Oracle® Application Server

Enterprise Deployment Guide

10*g* Release 2 (10.1.2) for Windows or UNIX

Part No. B13998-01

December 2004

ORACLE®

Oracle Application Server Enterprise Deployment Guide, 10*g* Release 2 (10.1.2) for Windows or UNIX

Part No. B13998-01

Primary Authors: Janga Aliminati, Peter Lubbers, Julia Pond, Greg Sowa, Tim Willard

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

Contents

A Sample Configurations for Certified Load Balancers

B Sample Files and Values

Index

Send Us Your Comments

Oracle Application Server Enterprise Deployment Guide, 10*g* Release 2 (10.1.2) for Windows or UNIX

Part No. B13998-01

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

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

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

- Electronic mail: appserverdocs_us@oracle.com
- FAX: 650.506.7375 Attn: Oracle Application Server Documentation Manager
- Postal service:

  Oracle Corporation
  Oracle Application Server Documentation
  500 Oracle Parkway, MS 1op6
  Redwood Shores, CA 94065
  USA

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

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

Preface

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.

Structure

This guide contains the following chapters and appendixes:

Chapter 1, "Overview"

This chapter describes the Enterprise Deployment topology, including the configuration and functionality of the components within the tiers.

Chapter 2, "Installing and Configuring the Security Infrastructure"

This chapter provides instructions on installing and configuring the components in the Security infrastructure.

In the myPortalCompany architecture, the Security Infrastructure has two tiers: a Data tier, on which Oracle Internet Directory and Directory Integration and Provisioning reside, and an Identity Management Tier, on which Oracle Application Server Single Sign-On and Oracle Delegated Administration Services reside.

In the myJ2EECompany architecture, the Security Infrastructure consists only of the Data Tier, because authentication is provided by the Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider and not OracleAS Single Sign-On.

Chapter 3, "Configuring the Application Infrastructure for myJ2EECompany.com"

This chapter provides instructions on installing and configuring the middle tier components in the myJ2EECompany.com Application infrastructure. Components in the Application Infrastructure are distributed between two tiers: a Web Server tier, on which OracleAS Web Cache and Oracle HTTP Server reside, and an Application Tier, containing the Oracle Application Server Containers for J2EE applications.

Chapter 4, "Configuring the Application Infrastructure for myPortalCompany.com"

This chapter provides instructions on installing and configuring the middle tier components in the myPortalCompany.com Application infrastructure. Components in

the Application Infrastructure are distributed between two tiers: a Web Server tier, on which OracleAS Web Cache resides and an Application Tier, on which Oracle HTTP Server, OracleAS Portal, and OracleAS Wireless reside.

Appendix A, "Sample Configurations for Certified Load Balancers"

This appendix contains sample configurations for load balancers and firewalls that are certified by Oracle for use with Oracle products.

Appendix B, "Sample Files and Values"

This appendix contains sample configuration files and values.

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 Installation Guide*
- *Oracle Internet Directory Administrator's Guide*
- *Oracle Application Server Single Sign-On Administrator's Guide*
- *Oracle Application Server Concepts*
- *Oracle Application Server Administrator's Guide*

Conventions

The following conventions are also used in this manual:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
boldface text	Boldface type in text indicates a term defined in the text, the glossary, or in both locations.
`monospace italic`	Variables whose value must be supplied by the user are shown in monospace italic text.
[ ]	Brackets enclose optional clauses from which you can choose one or none.
/	Used in all directory paths; assumed equivalent to the backward slash convention for Windows.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over

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

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

Accessibility of 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.

1

Overview

This chapter introduces the Oracle Application Server Enterprise Deployment configurations. It contains the following topics:

1.1 What is an Enterprise Deployment?

An enterprise deployment is an Oracle Application Server configuration that supports large-scale, mission-critical business software applications. The hardware and software in an enterprise deployment 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 A Standard Enterprise Deployment for J2EE Applications: myJ2EECompany.com

Figure 1–1 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 1.3.

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 OracleAS Web Cache and 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 1–1, Table 1–2 and Table 1–3 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 1–1 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

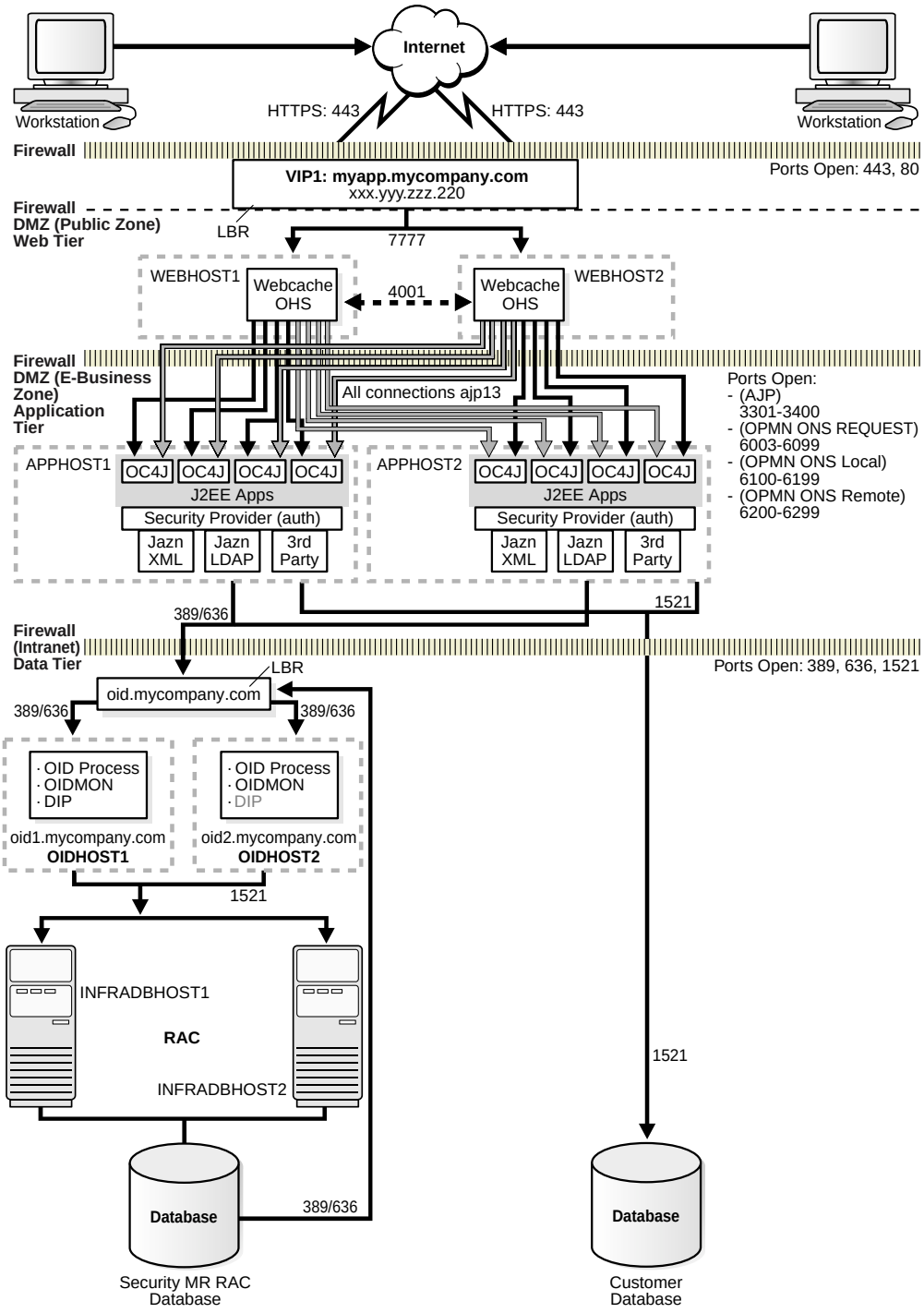
Table 1–2 myJ2EECompany Hardware Requirements (Linux)

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 1–3 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 1–1 Enterprise Deployment Architecture for myJ2EECompany.com



1.3 A Standard Enterprise Deployment for Portal Applications: myPortalCompany.com

Figure 1–2 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.

Table 1–4, Table 1–5 and Table 1–6 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 1–7 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 1–4 myPortalCompany Hardware Requirements (Windows)

Server	Processor	Disk	Memory	TMP Directory	Swap
APPHOST, IDMHOST, 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 1–5 myPortalCompany Hardware Requirements (Linux)

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

Table 1–6 myJ2EECompany Hardware Requirements (Solaris)

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

Figure 1–2 Enterprise Deployment Architecture for myPortalCompany.com

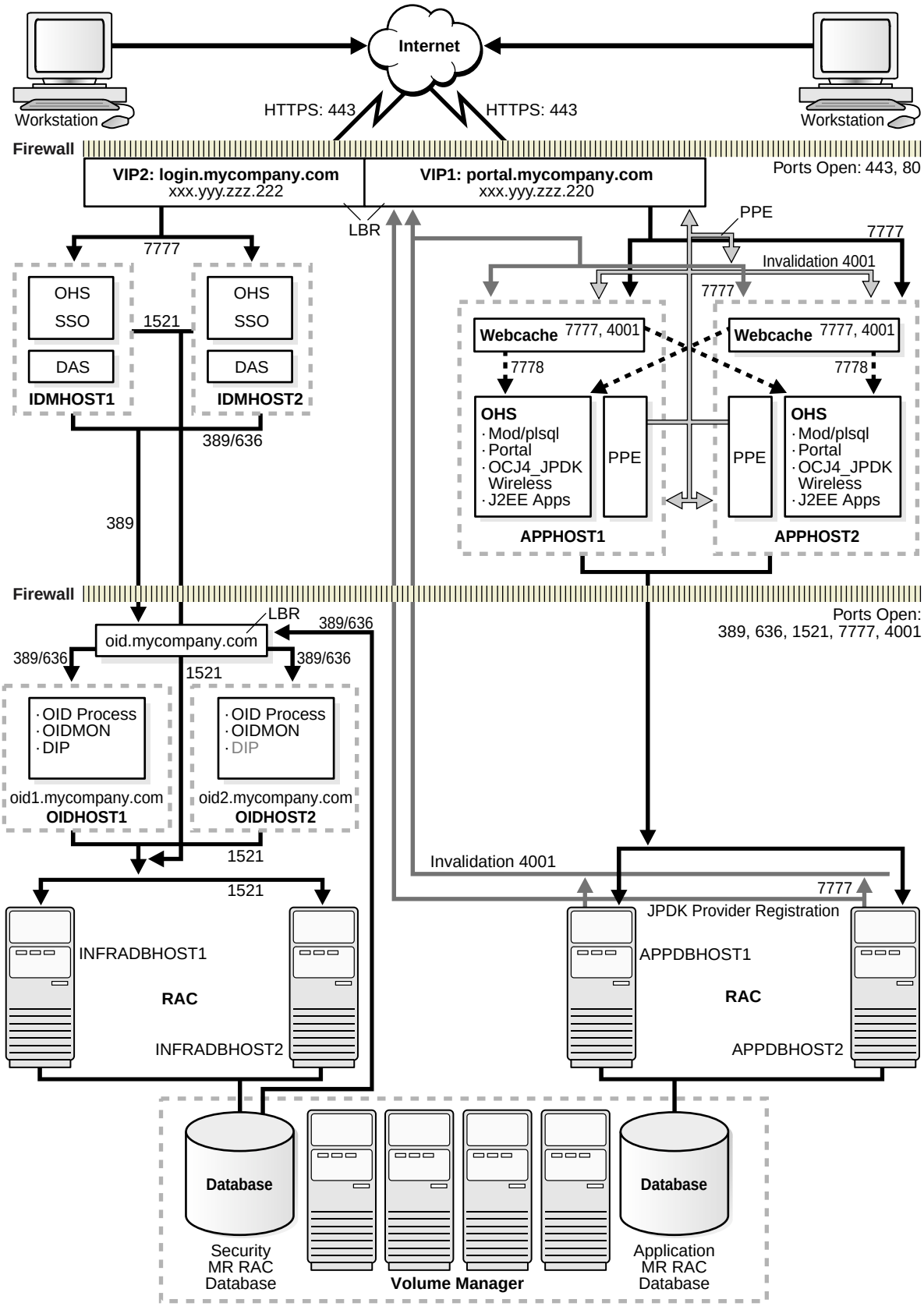


Table 1–7 myPortalCompany Servers in Oracle Test Environment

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

1.4 Benefits of the Standard Enterprise Topology

The Oracle Application Server configurations shown in myJ2EECompany.com and myPortalCompany.com are designed to ensure security of all transactions, maximize hardware resources, and provide a reliable, standards-compliant system for enterprise computing. This section explains how the security and high availability benefits of the two configurations are achieved.

1.4.1 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.
- 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.

1.4.2 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 4001, and balances the requests to one of the OracleAS Web Cache listeners on port 4001. In these cases, the Load Balancing Router functions as a proxy to receive internal requests on port 4001, 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 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.

4. The OracleAS Portal Parallel Page Engine also loops back to the Load Balancing Router (through the internal Network Address Translation port) to reach mod_plsql 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 mod_plsql 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 mod_plsql 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 mod_plsql 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.

1.5 Variants to the Standard Enterprise Deployment Configurations

Figure 1–1, "Enterprise Deployment Architecture for myJ2EECompany.com" and Figure 1–2, "Enterprise Deployment Architecture for myPortalCompany.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 1–2), or the Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider (on the Application Tier in Figure 1–1), 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.

1.5.1 Understanding Data Tier Variants

This section describes the variants for the Data Tier. The Data Tier is depicted in Figure 1–2, "Enterprise Deployment Architecture for myPortalCompany.com", and comprises the INFRADBHOST1 and INFRADBHOST2 computers.

