/slb/pip/pip1 192.168.0.150
/c/slb/pip/pip2 192.168.0.151
/c/slb/pip/pip3 192.168.0.152
```

```
/c/slb/pip/pip4 192.168.0.153
/c/slb/port 1
client ena
proxy ena
/c/slb/port 2
client ena
server ena
proxy ena
/c/slb/port 3
client ena
/c/slb/port 4
client ena
/c/slb/port 5
client ena
/c/slb/port 6
client ena
/c/slb/port 7
client ena
/c/slb/port 8
client ena
/c/slb/virt 1
ena
vip 192.168.200.10
dname "linux.pdx.com"
/c/slb/virt 1/service http
group 1
rport 7777
pbind clientip
/c/slb/virt 1/service 9401
group 1
/c/slb/virt 4
ena
vip 192.168.200.11
dname "sso-linux.pdx.com"
/c/slb/virt 4/service http
group 4
rport 7777
pbind clientip
/c/slb/virt 2/service 443/pbind sslid
/c/slb/filt 5
ena
action redir
proto tcp
dport https
group 5
rport 0
vlan any
/c/slb/port 1
filt ena
add 5
/c/slb/port 2
filt ena
add 5
/
script end /**** DO NOT EDIT THIS LINE!

SSL Configuration:
SSL >> Configuration# dump

Dump private keys (yes/no) [no]: no
```



```

Collecting data, please wait...
/*
/*
/* Configuration dump taken Tue Aug  3 12:54:14 PDT 2004
/* Version 4.1.2.3
/*
/*
/*
/cfg/.
/cfg/ssl/.
/cfg/ssl/dns/.
    cachesize 1000
    retransmit 2s
    count 3
    ttl 3h
    health 10s
    hdown 2
    hup 2
    fallthrough off
/cfg/ssl/cert 1/.
    name PDCQA-CA
    cert
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
...
/cfg/ssl/cert 1/revoke/.
/cfg/ssl/cert 1/revoke/automatic/.
    interval 1d
    ena disabled
/cfg/ssl/cert 2/.
    name linux.pdx.com
    cert
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
...
/cfg/ssl/cert 2/revoke/.
/cfg/ssl/cert 2/revoke/automatic/.
    interval 1d
    ena disabled
/cfg/ssl/cert 4/.
    name sso-linux.pdx.com
    cert
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
...
/cfg/ssl/cert 4/revoke/.
/cfg/ssl/cert 4/revoke/automatic/.
    interval 1d
    ena disabled
/cfg/ssl/server 1/.
    name linux.pdx.com
    vip 192.168.200.10
    port "443 (https)"
    rip 0.0.0.0
    rport "80 (http)"
    type http
    proxy off
    ena enabled
/cfg/ssl/server 1/trace/.
/cfg/ssl/server 1/ssl/.

```

```
cert 2
cachesize 9400
cachettl 5m
cacerts 1
cachain 1
protocol ssl3
verify none
ciphers ALL@STRENGTH
ena enabled
/cfg/ssl/server 1/tcp/.
  cwrite 15m
  ckeep 15m
  swrite 15m
  sconnect 10s
  csendbuf auto
  crecbuf auto
  ssendbuf auto
  srecbuf 6000
/cfg/ssl/server 1/http/.
  redirect on
  sslheader on
  addxfor off
  addvia on
  addxisd off
  addfront off
  addclcert off
  addbeassl off
  addbeaccli off
  addnostore off
  cmsie shut
  rhost off
  maxrcount 40
  maxline 8192
/cfg/ssl/server 1/http/rewrite/.
  rewrite off
  ciphers HIGH:MEDIUM
  response iSD
  URI "/cgi-bin/weakcipher"
/cfg/ssl/server 1/http/auth/.
  mode basic
  realm Xnet
  proxy off
  ena disabled
/cfg/ssl/server 1/dns/.
/cfg/ssl/server 1/adv/.
/cfg/ssl/server 1/adv/pool/.
  timeout 15s
  ena disabled
/cfg/ssl/server 1/adv/traflog/.
  sysloghost 0.0.0.0
  udpport 514
  priority info
  facility local4
  ena disabled
/cfg/ssl/server 1/adv/standalone/.
  ena disabled
/cfg/ssl/server 1/adv/standalone/iplist/.
/cfg/ssl/server 1/adv/loadbalancing/.
  type all
  persistence none
```