1.5.1.1 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".

1.5.1.2 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.

1.5.1.2.1 Using 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)".

1.5.2 Understanding Identity Management Tier Variants

This section describes the variants for the Identity Management Tier. The Identity Management Tier is depicted in Figure 1–2, "Enterprise Deployment Architecture for myPortalCompany.com", and comprises the IDMHOST1 and IDMHOST2 computers.

1.5.2.1 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.

1.5.2.2 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.

1.5.2.3 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`

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`

1.5.2.4 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.

1.5.3 Understanding Application Tier Variants

This section describes the variants for the Application Tier. The Application Tier is depicted in Figure 1–2, "Enterprise Deployment Architecture for myPortalCompany.com", and comprises the APPHOST1 and APPHOST2 computers.

1.5.3.1 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 1–8 compares repository types in light of operational considerations.

Table 1–8 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

1.5.4 Understanding Web Server Tier Variants

This section describes the variants for the Web Server Tier. The Web Server Tier is depicted in Figure 1–2, "Enterprise Deployment Architecture for myPortalCompany.com", and comprises the APPHOST1 and APPHOST2 computers.

In Figure 1–1, "Enterprise Deployment Architecture for myJ2EECompany.com", the Web Server Tier comprises the WEBHOST1 and WEBHOST2 computers.

1.5.4.1 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 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.*

1.5.4.2 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 more information on how to configure Oracle HTTP Server for these environments, see the *Oracle HTTP Server Administrator's Guide.*

For information on how to integrate OracleAS Web Cache with an additional proxy server, see the *Oracle Application Server Web Cache Administrator's Guide.*

1.6 Enterprise Deployment Nomenclature

The naming convention for the components and computers is established in Figure 1–1 and Figure 1–2, and is used throughout this guide. Server names, and their related URLs and IP addresses are provided in Table 1–9. The external load balancer nomenclature is provided in Table 1–10.

Table 1–9 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.xxxx.xxx.225
	INFRADBHOST2	infradbhost2.mycompany.com	xxx.xxxx.xxx.226
Servers with 2-node Real Application Clusters database for Application Metadata Repository	APPDBHOST1	appdbhost1.mycompany.com	xxx.xxxx.xxx.227
	APPDBHOST2	appdbhost2.mycompany.com	xxx.xxxx.xxx.228
Oracle Internet Directory servers	OIDHOST1	oidhost1.mycompany.com	xxx.xxx.xxx.229
	OIDHOST2	oidhost2.mycompany.com	xxx.xxx.xxx.230
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 1–10 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

1.7 How to Use This Guide: The Enterprise Deployment Configuration Process

This guide is organized to reflect the chronology of the installation and configuration process for the myJ2EECompany and myPortalCompany architectures. The configuration process for each is detailed in the following sections.

1.7.1 Installing and Configuring myJ2EE Company

1. Install the Metadata Repository on INFRADBHOST1 and INFRADBHOST2.

2. Install Oracle Internet Directory on OIDHOST1 and OIDHOST2.
3. Install an Oracle Application Server J2EE and Web Cache installation on APPHOST1 and APPHOST2. Configure OC4J, and disable OracleAS Web Cache and Oracle HTTP Server.
4. Create OC4J instances in the Oracle Application Server instances on APPHOST1 and APPHOST2, and deploy applications on the instances.
5. Create a DCM-Managed Oracle Application Server Cluster and add the instances to it.
6. Install an Oracle Application Server J2EE and Web Cache installation on WEBHOST1 and WEBHOST2. Configure OracleAS Web Cache and Oracle HTTP Server, and disable OC4J.
7. Configure the Load Balancing Router.
8. Configure the Oracle HTTP Server with the Load Balancing Router.
9. Configure OC4J routing.
10. Configure application authentication and authorization with the Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider.
11. (Optional) Configure Secure Sockets Layer for Oracle HTTP Server, OracleAS Web Cache, OC4J and mod_oc4J.

1.7.2 Installing and Configuring myPortalCompany

1. Install the Metadata Repository on INFRADBHOST1 and INFRADBHOST2.
2. Install Oracle Internet Directory on OIDHOST1 and OIDHOST2.
3. Install Oracle Delegated Administration Services, Oracle Application Server Single Sign-On and Oracle HTTP Server on IDMHOST1 and IDMHOST2.
4. Install OracleAS Web Cache, OracleAS Portal, and Oracle HTTP Server on APPHOST1.
5. Configure the Load Balancing Router and related components.
6. Install OracleAS Web Cache, OracleAS Portal, and Oracle HTTP Server on APPHOST2
7. Configure application server components and the Load Balancing Router on APPHOST2.
8. Configure OracleAS Web Cache and the Load Balancing Router.

1.8 Best Practices for Installing and Configuring Enterprise Deployments

Observation of 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.)

2

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 1–1, "Enterprise Deployment Architecture for myJ2EECompany.com" on page 1-4 and Figure 1–2, "Enterprise Deployment Architecture for myPortalCompany.com" on page 1-6.

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 2.5, "Installing the Identity Management Tier Components 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:

2.1 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 2.1.1.
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 2.1.2 or Section 2.1.3.
   - To install into a database using raw devices, follow the steps in Section 2.1.2, "Installing the Metadata Repository in a Database Using Raw Devices" on page 2-3.
   - To install into a database using Oracle Cluster File System, follow the steps in Section 2.1.3, "Installing the Metadata Repository in an Oracle Cluster File System (OCFS)" on page 2-5.
4. Perform the post-installation step described in Section 2.1.4.

2.1.1 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.

2.1.2 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** 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 Connection** 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*.

2.1.3 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** 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 Connection** 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, writeable 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.*

2.1.4 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.

2.2 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.

2.2.1 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. For detailed instructions, see the *Oracle Application Server Installation Guide*, Requirements chapter, section titled "Checking if a Port is in Use", in the documentation library for the platform you are using.
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:

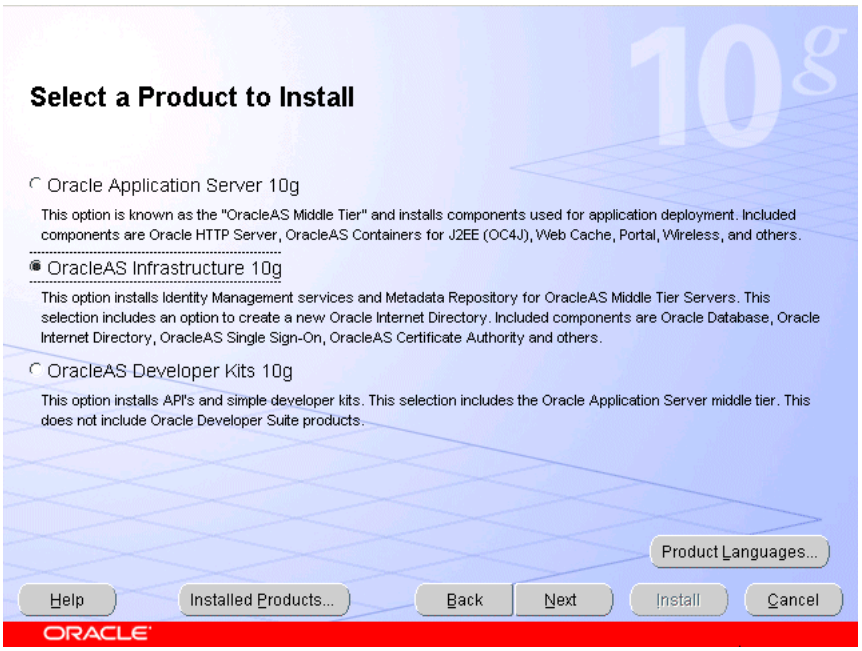
    - The product files for the installation (Source)
    - The name and path to an Oracle home (Destination)

    > **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 2–1 Oracle Universal Installer Select a Product to Install Screen



12. Select OracleAS Infrastructure 10***g***, as shown in Figure 2–1, and click **Next**.

The **Select Installation Type** screen appears.

13. Select **Identity Management**, as shown in Figure 2–2, and click **Next**.

Figure 2–2 Oracle Universal Installer Select Installation Type Screen

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

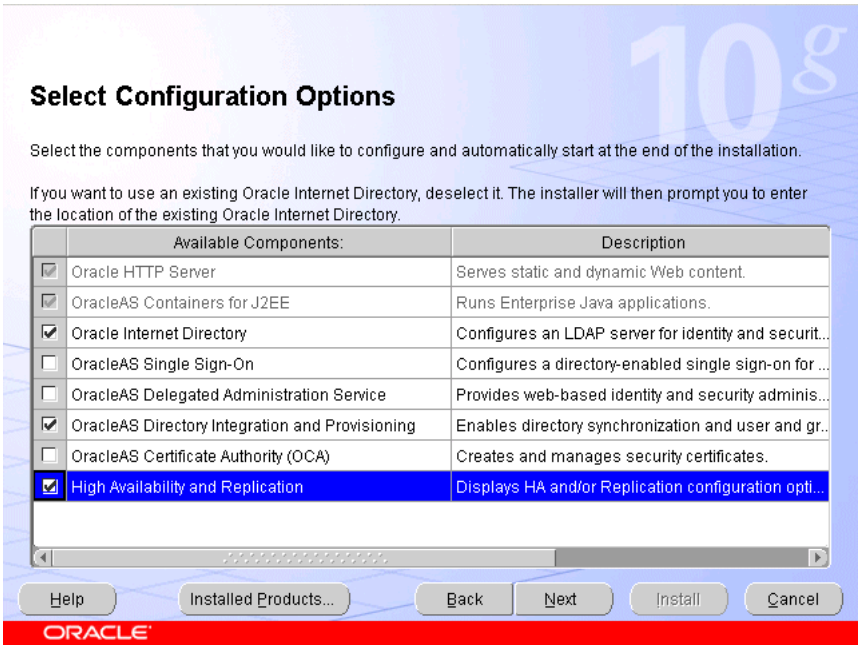
14. Click **Next**.

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.

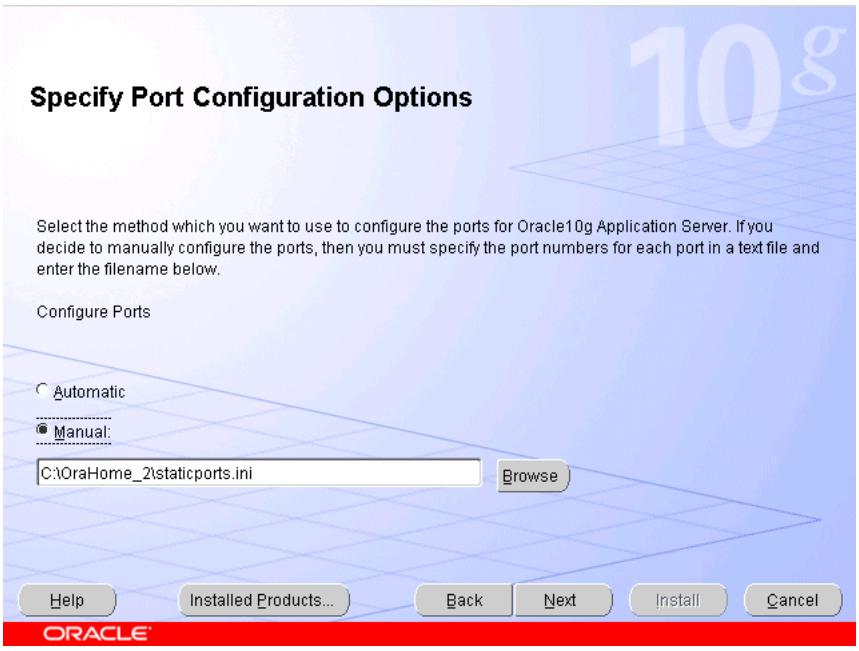
Figure 2–3 Oracle Universal Installer Select Configuration Options Screen



16. Select **Oracle Internet Directory**, **OracleAS Directory Integration and Provisioning**, and **High Availability and Replication**, as shown in Figure 2–3, and click **Next**.

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

Figure 2–4 Oracle Universal Installer Specify Port Configuration Options Screen

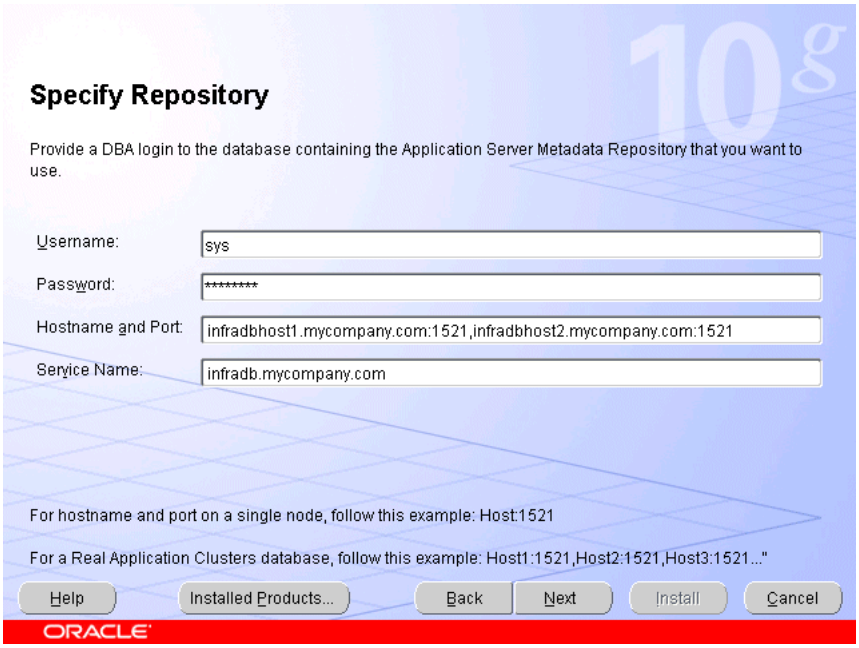


17. Select **Manual**, as shown in Figure 2–4, and click **Next**.

    The **Specify Repository** screen appears.

18. Provide the DBA login and computer information as shown in Figure 2–5 and click **Next**.

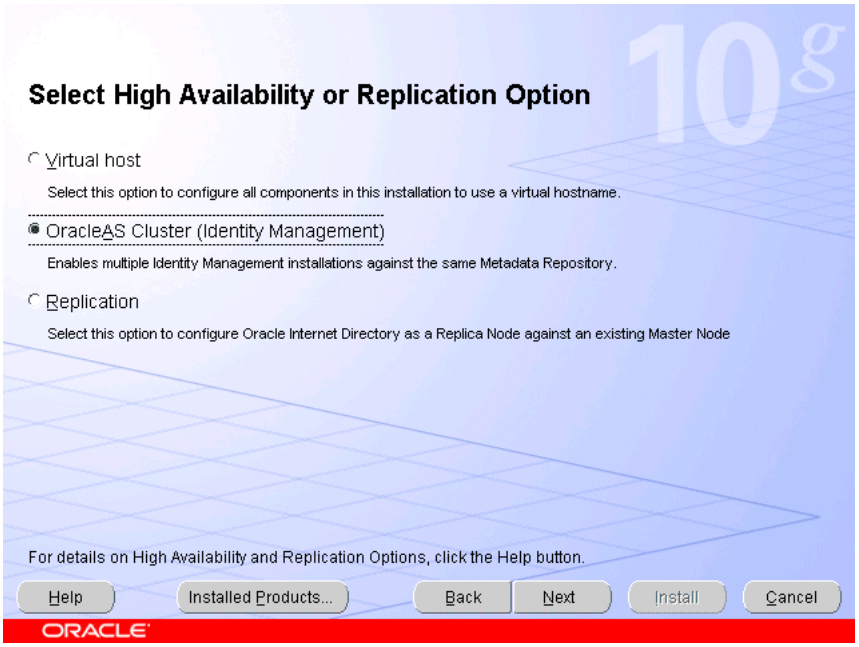
Figure 2–5 Oracle Universal Installer Specify Repository Screen



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

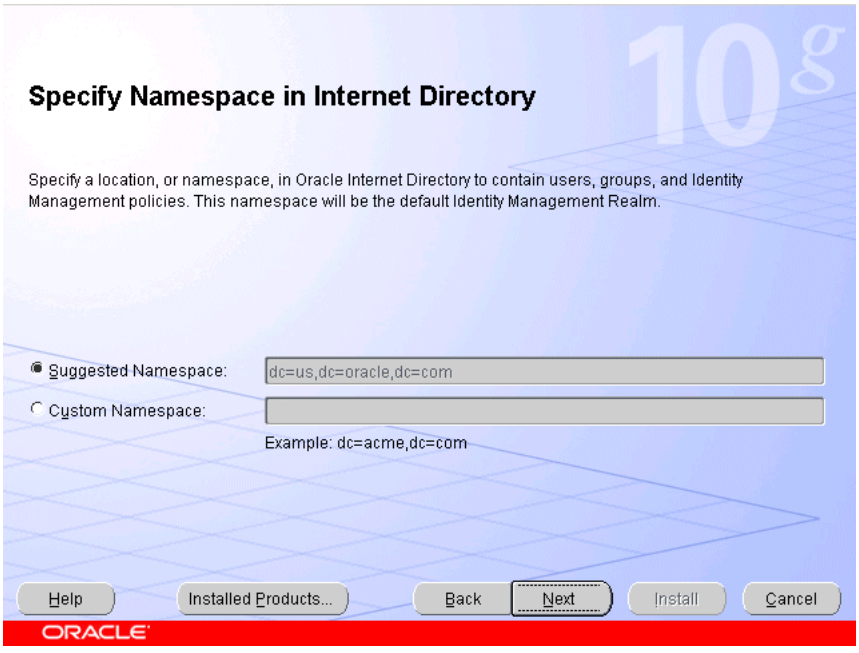
19. Select **OracleAS Cluster (Identity Management)**, as shown in Figure 2–6, and click **Next**.

Figure 2–6 Oracle Universal Installer Select High Availability or Replication Option Screen



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

Figure 2–7 Oracle Universal Installer Specify Namespace in Internet Directory



20. Click **Next** to specify the default **Suggested Namespace** shown in Figure 2–7, 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.

2.2.2 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. For detailed instructions, see the *Oracle Application Server Installation Guide*, Requirements chapter, section titled "Checking if a Port is in Use", in the documentation library for the platform you are using.
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:

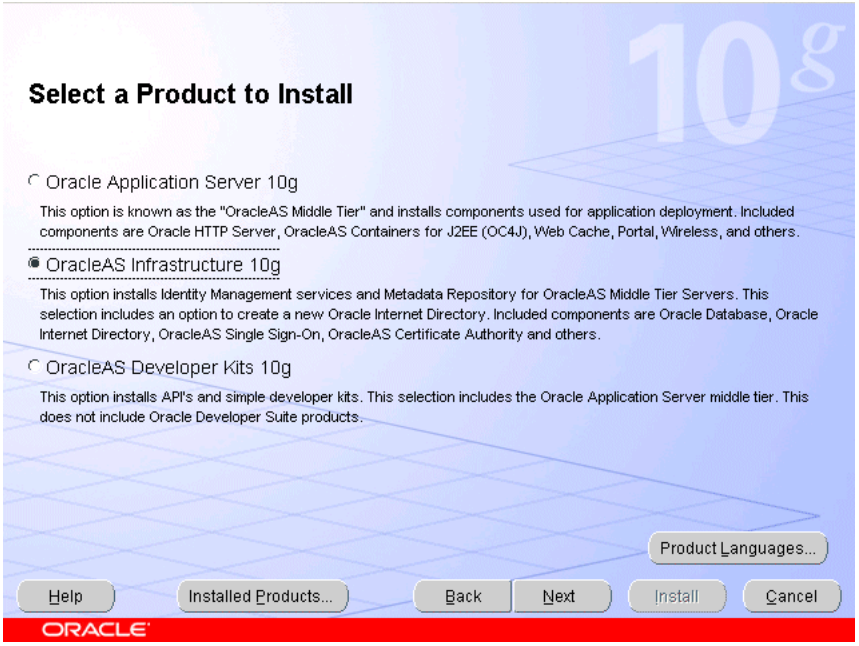
    `/u01/app/oracle/product/AS10gOID`

    ---

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

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

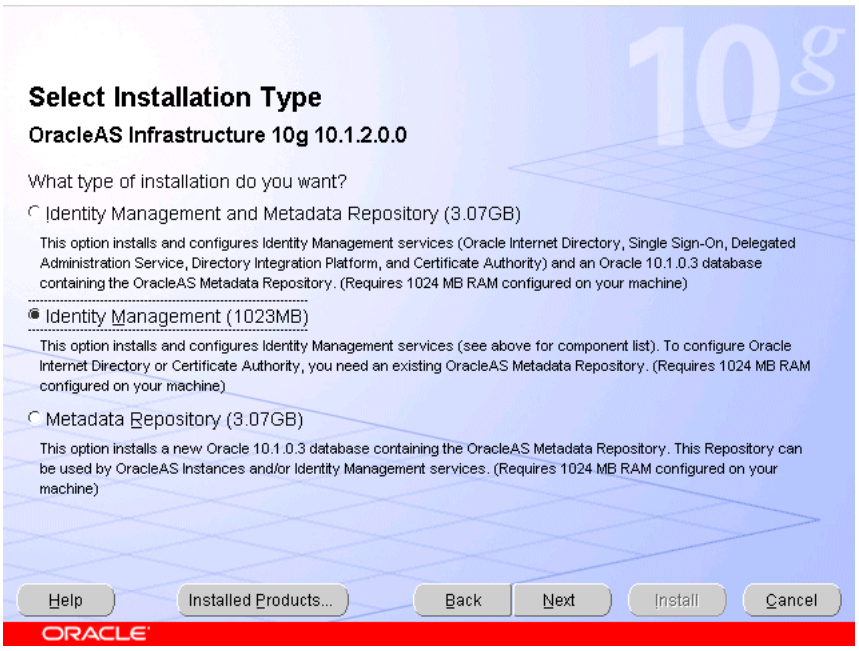
Figure 2–8 Oracle Universal Installer Select a Product to Install Screen



12. Select OracleAS Infrastructure 10***g***, as shown in Figure 2–8, and click **Next**.

    The **Select Installation Type** screen appears.
13. Select **Identity Management**, as shown in Figure 2–9, and click **Next**.

Figure 2–9 Oracle Universal Installer Select Installation Type Screen



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

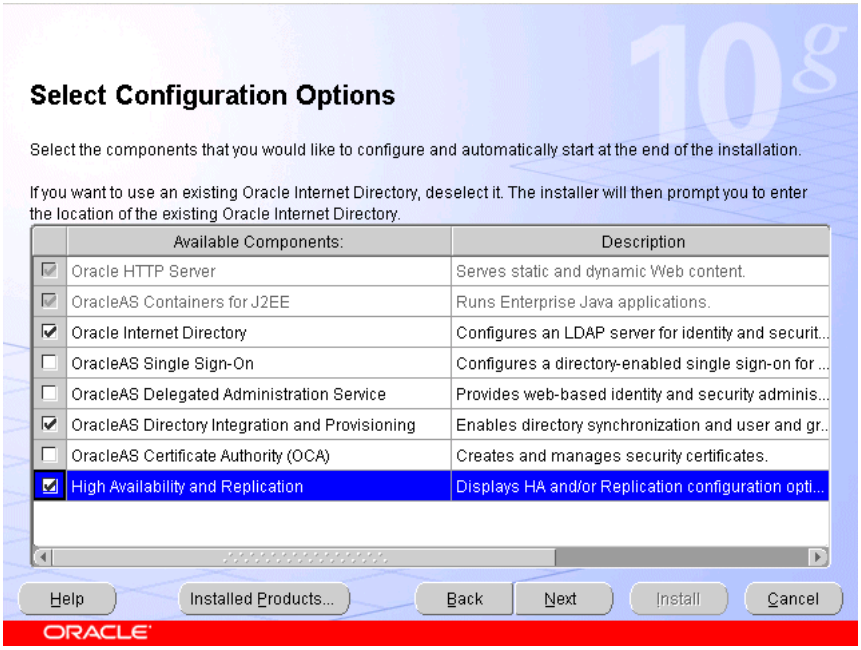
14. Click **Next**.

    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.

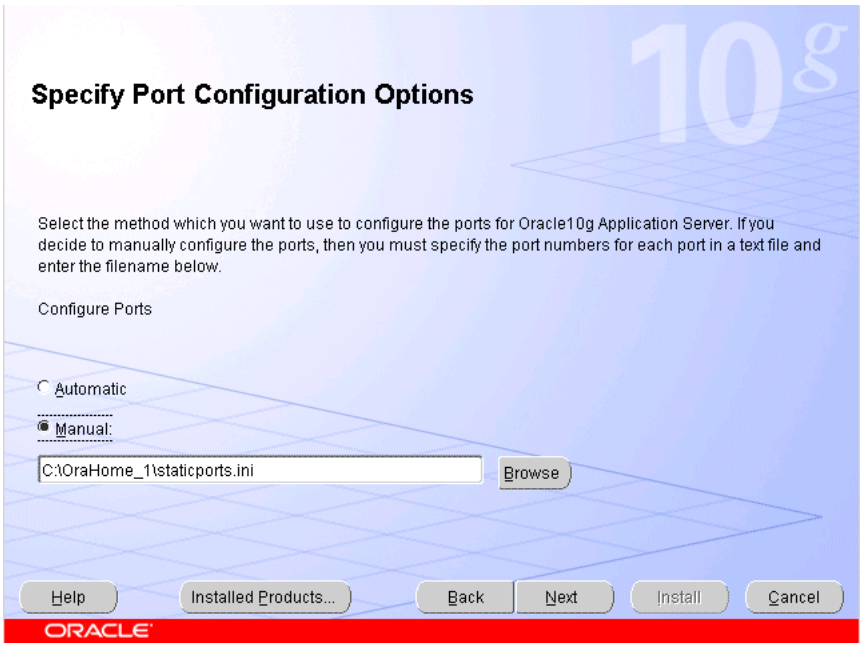
Figure 2–10 Oracle Universal Installer Select Configuration Options Screen



16. Select **Oracle Internet Directory**, **OracleAS Directory Integration and Provisioning**, and **High Availability and Replication**, as shown in Figure 2–10, and click **Next**.