```

        metric hash
        health auto
        interval 10s
        ena disabled
/cfg/ssl/server 1/adv/loadbalancing/script/.
/cfg/ssl/server 1/adv/loadbalancing/remotessl/.
    protocol ssl3
    ciphers ALL
/cfg/ssl/server 1/adv/loadbalancing/remotessl/verify/.
    verify none
/cfg/ssl/server 1/adv/sslconnect/.
    protocol ssl3
    ciphers EXP-RC4-MD5:ALL!DH
    ena disabled
/cfg/ssl/server 1/adv/sslconnect/verify/.
    verify none
/cfg/ssl/server 4/.
    Name sso-linux.pdx.com
    vip 192.168.200.11
    port "443 (https)"
    rip 0.0.0.0
    rport "80 (http)"
    type generic
    proxy off
    ena enabled
/cfg/ssl/server 4/trace/.
/cfg/ssl/server 4/ssl/.
    cert 4
    cachesize 9400
    cachettl 5m
    protocol ssl3
    verify none
    ciphers ALL@STRENGTH
    ena enabled
/cfg/ssl/server 4/tcp/.
    cwrite 15m
    ckeep 15m
    swrite 15m
    sconnect 10s
    csendbuf auto
    crecbuf auto
    ssendbuf auto
    srecbuf 6000
/cfg/ssl/server 4/adv/.
/cfg/ssl/server 4/adv/standalone/.
    ena disabled
/cfg/ssl/server 4/adv/standalone/iplist/.
/cfg/ssl/server 4/adv/loadbalancing/.
    type all
    persistence none
    metric hash
    health auto
    interval 10s
    ena disabled
/cfg/ssl/server 4/adv/loadbalancing/script/.
/cfg/ssl/server 4/adv/loadbalancing/remotessl/.
    protocol ssl3
    ciphers ALL
/cfg/ssl/server 4/adv/loadbalancing/remotessl/verify/.
    verify none

```

```
/cfg/ssl/server 4/adv/sslconnect/.
    protocol ssl3
    ciphers EXP-RC4-MD5:ALL!DH
    ena disabled
/cfg/ssl/server 4/adv/sslconnect/verify/.
    verify none
/cfg/xnet/.
    ttl 15m
    log login
/cfg/sys/.
/cfg/sys/routes/.
/cfg/sys/time/.
    tzone "America/Los_Angeles"
/cfg/sys/time/ntp/.
/cfg/sys/dns/.
/cfg/sys/syslog/.
/cfg/sys/cluster/.
    mip 192.168.100.15
/cfg/sys/cluster/host 1/.
    type master
    ip 192.168.100.10
    gateway 192.168.100.1
/cfg/sys/cluster/host 1/routes/.
/cfg/sys/cluster/host 1/interface 1/.
    ip 192.168.100.10
    netmask 255.255.255.0
    vlanid 0
    mode failover
    primary 0
/cfg/sys/cluster/host 1/interface 1/ports/.
    add 1
/cfg/sys/accesslist/.
/cfg/sys/adm/.
    clitimeout 10m
    telnet off
    ssh off
/cfg/sys/adm/snmp/.
/cfg/sys/adm/snmp/snmpv2-mib/.
    snmpEnableAuthenTraps disabled
/cfg/sys/adm/snmp/community/.
    read public
    trap trap
/cfg/sys/adm/audit/.
    vendorid "1872 (alteon)"
    vendortype 2
    ena false
/cfg/sys/adm/audit/servers/.
/cfg/sys/adm/http/.
    port 80
    ena false
/cfg/sys/adm/https/.
    port 443
    ena false
/cfg/sys/user/.
    expire 0
```

## A.6 Radware Web Server Director NP with SynApps 7.50.05

This section describes the network configuration necessary to test the Radware Web Server Director NP load balancer with the Oracle Application Server 10g Release 2 (10.1.2) application server.

### A.6.1 Subnets for the Radware Web Server Director NP Configuration

The following subnets were used in the Foundry Server Iron v08.1.00cT24 configuration:

- External: 192.168.200.0/24 (DMZ2)
- Internal: 192.168.0.0/24 (DMZ1)

### A.6.2 Servers/Nodes for the Radware Web Server Director NP Configuration

As shown in [Figure A-1, "Test Network Configuration"](#), the following servers were used for the middle tier installations and OracleAS Single Sign-On servers:

- pdln-mid1.pdx.com
- pdln-mid2.pdx.com
- pdln-sso1.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)
- pdln-sso2.pdx.com (Identity Management, OracleAS Single Sign-On middle tier)

### A.6.3 Farms for the Radware Web Server Director NP Configuration

The following farms were created for the Radware Web Server Director NP Configuration:

**Farm 1:** 192.168.0.150 HTTP

**Farm 2:** 192.168.0.151 OracleAS Web Cache invalidation

**Farm 3:** 192.168.0.152 OracleAS Single Sign-On

**Farm 4:** 192.168.0.153 CT100 — linux.pdx.com

**Farm 5:** 192.168.0.154 CT100 — sso-linux.pdx.com

### A.6.4 Servers for the Radware Web Server Director NP Configuration

[Table A-5](#) lists the servers used in the test configuration.

**Table A-5 Servers**

Farm Address	Server Address	Name	Multiplexed Server Port
192.168.0.150	192.168.0.104	pdln-mid2	7777
192.168.0.150	192.168.0.105	pdln-mid1	7777
192.168.0.151	192.168.0.104	pdln-mid2	7777
192.168.0.151	192.168.0.105	pdln-mid2	7777
192.168.0.152	192.168.0.100	pdln-sso1 (OracleAS Single Sign-On)	7777
192.168.0.152	192.168.0.101	pdln-sso2 (OracleAS Single Sign-On)	7777
192.168.0.153	192.168.100.10	CT100 (linux.pdx.com)	7777
192.168.0.154		CT100 (sso-linux.pdx.com)	7777

## A.6.5 Additional Server Configuration for the Radware Web Server Director NP

The following additional configuration is necessary for the Radware Web Server Director NP:

1. Enable client NAT. Do not specify any address under **Use Specific NAT Address**.
2. Specify the NAT address range to use.
3. Specify the client addresses for NAT:  
192.168.0.104 - 192.168.0.105 for middle tier  
192.168.2.100 - 192.168.2.100 for Infrastructure invalidation requests.
4. Specify client NAT **Enable** in the server configuration.

## A.6.6 Super Farms for the Radware Web Server Director NP Configuration

Table A-6 lists the super farms for the Radware Web Server Director NP configuration:

**Table A-6 Super Farms**

IP Address	Port Number	Farm Address	Function
192.168.200.10	80	192.168.0.150	linux.pdx.com HTTP
192.168.200.10	443	192.168.0.153	linux.pdx.com HTTPS --> CT100
192.168.200.10	9401	192.168.0.151	Invalidation VIP
192.168.200.11	80	192.168.0.152	OracleAS Single Sign-On HTTP
192.168.200.11	443	192.168.0.154	OracleAS Single Sign-On HTTPS --> CT100

## A.6.7 Load Balancing Method for the Radware Web Server Director NP Configuration

The following load balancing methods were used:

- Middle tiers: Cyclic with HTTP health check on port 7777
- Identity Management: Cyclic with HTTP health check on port 7777

To verify the Parallel Page Engine communication from the middle tiers, follow these steps:

1. Test the configuration with a telnet command from one of the middle tiers to the VIP address on port 80, with a HEAD request, for example:

```
telnet 192.168.200.10 80
```

```
HEAD
```

A response similar to the following should be returned:

```
Date: Wed, 02 Jun 2004 15:08:25 GMT
```

```
Allow: GET, HEAD
```

```
Server: OracleAS-Web-Cache-10g/10.1.2.0.0
```

```
Content-Type: text/html
```

```
Content-Length: 100
```

```
Cache-Control: public
```

---

**Note:** You can perform the same test for the invalidation communication from the Infrastructure database. Syntax errors may occur with these requests, but if the response contains the preceding information, the communication is functioning properly.

---

## A.6.8 OracleAS Portal Configuration Notes for Radware Web Server Director NP

In order for invalidation to work correctly, you must ensure that client NAT is enabled on each of the real servers on which OracleAS Web Cache is installed. You may also need to create a static route on the firewall to ensure that invalidation requests are routed properly.

If you are using SSL and routing Parallel Page Engine and Invalidation requests through the load balancer/SSL accelerator, you must import the trusted site certificate. To do this, follow the instructions in the *Oracle Application Server Portal Configuration Guide*, section titled "Adding Certificates for Trusted Sites".

## A.6.9 OracleAS Wireless Configuration Notes for Radware Web Server Director NP

The configuration described in the preceding sections can also be applied to OracleAS Wireless. The only difference is that the middle tiers must know the IP address of the OracleAS Single Sign-On pool, and be able to route requests to that pool to authenticate clients. If you are using SSL, you must also import CA and Site certificates into the OracleAS Wireless configuration. See the *Oracle Application Server Wireless Administrator's Guide* for instructions.

## A.6.10 Test Configuration: Radware Web Server Director NP

```
system config

!
!Device Configuration
!Date: 15-06-2004 21:44:33
!Device Description: Web Server Director NP with SynApps
!Base MAC Address: 00:03:b2:0d:43:c0
!Software Version: 7.50.05 (build 49dee4)
!
net route table cdbset 192.168.4.2 255.255.255.255 192.168.0.200
net route table cdbset 192.168.2.0 255.255.255.0 192.168.0.200
net route table cdbset 0.0.0.0 0.0.0.0 192.168.200.1
manage snmp community-table cdbset 0.0.0.0 public -ca super -st trapsEnable
system tune bridge-fft-table cdbset 1024
system tune ip-fft-table cdbset 8192
system tune arp-table cdbset 1024
system tune client-table cdbset 16384
system tune routing-table cdbset 512
wsd farm table cdbset 192.168.0.151 WCACHE_INVALID -as enable
wsd farm table cdbset 192.168.0.154 CT100-SSO -as enable -dm cyclic -cp 443
wsd farm table cdbset 192.168.0.154 CT100-SSO -as enable -dm cyclic -cp 443
wsd farm table cdbset 192.168.0.153 CT100 -as enable -dm cyclic -cp 443
wsd farm table cdbset 192.168.0.153 CT100 -as enable -dm cyclic -cp 443
wsd farm table cdbset 192.168.0.150 HTTP -as enable -dm cyclic -cp 7777
wsd farm table cdbset 192.168.0.150 HTTP -as enable -dm cyclic -cp 7777
wsd farm table cdbset 192.168.0.152 SSO -as enable -dm cyclic -cp 7777
wsd farm table cdbset 192.168.0.152 SSO -as enable -dm cyclic -cp 7777
wsd farm table cdbset 192.168.0.151 WCACHE_INVALID -as enable -dm cyclic
wsd farm table cdbset 192.168.0.151 WCACHE_INVALID -as enable -dm cyclic
```

```
wsd farm table cdbset 192.168.0.151 WCACHE_INVALID -as enable -dm cyclic
wsd farm server table cdbset 192.168.0.154 192.168.100.11 ct100-sso
wsd farm server table cdbset 192.168.0.153 192.168.100.10 CT100
wsd farm server table cdbset 192.168.0.150 192.168.0.105 pdln-mid1
wsd farm server table cdbset 192.168.0.150 192.168.0.104 pdln-mid2
wsd farm server table cdbset 192.168.0.152 192.168.0.100 pdln-cache1
wsd farm server table cdbset 192.168.0.151 192.168.0.105 pdln-mid1
wsd farm server table cdbset 192.168.0.151 192.168.0.104 pdln-mid2
wsd physical-server statistics cdbset pdln-cache1
wsd physical-server statistics cdbset pdln-mid2
wsd physical-server statistics cdbset ct100-sso
wsd physical-server statistics cdbset CT100
wsd physical-server statistics cdbset pdln-mid1
wsd super-farm cdbset 192.168.200.11 443 192.168.0.154
wsd super-farm cdbset 192.168.200.10 443 192.168.0.153
wsd super-farm cdbset 192.168.200.11 80 192.168.0.152
wsd super-farm cdbset 192.168.200.10 80 192.168.0.150
wsd super-farm cdbset 192.168.200.10 9401 192.168.0.151
wsd nat server status cdbset disable
system tune dynamic-proximity-table cdbset 4096
wsd farm connectivity-check httpcode cdbset 192.168.0.154 200
wsd farm connectivity-check httpcode cdbset 192.168.0.153 200
wsd farm connectivity-check httpcode cdbset 192.168.0.152 200
wsd farm connectivity-check httpcode cdbset 192.168.0.150 200
wsd farm connectivity-check httpcode cdbset 192.168.0.151 200
wsd nat server specific-nat-address cdbset 0.0.0.0
system tune url-table cdbset 256
system tune request-table cdbset 200
system tune ssl-id-table cdbset 1024
net next-hop-router cdbset 192.168.200.1
net next-hop-router cdbset 138.1.34.229
wsd farm nhr cdbset 0.0.0.0 -ip 192.168.200.1
wsd farm extended-params cdbset 192.168.0.150
net ip-interface cdbset 192.168.200.5 255.255.255.0 2
net ip-interface cdbset 192.168.100.1 255.255.255.0 16
net ip-interface cdbset 192.168.0.1 255.255.255.0 1
wsd nat client address-range cdbset 192.168.0.25 -t 192.168.0.25
wsd nat client range-to-nat cdbset 192.168.2.100 -t 192.168.2.155
wsd nat client range-to-nat cdbset 192.168.0.100 -t 192.168.0.105
wsd nat client status cdbset enable
system tune nat-address-table cdbset 1
system tune nat-ports-table cdbset 64512
bwm modify policy cdbset Default -i 0 -dst any -src any
bwm modify policy cdbset Default -i 0 -dst any -src any -dr oneway
health-monitoring response-level-samples cdbset 0
manage user table cdbset radware -pw radware

manage telnet status cdbset enable
manage web status cdbset enable
manage ssh status cdbset enable
manage secure-web status cdbset enable
net physical-interface cdbset 1 -s fe100 -d full -a on
net physical-interface cdbset 2 -s fe100 -d full
wsd#
```



## Sample Files and Values

This appendix contains sample files and recommended values you will use throughout the Enterprise Deployment configuration.

### B.1 Metadata Repository Tablespaces

Tablespaces for raw devices in the Metadata Repository are listed in [Table B-1](#), with minimum sizes and recommended names.

**Table B-1** Raw Devices for the OracleAS Metadata Repository

Tablespace	Minimum Size (MB)	Recommended Name
PORTAL	128	<i>dbname_raw_portal_128m</i>
PORTAL_DOC	64	<i>dbname_raw_portaldoc_64m</i>
PORTAL_IDX	64	<i>dbname_raw_portalidx_64m</i>
PORTAL_LOG	64	<i>dbname_raw_portallog_64m</i>
DCM	256	<i>dbname_raw_dcm_256m</i>
OCATS	64	<i>dbname_raw_ocats_64m</i>
DISCO_PTM5_CACHE	64	<i>dbname_raw_discoptm5cache_64m</i>
DISCO_PTM5_META	64	<i>dbname_raw_discoptm5meta_64m</i>
WCRSYS_TS	64	<i>dbname_raw_wcrsysys_64m</i>
UDDISYS_TS	64	<i>dbname_raw_uddisysys_64m</i>
OLTS_ATTRSTORE	128	<i>dbname_raw_oltsattrstore_128m</i>
OLTS_BTTRSTORE	64	<i>dbname_raw_oltsbttrstore_128m</i>
OLTS_CT_STORE	256	<i>dbname_raw_oltsctstore_256m</i>
OLTS_DEFAULT	128	<i>dbname_raw_oltsdefault_128m</i>
OLTS_SVRMGSTORE	64	<i>dbname_raw_oltssvrmgstore_64m</i>
IAS_META	256	<i>dbname_raw_iasmeta1_128m</i>
DSGATEWAY_TAB	64	<i>dbname_raw_dsgatewaytab_64m</i>

### B.2 Tablespace Mapping to Raw Devices Sample File

[Example B-1](#) shows the format of the file you use to map tablespaces to raw devices. The DBCA\_RAW\_CONFIG environment variable reads this file during tablespace creation.

**Example B–1 Tablespace to Raw Device Mapping (Sample File)**