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

Figure 2–11 Oracle Universal Installer Specify Port Configuration Options Screen

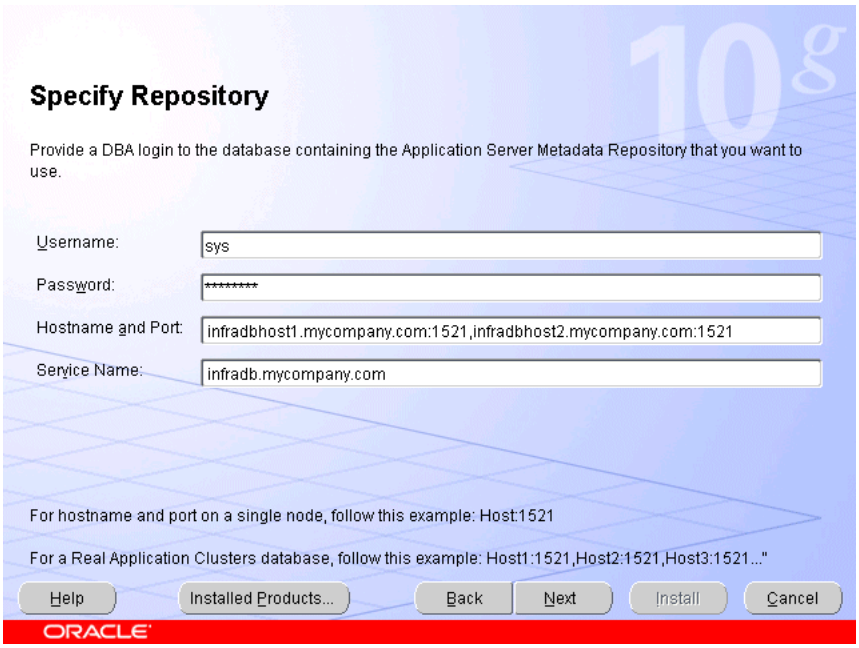


17. Select **Manual**, as shown in Figure 2–11, and click **Next**.

    The **Specify Repository** screen appears.

18. Provide the DBA login and computer information as shown in Figure 2–12 and click **Next**.

Figure 2–12 Oracle Universal Installer Specify Repository Screen



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.

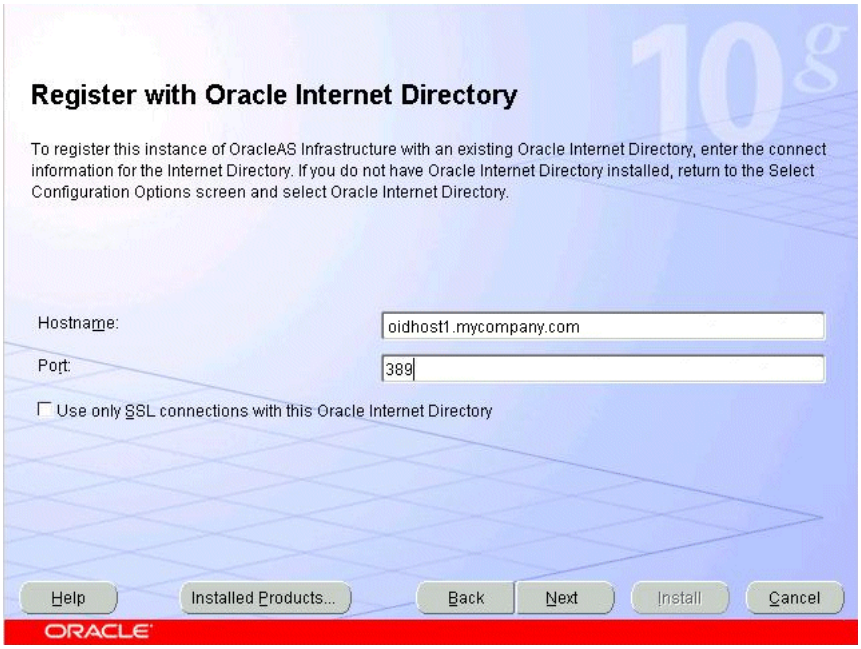
20. Specify the ODS password (by default, the ias_admin password) as shown in Figure 2–13 and click **Next**.

Figure 2–13 Oracle Universal Installer Specify ODS Password Screen

The **Register with Oracle Internet Directory** screen appears.

21. Specify the host name and port, as shown in Figure 2–14, and click **Next**.

Figure 2–14 Oracle Universal Installer Register with Oracle Internet Directory Screen



The **Specify OID Login** screen appears.

22. Specify the user name and password, as shown in Figure 2–15, and click **Next**.

Figure 2–15 Oracle Universal Installer Specify OID Login Screen

The **Specify Instance Name and ias_admin Password** screen appears.

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

The **Summary** screen appears.

24. 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.

25. 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.

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

2.3 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.
- 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.