```
PORTAL1=/dev/vx/rdisk/oracle/mydb_raw_portal_128m
PORTAL_DOC1=/dev/vx/rdisk/oracle/mydb_raw_portal_doc_64m
PORTAL_IDX1=/dev/vx/rdisk/oracle/mydb_raw_portal_idx_64m
PORTAL_LOG1=/dev/vx/rdisk/oracle/mydb_raw_portal_log_64m
IAS_META1=/dev/vx/rdisk/oracle/mydb_raw_ias_meta_256m
DISCO_PTM5_META1=/dev/vx/rdisk/oracle/mydb_raw_disco_meta_64m
DISCO_PTM5_CACHE1=/dev/vx/rdisk/oracle/mydb_raw_disco_cache_64m
DCM1=/dev/vx/rdisk/oracle/mydb_raw_dcm_256m
WCRSYS_TS1=/dev/vx/rdisk/oracle/mydb_raw_clip_64m
OCATS1=/dev/vx/rdisk/oracle/mydb_raw_oca_64m
UDDISYS_TS1=/dev/vx/rdisk/oracle/mydb_raw_uddi_64m
OLTS_ATTRSTORE1=/dev/vx/rdisk/oracle/mydb_raw_olts_attr_128m
OLTS_BATTRSTORE1=/dev/vx/rdisk/oracle/mydb_raw_olts_battr_64m
OLTS_CT_STORE1=/dev/vx/rdisk/oracle/mydb_raw_olts_ct_store_256m
OLTS_DEFAULT1=/dev/vx/rdisk/oracle/mydb_raw_olts_default_128m
OLTS_SVRMGSTORE1=/dev/vx/rdisk/oracle/mydb_raw_olts_svrmgstore_64m
DSGATEWAY_TAB1=/dev/vx/rdisk/oracle/mydb_raw_synd_64m
b2b_dt1=/dev/vx/rdisk/oracle/mydb_raw_b2b_dt_256m
b2b_rt1=/dev/vx/rdisk/oracle/mydb_raw_b2b_rt_256m
b2b_lob1=/dev/vx/rdisk/oracle/mydb_raw_b2b_lob_256m
b2b_idx1=/dev/vx/rdisk/oracle/mydb_raw_b2b_idx_256m
```

## B.3 Using the Static Ports Feature with Oracle Universal Installer

The Static Ports feature enables you to assign ports during installation. The Oracle Universal Installer reads the `staticports.ini` file, assigning the port values to Oracle Application Server components as specified.

A sample `staticports.ini` file, shown in [Example B–2](#), is provided on:

Disk 1: `mount_point/1012disk1/stage/Response/staticports.ini`

**Example B–2 Sample staticports.ini File**

```
# staticports.ini Template File

# This file is a template for specifying port numbers at installation time.
# To specify a port number, uncomment the appropriate line (remove #) and
# replace "port_num" with the desired port number.
# You can then launch Oracle Universal Installer with special options to use this
file.
# Please refer to Oracle Application Server 10g Installation Guide for
instructions.

# J2EE and Web Cache

#Oracle HTTP Server port = port_num
#Oracle HTTP Server Listen port = port_num
#Oracle HTTP Server SSL port = port_num
#Oracle HTTP Server Listen (SSL) port = port_num
#Oracle HTTP Server Diagnostic port = port_num
#Java Object Cache port = port_num
#DCM Java Object Cache port = port_num
#DCM Discovery port = port_num
#Oracle Notification Server Request port = port_num
#Oracle Notification Server Local port = port_num
#Oracle Notification Server Remote port = port_num
#Application Server Control port = port_num
#Application Server Control RMI port = port_num
```

```

#Oracle Management Agent port = port_num
#Web Cache HTTP Listen port = port_num
#Web Cache HTTP Listen (SSL) port = port_num
#Web Cache Administration port = port_num
#Web Cache Invalidation port = port_num
#Web Cache Statistics port = port_num
#Log Loader port = port_num