2.4 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
   ```

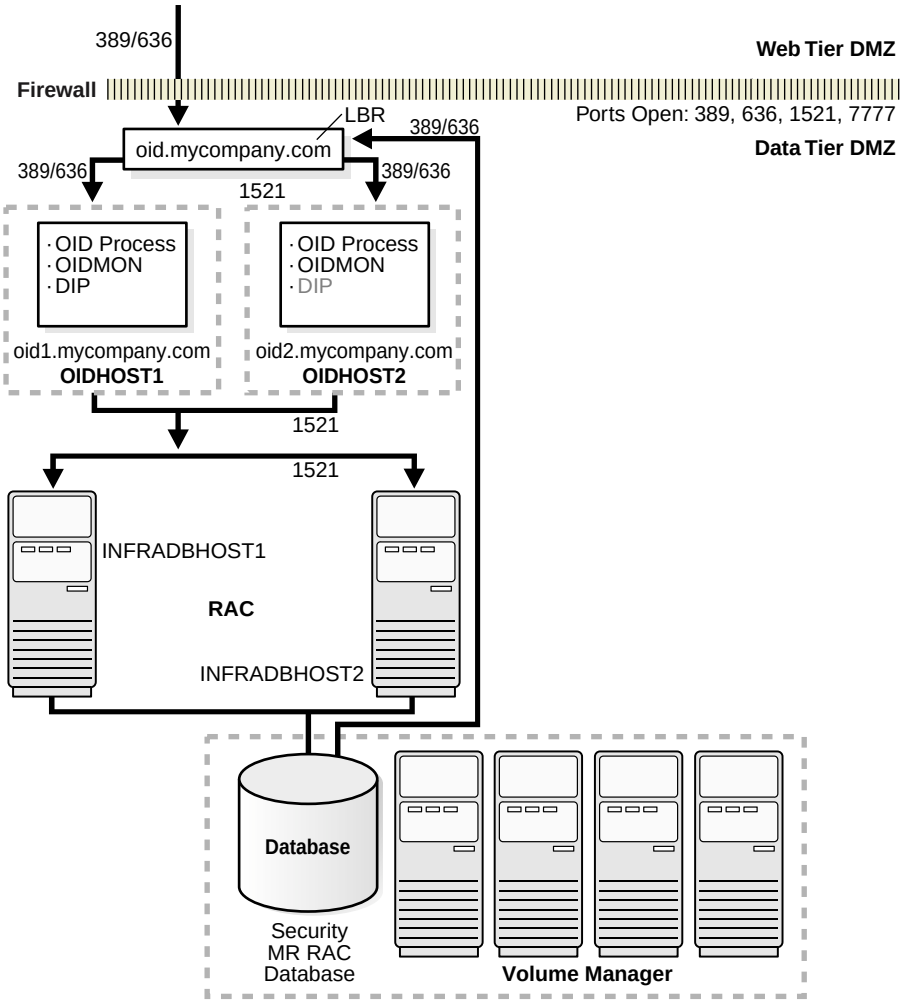
   ```
   ldapbind -p 389 -h oid.mycompany.com
   ```

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 2–16.

Figure 2–16 Data Tier Configuration



2.5 Installing the Identity Management Tier Components for myPortalCompany.com

If you are creating a Security Infrastructure for the myPortalCompany configuration shown in Figure 1–2, "Enterprise Deployment Architecture for myPortalCompany.com" on page 1-6, you must configure Identity Management components. Do not perform the steps in this section if you are configuring myJ2EECompany.

Follow these steps to install the Identity Management components (IDMHOST1 and IDMHOST2) into the Web tier on APPHOST1, after the Data Tier is complete.

Note: You must configure the Load Balancing Router (login.mycompany.com) shown in Figure 2–33, "Identity Management Tier Configuration"for persistent HTTP sessions.

2.5.1 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**.

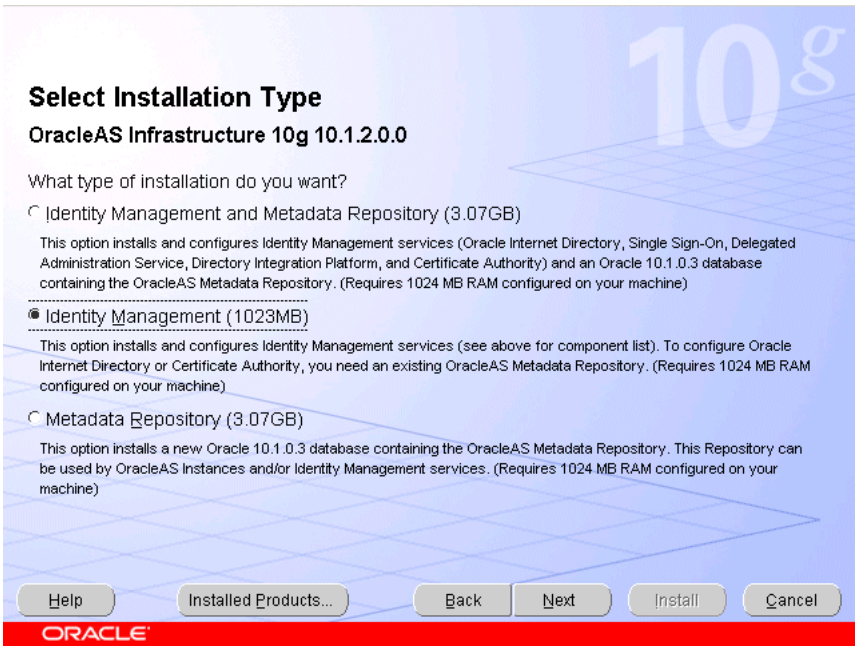
    The **Select a Product to Install** screen appears.

Figure 2–17 Oracle Universal Installer Select a Product to Install Screen

11. Select OracleAS Infrastructure 10***g***, as shown in Figure 2–17, and click **Next**.

The **Select Installation Type** screen appears.

Figure 2–18 Oracle Universal Installer Select Installation Type Screen



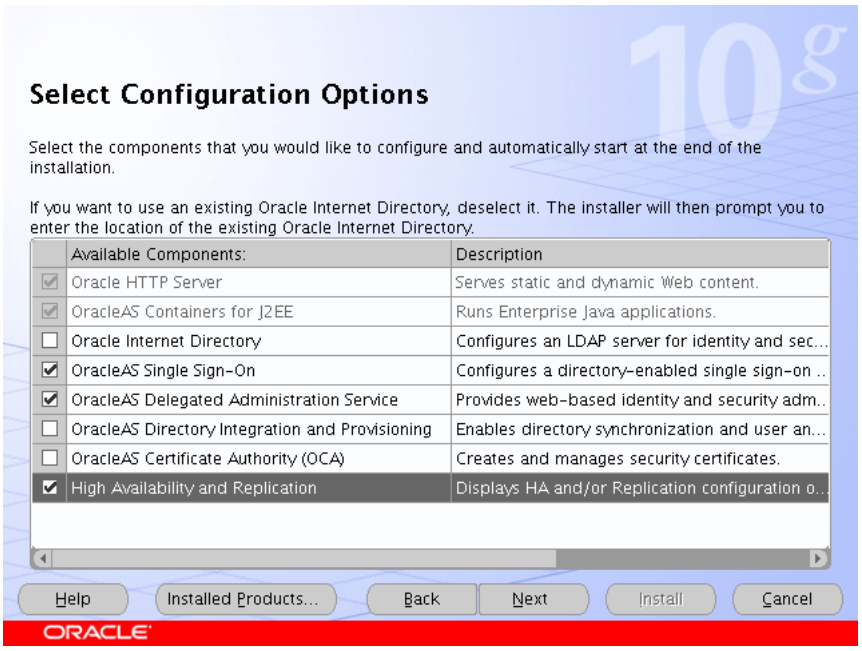
12. Select **Identity Management**, as shown in Figure 2–18, and click **Next**.

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

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

The **Select Configuration Options** screen appears.

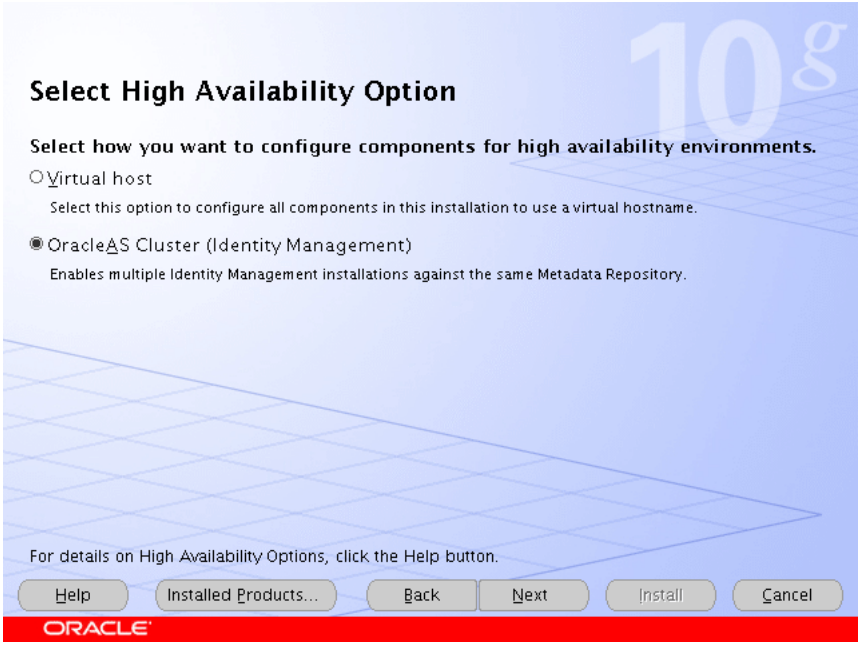
Figure 2–19 Oracle Universal Installer Select Configuration Options Screen



14. Select **OracleAS Single Sign-On, Oracle Delegated Administration Services**, and **High Availability and Replication**, as shown in Figure 2–19.
15. Click **Next**.

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

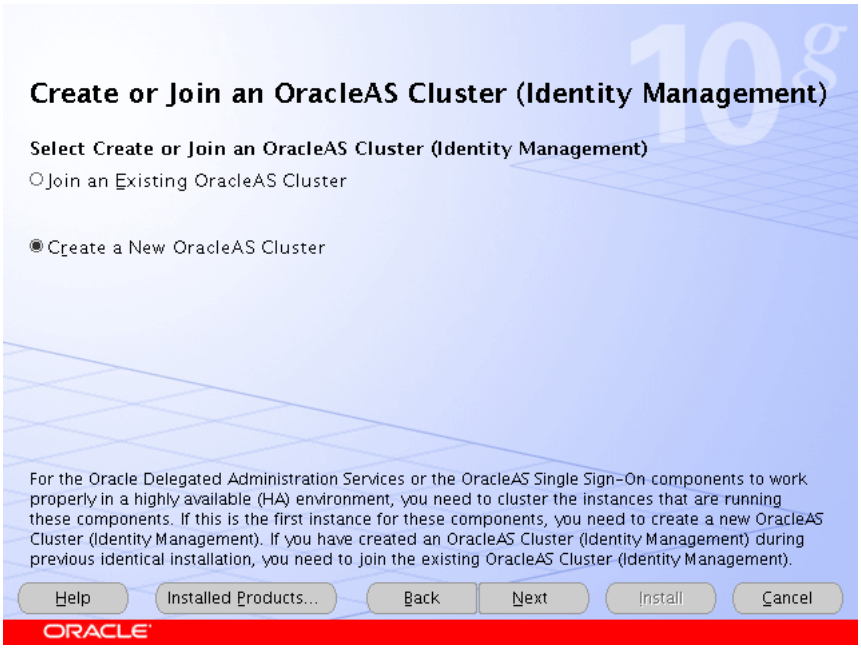
Figure 2–20 Oracle Universal Installer Select High Availability Option Screen



16. Select **OracleAS Cluster (Identity Management)**, as shown in Figure 2–20, and click **Next**.

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

Figure 2–21 Oracle Universal Installer Create or Join an OracleAS Cluster (Identity Management) Screen



17. Select **Create a New OracleAS Cluster**, as shown in Figure 2–21, and click **Next**.

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

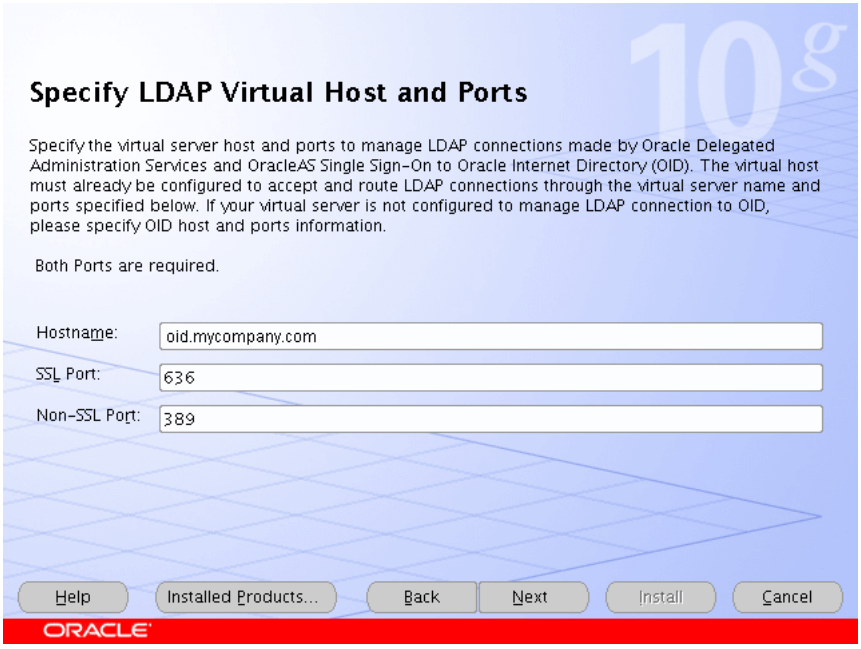
Figure 2–22 Oracle Universal Installer Specify New OracleAS Cluster Name Screen

18. Complete the **New OracleAS Cluster Name** field with a name for the cluster, as shown in Figure 2–22, 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 2–23 Oracle Universal Installer Specify LDAP Virtual Host and Ports Screen

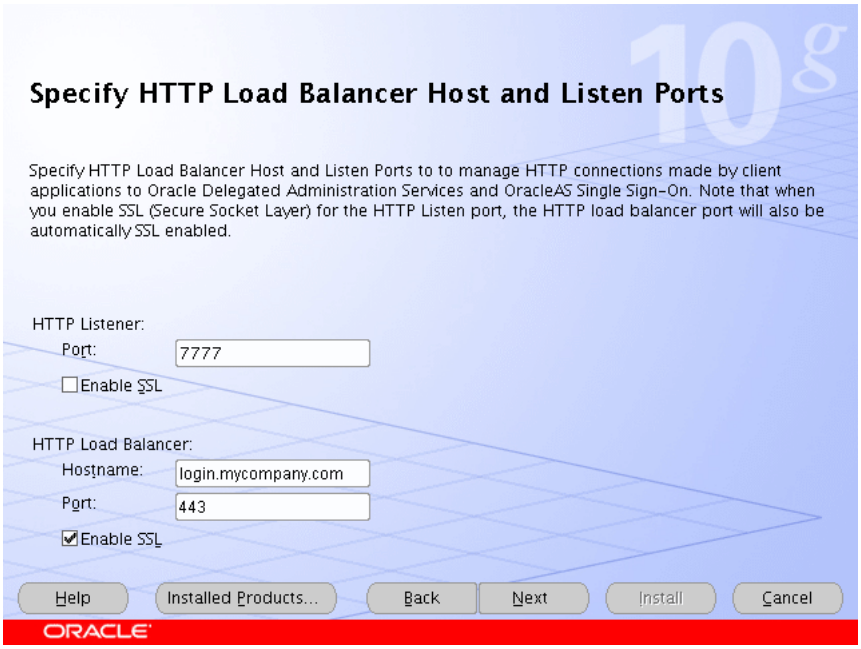


19. Enter the name of the Load Balancing Router, the SSL port, and the non-SSL port, as shown in Figure 2–23.
20. Click **Next**.

    The **Specify OID Login** screen appears.
21. Complete the fields and click **Next**.

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

Figure 2–24 Oracle Universal Installer Specify HTTP Load Balancer Host and Listen Ports Screen



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 2–24.

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.

2.5.2 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`

2.5.3 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**.

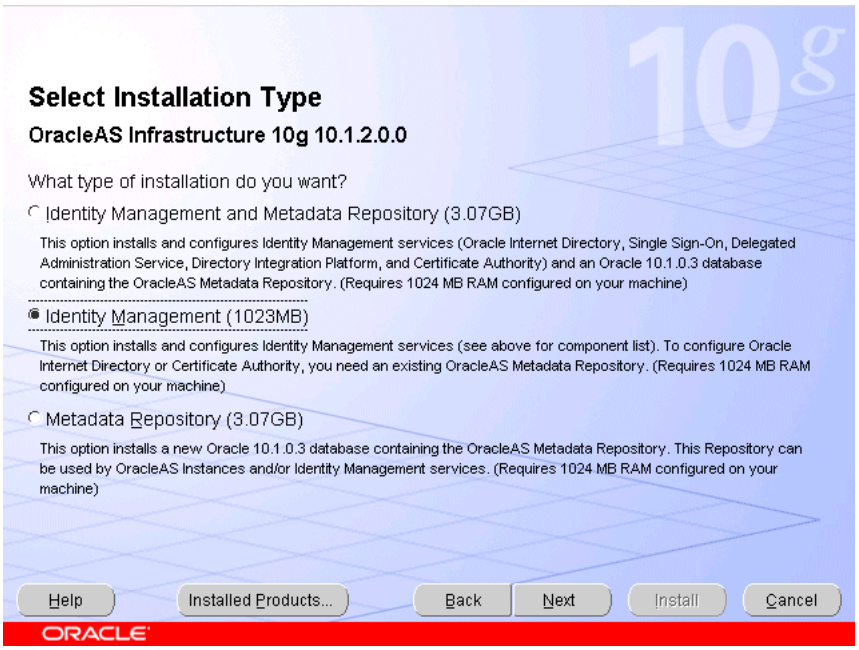
    The **Select a Product to Install** screen appears.

Figure 2–25 Oracle Universal Installer Select a Product to Install Screen

11. Select OracleAS Infrastructure 10***g***, as shown in Figure 2–25, and click **Next**.

    The **Select Installation Type** screen appears.

Figure 2–26 Oracle Universal Installer Select Installation Type Screen



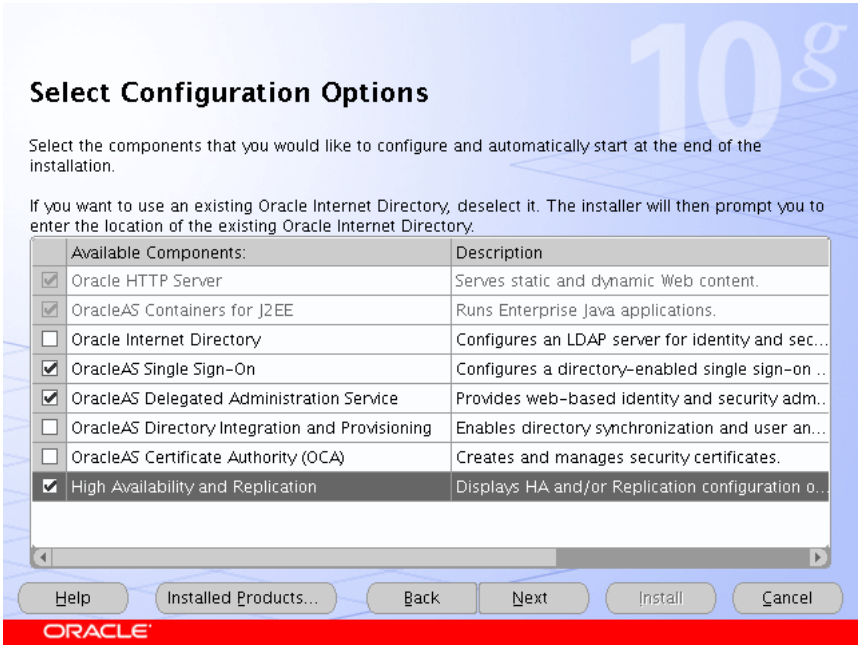
12. Select **Identity Management** as shown in Figure 2–26, and click **Next**.

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

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

    The **Select Configuration Options** screen appears.

Figure 2–27 Oracle Universal Installer Select Configuration Options Screen

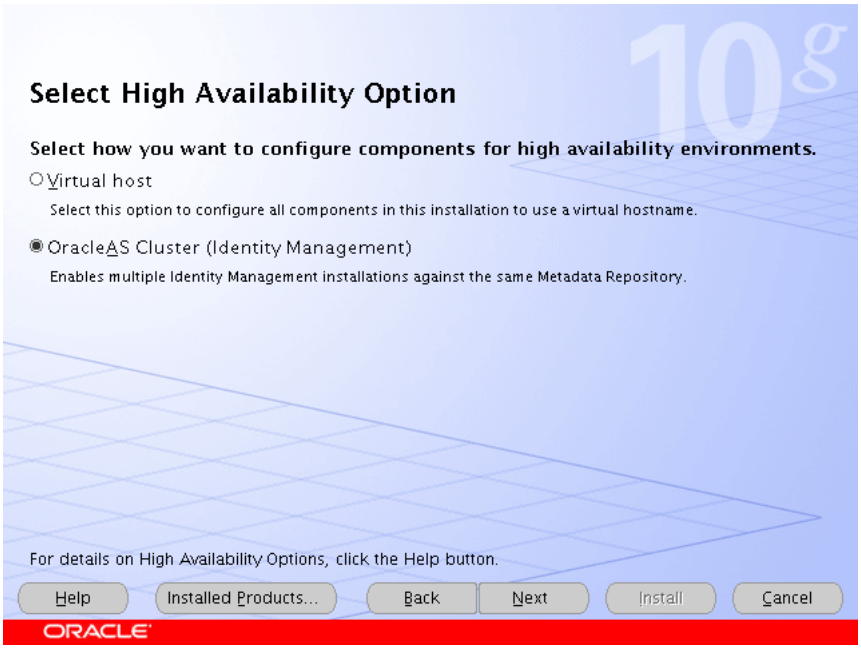


14. Select **OracleAS Single Sign-On, Oracle Delegated Administration Services**, and **High Availability and Replication**, as shown in Figure 2–27.

15. Click **Next**.

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

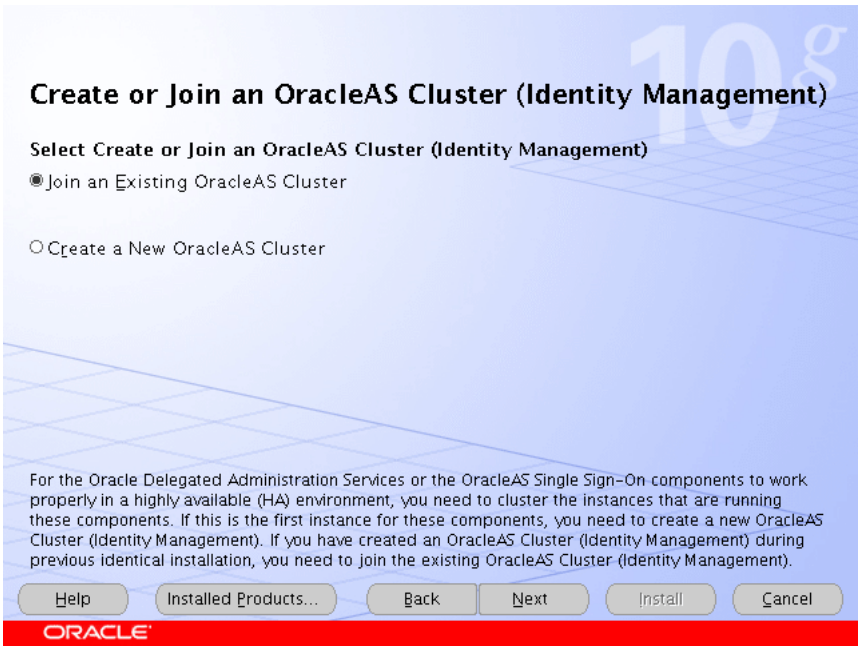
Figure 2–28 Oracle Universal Installer Select High Availability Option Screen



16. Select **OracleAS Cluster (Identity Management)**, as shown in Figure 2–28, and click **Next**.

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

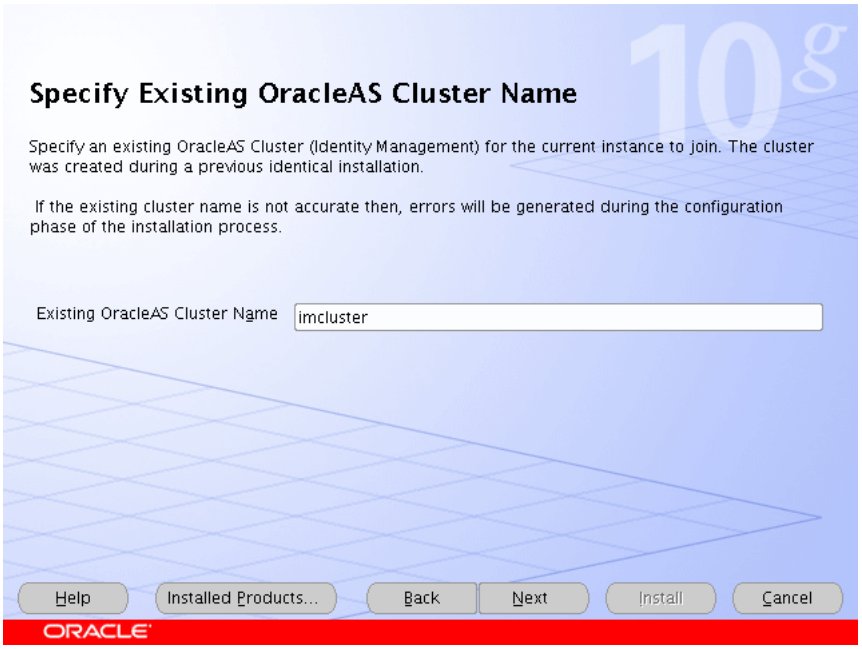
Figure 2–29 Oracle Universal Installer Create or Join an OracleAS Cluster (Identity Management) Screen



17. Select **Join an Existing OracleAS Cluster**, as shown in Figure 2–21, and click **Next**.

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

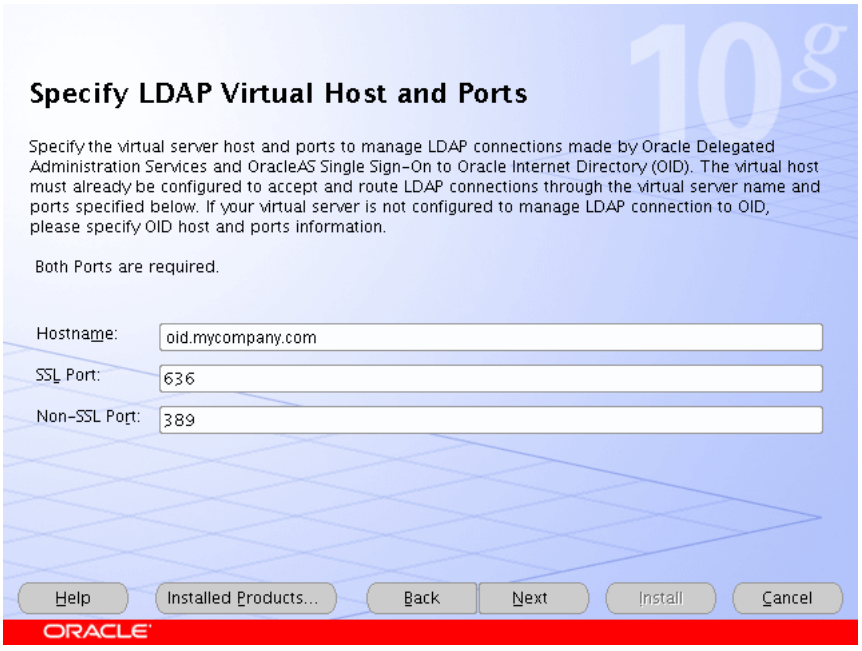
Figure 2–30 Oracle Universal Installer Specify Existing OracleAS Cluster Name Screen



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 2–22, and click **Next**.

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

Figure 2–31 Oracle Universal Installer Specify LDAP Virtual Host and Ports Screen

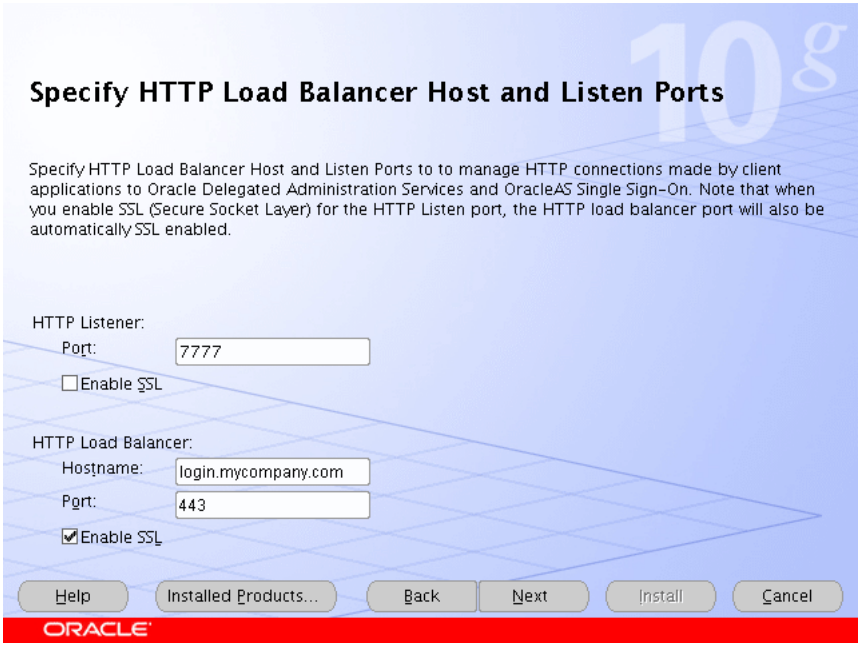


19. Enter the name of the Load Balancing Router, the SSL port, and the non-SSL port, as shown in Figure 2–23.
20. Click **Next**.

    The **Specify OID Login** screen appears.
21. Complete the fields and click **Next**.

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

Figure 2–32 Oracle Universal Installer Specify HTTP Load Balancer Host and Listen Ports Screen



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 2–32.
23. Click **Next**.

    The **Specify Instance Name and ias_admin Password** screen appears.
24. Specify the instance name and password and click **Next**.

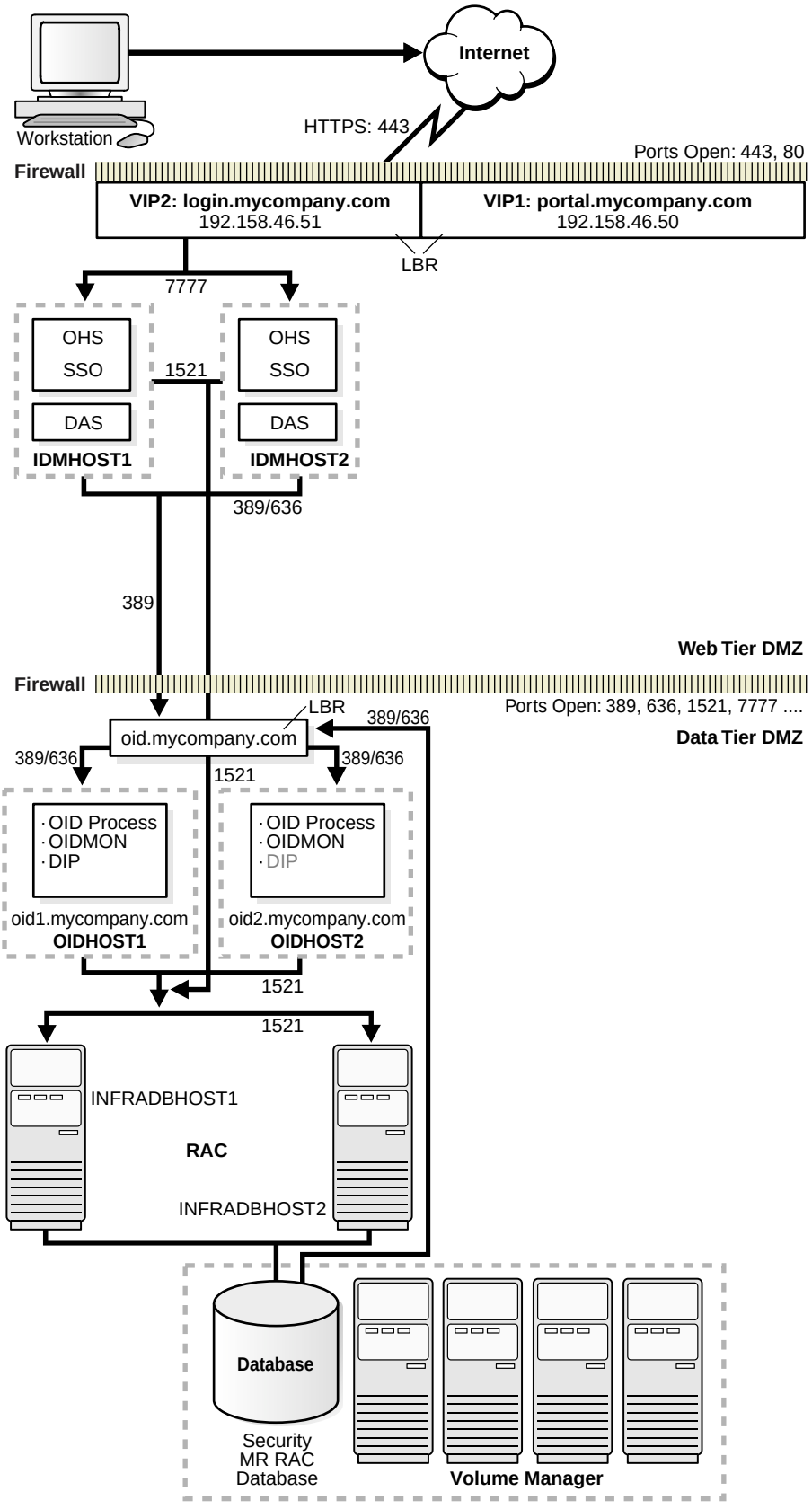
    The **Summary** screen appears.
25. Review the selections to ensure that they are correct (if they are not, click **Back** to modify selections on previous screens), and click **Install**.

    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 2–33.

Figure 2–33 Identity Management Tier Configuration



2.6 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.

3

Configuring the Application Infrastructure for myJ2EECompany.com

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 1–1 on page 1-4.

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:

3.1 Installing and Configuring the Security Infrastructure

The security infrastructure for myJ2EECompany contains the components depicted in Figure 2–16, "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 10*g* Application Server Control Console.

To install and configure this security infrastructure:

1. Follow all instructions in Section 2.1, "Installing the Oracle Application Server Metadata Repository for the Security Infrastructure" on page 2-1.
2. Follow all instructions in Section 2.2, "Installing the Oracle Internet Directory Instances in the Data Tier" on page 2-6.
3. Follow all instructions in Section 2.3, "Configuring the Virtual Server to Use the Load Balancing Router" on page 2-18.

4. Follow all instructions in Section 2.4, "Testing the Data Tier Components" on page 2-18.

3.2 Installing and Configuring the Application Tier

The application tier consists of multiple computers hosting middle tier Oracle Application Server instances, which contain multiple Oracle Application Server Containers for J2EE instances and deployed applications. In the complete configuration, requests are balanced among the OC4J instances on the application tier computers to create a performant, fault tolerant, and secure application environment. Figure 1–1, "Enterprise Deployment Architecture for myJ2EECompany.com" on page 1-4, shows the application tier (APPHOST1 and APPHOST2).

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 inTable 3–1 by default when the instance is installed.

Table 3–1 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 *ORACLE_HOME*`/dcm/config/dcmCache.xml` file, in the `discoverer` element (for example, `<discoverer discovery-port ="7100" original-"true" xmlns=""/>`