# Infrastructure

#Oracle Internet Directory port = port_num
#Oracle Internet Directory (SSL) port = port_num
#Oracle Certificate Authority SSL Server Authentication port = port_num
#Oracle Certificate Authority SSL Mutual Authentication port = port_num
#Ultra Search HTTP port number = port_num

```

To use the file:

1. Copy the file from Disk 1 to the ORACLE\_HOME or TMP directory.
2. Edit the file to include the port numbers you want to assign during installation.
3. Provide the path to the file to Oracle Universal Installer during installation.

## B.4 dads.conf File

[Example B–3](#) shows a typical `dads.conf` file for the Single Sign-On Database Access Descriptor in the Identity Management configuration:

### **Example B–3** *dads.conf File*

```

<Location /pls/orasso>
  SetHandler pls_handler
  Order deny,allow
  Allow from All
  AllowOverride None
  PlsqlDatabaseUsername orasso
  PlsqlDatabasePassword @BVXkuI3MPMlyWJArZp1kz4M4RP7rzEr/zQ==
  PlsqlDatabaseConnectString cn=racdb,cn=oraclecontext NetServiceNameFormat
  PlsqlNLSLanguage AMERICAN_AMERICA.UTF8
  PlsqlAuthenticationMode SingleSignOn
  PlsqlSessionCookieName orasso
  PlsqlDocumentTablename orasso.wwdoc_document
  PlsqlDocumentPath docs
  PlsqlDocumentProcedure orasso.wwdoc_process.process_download
  PlsqlDefaultPage orasso.home
  PlsqlPathAlias url
  PlsqlPathAliasProcedure orasso.wwpth_api_alias.process_download
</Location>

```



---

# Index

## A

Active Directory (AD) Synchronization to Oracle Internet Directory, 2-13  
administrator password, OracleAS Web Cache, 7-59, 9-51  
alias, adding, 9-48  
APPDBHOST computers, description, 2-1  
APPHOST computers, description, 2-2  
Application middle tier servers, 2-2  
Application Tier  
    communication, 1-2  
    installing in myBICompany, 8-2  
    installing in myPortalCompany, 7-13  
    installing myJ2EECompany, 6-2, 9-52  
    variants, 2-10, 2-13  
application.log file, 7-40, 7-75  
applications, external, 7-85  
authentication  
    OC4J applications and, 5-15, 6-18  
    OracleAS Single Sign-On, 5-1  
availability, Load Balancing Router tuning and, 4-20

## B

b64InternetCertificate.txt file, 7-37  
base configuration, OracleAS Cluster, 6-12  
best practices, enterprise deployment  
    configuration, 3-1  
Big IP, protecting URLs, 7-86

## C

cacerts file, 7-67, 7-70  
cache cluster (OracleAS Web Cache), 2-14  
Cache Operations page, OracleAS Web Cache Manager, 9-8  
cache.conf file, 7-53  
certificate authority, 9-29  
certificate format, 9-29  
Check Point Firewall-1 NG internal firewall, A-3  
Cisco Pix gateway firewall, A-3  
cleartext password, 7-32  
clocks, synchronization, Oracle Internet Directory and, 4-7  
Clustering page, OracleAS Web Cache Manager, 9-7

Cold Failover Cluster (Identity Management)  
    solution, 2-11  
configMyPortal.xml file, 7-26  
configuration process  
    enterprise deployment architectures, 2-16  
    myBICompany, 8-1  
    myJ2EECompany, 6-1  
    myPortalCompany, 7-1  
connection  
    component and firewall timeout values, 3-2  
    OracleAS Portal, managing, 7-87  
    OracleAS Reports, managing, 8-13  
create user command (SQL\*Plus), 7-28  
custom port assignments, B-2

## D

Data Tier  
    configuration, 4-21  
    variants, 2-10  
database  
    connections, timeout and, 3-2  
    prerequisite for Security infrastructure, 4-1  
    using OCFS file system, 4-5, 7-7  
    using raw devices, 4-3, 7-4  
Database Access Descriptor, dads.conf file, B-3  
data-sources.xml file, 7-29, 7-30, 7-32, 7-56  
DCM Discovery Port, 6-10, 9-53, 9-57  
dcmCache.xml file, 9-53  
DCM-Managed OracleAS Cluster, creating, 6-12  
deploying applications, 6-11  
Distinguished Name (DN), 7-9

## E

enterprise deployment, defined, 1-1  
error\_log file, 7-39, 7-75  
external applications, query path URL, 7-85

## F

F5 Big IP load balancer, A-3  
failover virtual IP addresses, 2-2

## file

- application.log, 7-40, 7-75
- b64InternetCertificate.txt, 7-37
- cacerts, 7-67, 7-70
- cache.conf, 7-53
- configMyPortal.xml, 7-26
- data-sources.xml, 7-29, 7-30, 7-32, 7-56
- dcmCache.xml, 9-53
- error\_log, 7-39, 7-75
- ForwardedHostReplace.pm, 9-36
- hosts, 9-45
- httpd.conf, 9-33, 9-35
- iasconfig.xml, 7-22, 7-55, 9-13, 9-16, 9-46