50	Range of ports for inter-instance communication: 7120 to 7179. These are defined in the *ORACLE_HOME*`/dcm/config/dcmCache.xml` file, in the port element (for example, `<port lower="7120" upper="7179">`.) 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: ■ 1 for the Oracle Enterprise Manager 10*g* Application Server Control Console on each instance ■ 1 for the DCM daemon on each instance ■ 1 for each dcmctl client operating on each instance

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 3–1.

- After installation of all instances, configure the firewall to close the unused ports within the assigned range on each instance.

3.2.2 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
Web Cache HTTP Listen Port = 7777
Web Cache HTTP Administration Port = 4000
Web Cache HTTP Invalidation Port = 4001
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**.

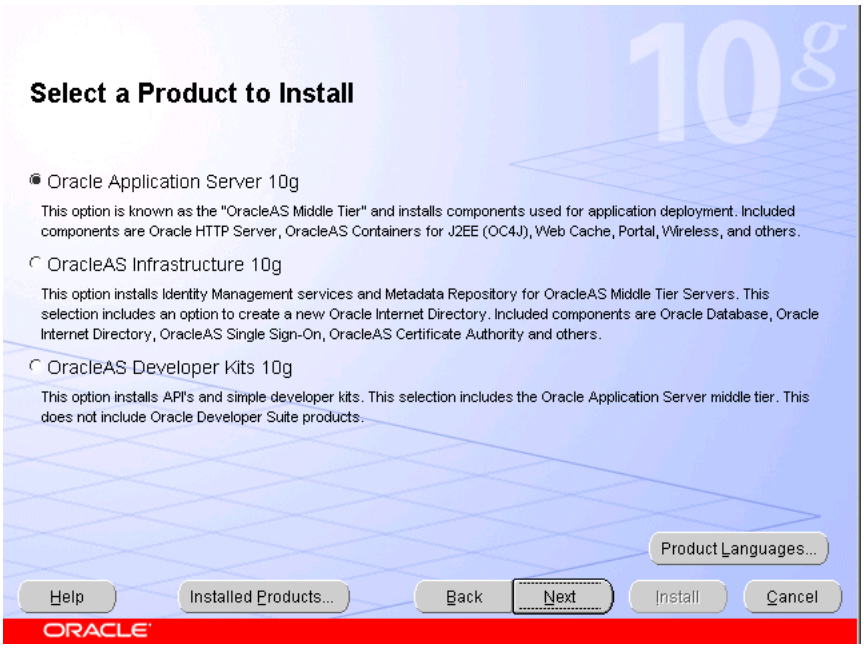
The **Specify File Locations** screen appears with default locations for:

- 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.

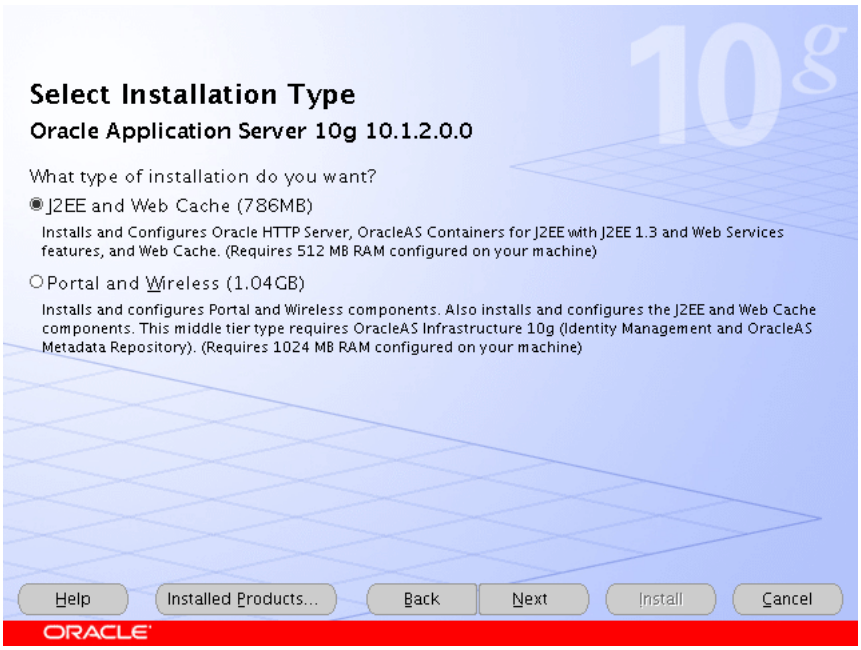
Figure 3–1 Oracle Universal Installer Select a Product to Install Screen



11. Select **Oracle Application Server 10*g***, as shown in Figure 3–1, and click **Next**.

    The **Select Installation Type** screen appears.

Figure 3–2 Oracle Universal Installer Select Installation Type Screen



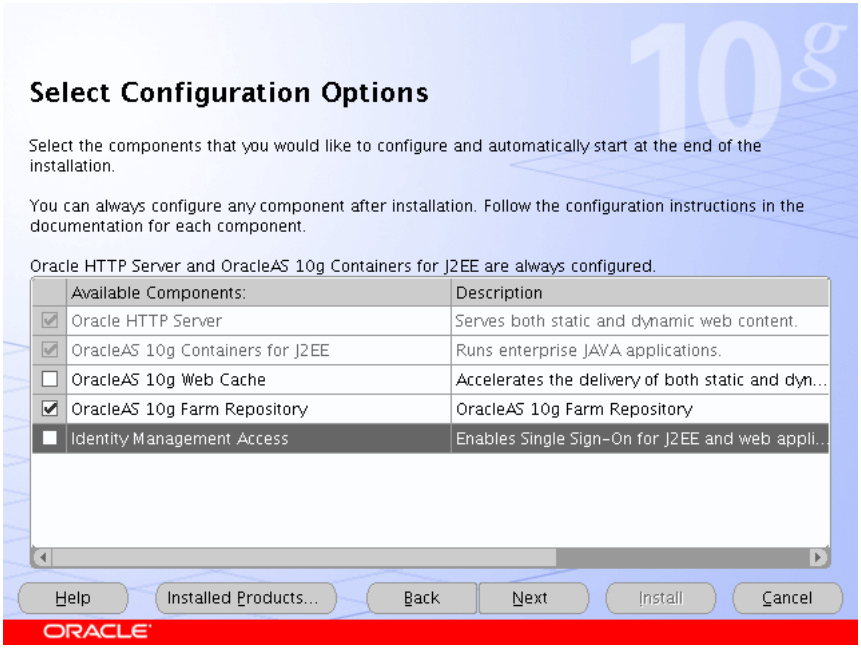
12. Select **J2EE and Web Cache**, as shown in Figure 3–2, and click **Next**.

    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 3–3 Oracle Universal Installer Select Configuration Options Screen



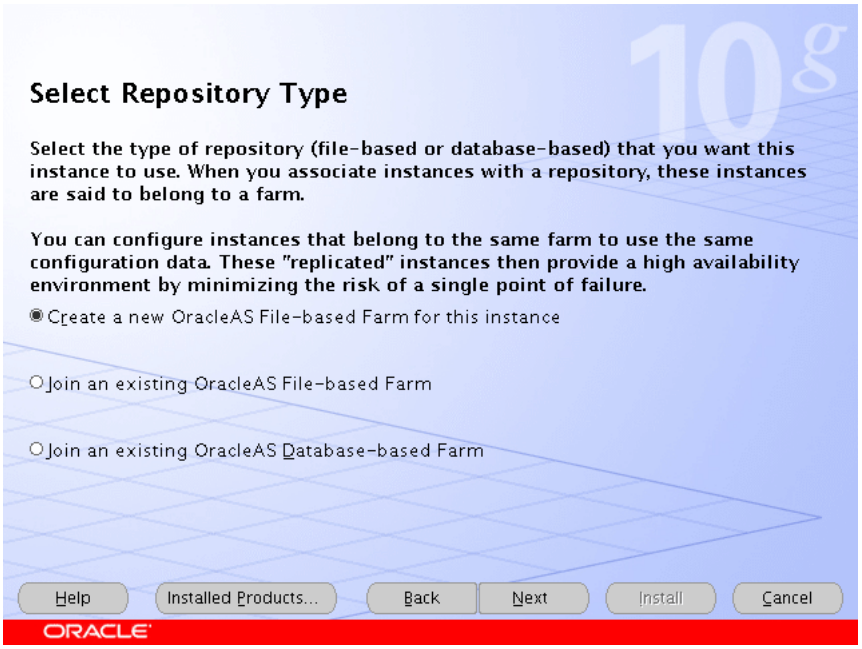
15. Select **OracleAS 10*g* Farm Repository**, as shown in Figure 3–3, and click **Next**.

    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 3–4 Oracle Universal Installer Select Repository Type Screen



17. Select **Create a new OracleAS File-based Farm for this instance**, as shown in Figure 3–4, and click **Next**.

    The **Specify Instance Name and ias_admin Password** screen appears.

18. Specify an instance name and the OracleAS 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 10*g*. Start a browser and access **`http://hostname:1810`**.

3.2.3 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
Web Cache HTTP Listen Port = 7777
Web Cache HTTP Administration Port = 4000
Web Cache HTTP Invalidation Port = 4001
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**.

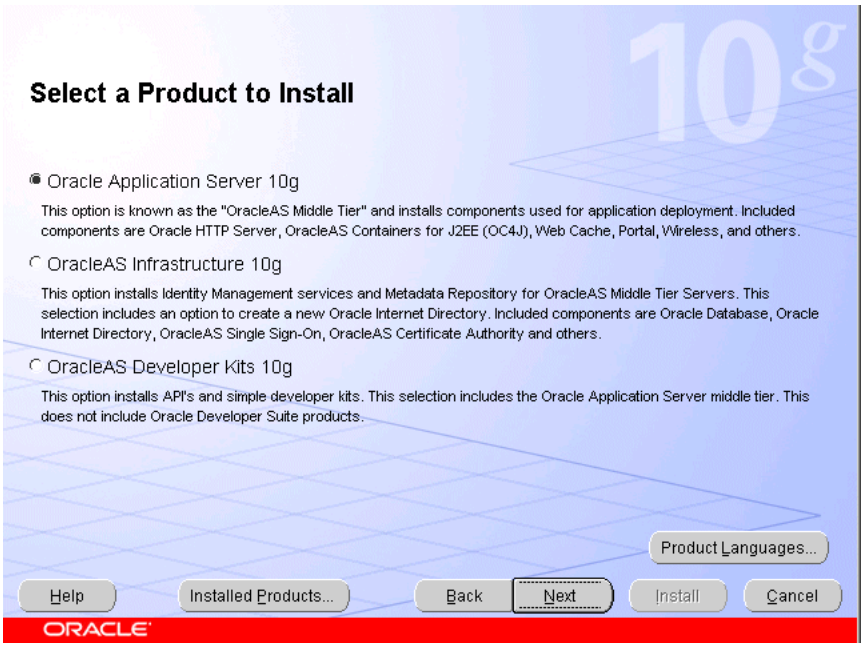
   On UNIX systems, a dialog appears, prompting you to run the `orainstRoot.sh` script.
8. Open a window and run the script, following the prompts in the window.
9. Return to the Oracle Universal Installer screen and click **Next**.

   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.

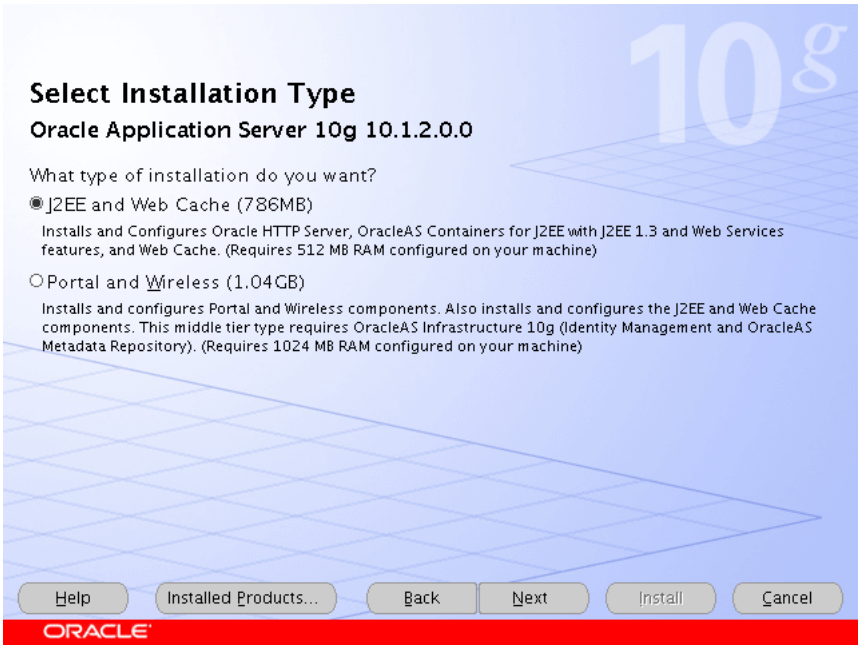
Figure 3–5 Oracle Universal Installer Select a Product to Install Screen



11. Select **Oracle Application Server 10*g***, as shown in Figure 3–5, and click **Next**.

    The **Select Installation Type** screen appears.

Figure 3–6 Oracle Universal Installer Select Installation Type Screen



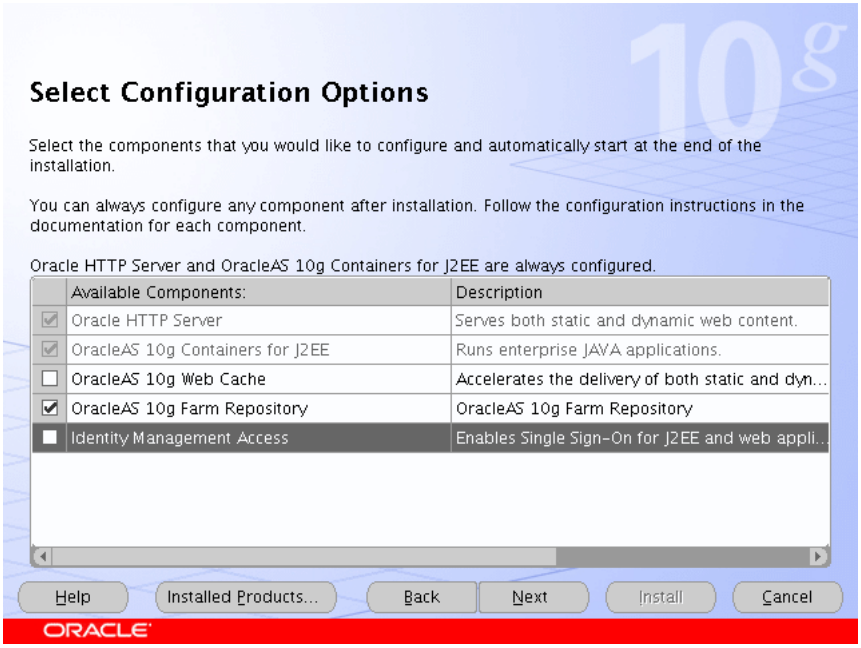
12. Select **J2EE and Web Cache**, as shown in Figure 3–6, and click **Next**.

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

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

    The **Select Configuration Options** screen appears.

Figure 3–7 Oracle Universal Installer Select Configuration Options Screen

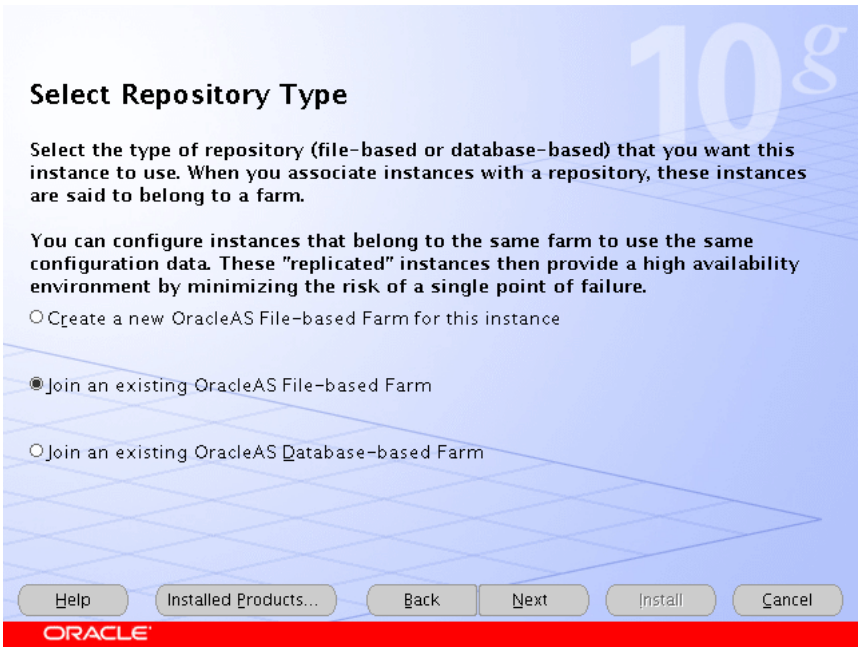


14. Select **OracleAS 10*g* Farm Repository**, as shown in Figure 3–7, and click **Next**.

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

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

Figure 3–8 Oracle Universal Installer Select Repository Type Screen



16. Select **Join an existing OracleAS File-based Farm**, as shown in Figure 3–8, and click **Next**.

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

17. 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 3.2.1, "A Note About Port Assignments for the Oracle Application Server File-based Farm" on page 3-2 for more information.

    The **Specify Instance Name and ias_admin Password** screen appears.

18. Specify an instance name and the OracleAS 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 10*g*. Start a browser and access **`http://`*`hostname`*`:1810`**.