- jazn-data.xml, 7-33, 7-56
- jazn.xml, 7-83
- mod\_oc4j.conf, 6-17
- oc4j.properties, 8-13
- ons.conf, 6-15
- oradav.conf, 7-53, 8-11
- orion-application.xml, 6-11
- osso.conf, 7-58, 9-42
- provider.xml, 7-30, 7-56
- setdasurl.ldif, 9-41
- sqlnet.ora, 4-6
- sso\_apache.conf, 7-86, 7-87
- staticports.ini, 9-53
- targets.xml, 7-38, 9-14, 9-19, 9-40
- webcache.xml, 7-59, 9-47
- web.xml, 7-54, 9-47, 9-48, 9-49, 9-51

File-based Farm Repository, DCM Discovery port and, 6-9, 9-57

## firewall

- cluster members separated by, 9-6
- communication restrictions and security, 1-2
- disabling external to internal communication, 9-9
- dropped connections and, 3-2
- HTTP requests and port 80, 8-2
- instance communication across, 9-53
- internal and external OracleAS Web Cache instances, 9-6
- invalidation messages and, 9-13
- Oracle Internet Directory and, 2-12
- OracleAS Portal connections and, 7-87
- OracleAS Reports and, 8-13
- OracleAS Web Cache administration and, 9-22
- port 80 and HTTP requests, 7-15, 9-4
- ports open for DCM instances, 9-53
- reverse proxy server and, 9-25
- timeout settings and OracleAS Portal, 7-87
- timeout value and OC4J connection, 6-18

ForwardedHostReplace.pm file, 9-36

## G

grant connect command (SQL\*Plus), 7-28

## H

hardware cluster, 2-11

health monitor, OracleAS Web Cache, 7-12, 7-24, 8-12

## high availability

- enterprise deployment architectures and, 1-2
- Manually-Managed OracleAS cluster and, 7-77

hosts file, 9-45

HTTP, persistent sessions, Load Balancing Router, 4-21, 5-1

httpd.conf file, 9-33, 9-35

## I

iasconfig.xml file, 7-22, 7-55, 9-13, 9-16, 9-46

Identity Management servers, 2-2

## Identity Management Tier

- communication, 1-4

- variants, 2-10, 2-12

Identity Management, configuring, 5-15

idleTimeout attribute, OracleAS Reports Services and, 8-13

IDMHOST computers, description, 2-2

IIS listener, configuring, 9-38

index.html page, customizing, 7-66

indirect password, 7-32

INFRADBHOST computers, description, 2-1

internal load balancer, 2-2

internal server names, hiding, 7-66

Internet Information Services (IIS) listener, as reverse proxy, 9-36

invalidation messages, reverse proxy server and, 9-45

IP Addresses, 2-2

## J

J2EE applications, enterprise deployment architecture, 2-2

## JAAS

- authentication, 5-15

- provider, 6-1, 6-18

JAZN LDAP User Manager, 6-1

jazn-data.xml file, 7-33, 7-56

jazn.xml file, 7-83

JPKD providers, types, 7-76

## K

Kerberos credentials, 2-13

## L

LDAP, internal load balancer, 2-2

LDAP-based provider, OC4J applications

- authentication and authorization, 5-15, 6-18

ldapbind command, 7-26

ldap.cache.session.enable property, 7-84

listener, Net, restarting, 4-6

load balancer, F5 Big IP, A-3

- Load Balancing Router
  - accepting and forwarding requests, 9-9
  - configuring Network Address Translation bounce back, 9-12
  - invalidation requests and, 9-12
  - myJ2EECompany, 6-2
  - OID hosts and, 4-20
- Load Balancing Router, tuning monitoring, 4-20
- load balancing, Manually-Managed OracleAS Cluster and, 7-77
- log files, OracleAS Metadata Repository Creation Assistant, 4-4

## M

- Manually-Managed OracleAS Cluster, OC4J\_JPDK applications and, 7-77
- mapping tablespaces to raw devices, B-1
- metadata repository configuration (file vs. database), 2-13
- Microsoft Active Directory, 2-12
- mod\_oc4j, request routing and, 6-17
- mod\_oc4j.conf file, 6-17
- monitoring
  - OracleAS Portal metrics, 7-36
  - OracleAS Web Cache ports, 7-12
- monitoring OracleAS Portal metrics, 7-63
- multimaster replication, Oracle Internet Directory, 2-10

## N

- Net listener, restarting, 4-6
- Netegrity Siteminder Agent, 2-13
- Network Address Translation (NAT) bounce back, 9-12
- NLS\_LANG environment variable, 4-4

## O

- OC4J applications, authentication and, 5-15, 6-18
- OC4J instances, application tier (myJ2EECompany), OracleAS Clusters and, 6-10
- OC4J JVM default trust store, 7-69
- oc4j.properties file, 8-13
- oidadmin tool, starting, 4-20
- OIDHOST computers, description, 2-1
- oid.mycompany.com, configuring for Load Balancing Router, 4-20
- OmniPortlet, configuring, 7-28
- ons.conf file, 6-15
- Oracle Application Server Java Authentication and Authorization Service (JAAS)
  - provider, 6-1
  - support, 6-18
- Oracle Application Server Java Authentication and Authorization Service (JAAS) Support, 5-15
- Oracle Business Intelligence applications, enterprise deployment architecture, 2-8

- Oracle Internet Directory
  - clocks, 4-7
  - installing, 4-7
  - multimaster replication, 2-10
  - servers, 2-1
- oracle\_sso\_server target type, 9-40
- OracleAS Cold Failover Cluster (Identity Management) solution, 2-11
- OracleAS Metadata Repository
  - installing, 4-1
  - invalidation requests, 9-12
- OracleAS Portal
  - applications, enterprise deployment architecture, 2-5
  - cache, session binding and, 7-63
  - configuring on APPHOST2, 7-49
  - metrics, monitoring, 7-36, 7-63
  - tools providers, 7-28
- OracleAS Proxy Plug-in, IIS listener and, 9-36
- OracleAS Web Cache
  - administrator password, 7-59, 7-64, 9-51
  - cluster members, administrator password and, 7-64
  - clusters, creating, 7-59
  - Manager and cluster configuration, 9-7
  - Manager and propagating configuration, 9-8
  - monitoring, 7-12, 7-24, 8-12
  - ports, monitoring, 7-12
- OracleAS Web Clipping, 7-31
- OracleASPortal
  - database connections, 7-87
- oradav.conf file, 7-53, 8-11
- orapki utility, 7-38
- orion-application.xml file, 6-11
- osso.conf file, 7-58, 9-42

## P

- password
  - cleartext, 7-32
  - indirect, 7-32
  - plain text, encrypting, 9-46
- performance, OracleAS Web Cache and, 2-14
- persistent HTTP sessions, Load Balancing Router and, 4-21, 5-1
- plain text password
  - encrypting, 9-46
- PlsqlIdleSessionCleanupInterval parameter, OracleAS Portal database connections and, 7-87
- pooled connections, timeout and, 3-2
- port
  - assignments, Distributed Configuration Management and firewall, 9-53
  - assignments, enterprise deployment architectures, 2-2
  - default, request redirection, 7-15, 8-2, 9-4
  - ONS, 6-15
- providers, OracleAS Portal Tools, 7-28
- provider.xml file, 7-30, 7-56
- proxy server, OracleAS Web Cache integration, 2-15

proxy, forward and reverse, Oracle HTTP  
Server, 2-15  
ProxyPass directive, 9-33  
ProxyPassReverse directive, 9-33  
ProxyPreserveHost directive, 9-34

## R

---

realm, 7-10  
remote listening port, ONS, 6-15  
replication of session state, 7-77  
repository configuration (file vs. database), 2-13  
reverse proxy, Oracle HTTP Server, 2-15  
RewriteRule directive, 9-34  
round robin load balancing, timeout value and, 4-20

## S

---

security  
    enterprise deployment configurations and, 1-1,  
        1-2  
    firewalls and, 1-2  
    infrastructure, myJ2EECompany, 6-1  
session binding  
    enabling in OracleAS Web Cache, 7-63, 9-23  
session state replication, 7-77  
setdasurl.ldif file, 9-41  
Source Network Address Translation ports, 2-2  
SQLNET.EXPIRE\_TIME parameter, 4-6  
sqlnet.ora file, 4-6  
    file  
        sqlnet.ora, 7-13  
SSL certificate, obtaining, 9-28  
SSL communication, enabling on Oracle HTTP  
Server, 9-27  
SSLConfigTool command, 7-26  
sso\_apache.conf file, 7-86, 7-87  
ssoreg script, executing, 7-34, 8-8, 9-20, 9-50  
standalone instances in OracleAS Farm, 6-12  
state replication  
    JPDK instances, 7-77  
    OracleAS Cluster, 7-77  
Static Ports feature, Oracle Universal Installer, B-2  
staticports.ini file, 9-53  
SunONE Directory Server, 2-12

## T

---

tablespaces, mapping to raw devices, 4-3, 7-4, B-1  
targets.xml file, 7-38, 9-14, 9-19, 9-40  
timeout, 7-87  
    Load Balancing Router, 4-20  
    values, Oracle Application Server components and  
        firewall/load balancer, 3-2  
trust store, OC4J JVM, 7-69

## U

---

upgrade, OracleAS File-based Farm and, 2-14  
URL prefix, OracleAS Single Sign-On, 7-85  
user names, mapping for external applications, 7-85

UTL\_HTTP package, 7-85

## V

---

variants. enterprise deployment architectures, 2-10  
virtual IP addresses, 2-2

## W

---

Web Clipping Studio, session binding and, 7-63  
Web Tier  
    communication, 1-2  
    myJ2EECompany, 6-13, 9-54  
    servers, 2-2  
    variants, 2-10, 2-14  
webcache.xml file, 7-59, 9-47  
WEBHOST computers, description, 2-2  
web.xml file, 7-54, 9-47, 9-48, 9-49, 9-51  
Welcome page, modifying or substituting, 7-66  
Windows native authentication, 2-13

## X

---

x509certfile, 9-51