3.2.4 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 10*g* 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.

3.2.5 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.

1. On the **Oracle Enterprise Manager 10*g* 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 Manger** 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 10*g* 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.

12. Repeat the steps in this procedure, selecting the APPHOST2 instance in Step 1.

3.2.6 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.

3.2.6.1 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 10*g* 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.

4. Click **Start** in the clusters section to start the cluster.

3.2.6.2 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 10*g* 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 3.2.6.1 and click **Join**.

   A confirmation screen appears.

4. Click **OK**.

   The **Farm** page appears.

5. Start the cluster created in Section 3.2.6.1.

6. Start the APPHOST2 instance.

7. Select the APPHOST2 instance.

8. Click **Join Cluster**.

The **Join Cluster** page appears.

9. Select the cluster created in Section 3.2.6.1 and click **Join**.

   A confirmation screen appears.

10. Click **OK**.

    The **Farm** page appears.

11. Start the APPHOST2 instance.

3.3 Installing and Configuring the Web Tier

The Web Tier consists of multiple middle tier Oracle Application Server instances, with only OracleAS Web Cache and Oracle HTTP Server configured. In the complete configuration, the OracleAS Web Cache instances balance incoming requests to the Oracle HTTP Servers, which route the requests to the OC4J instances on the application tier computers.

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
Web Cache HTTP Listen Port = 7777
Web Cache HTTP Administration Port = 4000
Web Cache HTTP Invalidation Port = 4001
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**.

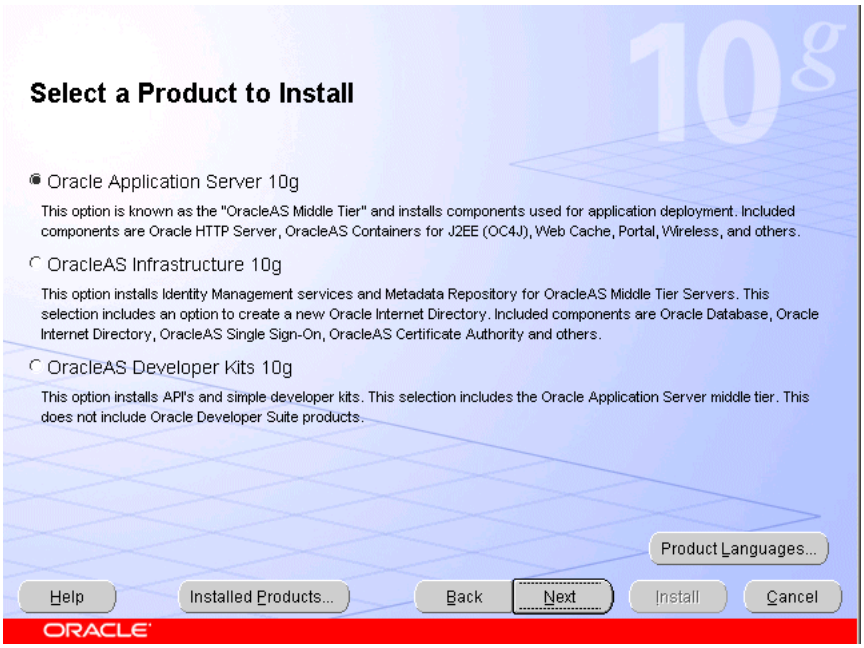
   The **Specify File Locations** screen appears with default locations for:

   - 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.

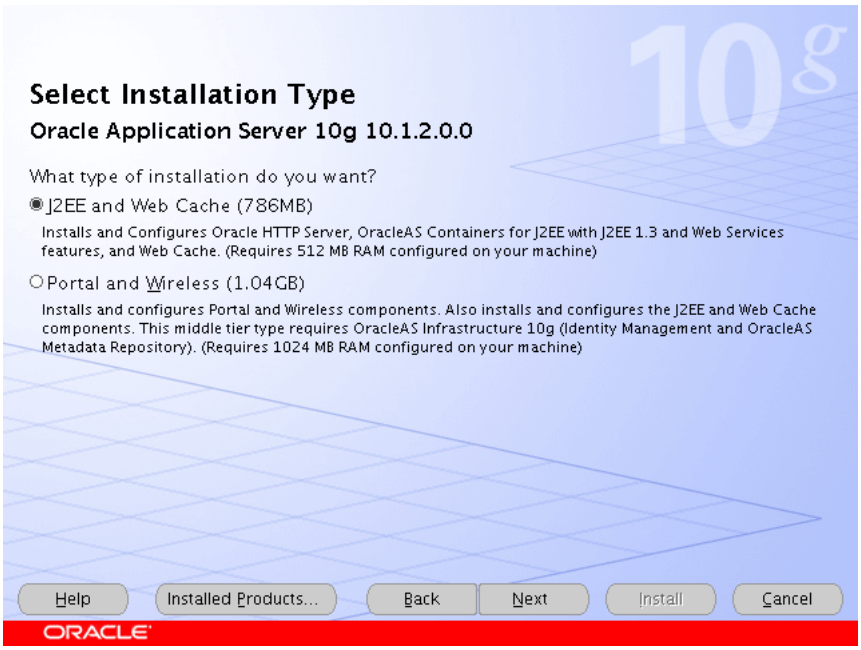
Figure 3–9 Oracle Universal Installer Select a Product to Install Screen



11. Select **Oracle Application Server 10*g***, as shown in Figure 3–9, and click **Next**.

    The **Select Installation Type** screen appears.

Figure 3–10 Oracle Universal Installer Select Installation Type Screen



12. Select **J2EE and Web Cache**, as shown in Figure 3–10, and click **Next**.

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

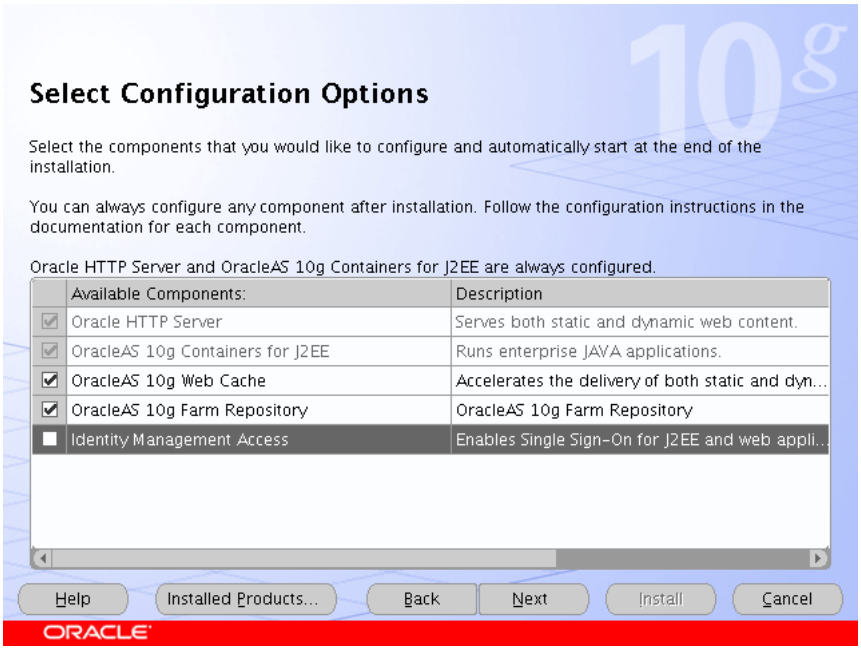
13. Click **Next**.

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

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

    The **Select Configuration Options** screen appears.

Figure 3–11 Oracle Universal Installer Select Configuration Options Screen



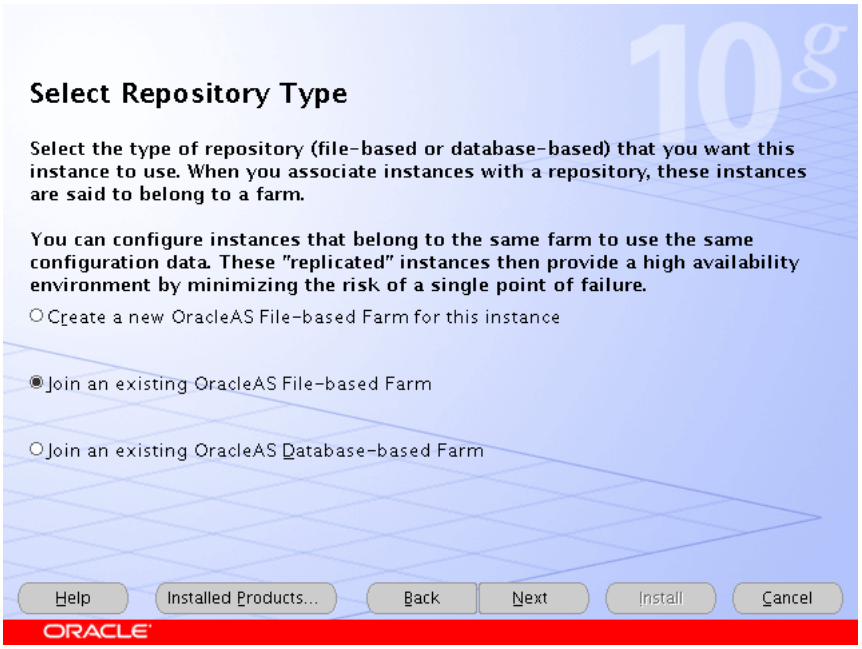
15. Select **OracleAS Web Cache and OracleAS 10*g* Farm Repository**, as shown in Figure 3–11, and click **Next**.

    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 3–12 Oracle Universal Installer Select Repository Type Screen



17. Select **Join an existing OracleAS File-based Farm**, as shown in Figure 3–8, 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 3.2.1, "A Note About Port Assignments for the Oracle Application Server File-based Farm" on page 3-2 for more information.

    The **Specify Instance Name and ias_admin Password** screen appears.

19. Specify an instance name and the OracleAS 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 10*g*. Start a browser and access **`http://hostname:1810`**.

3.4 Configuring the Load Balancing Router

The Load Balancing Router (myapp.mycompany.com, shown in Figure 1–1, "Enterprise Deployment Architecture for myJ2EECompany.com", must be configured to receive client requests and balance them to the two Oracle HTTP Server instances on the Web tier.

3.5 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 1–1.

1. Access the Oracle Enterprise Manager 10*g* Application Server Control Console.
2. Click the link for the WEBHOST1 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
      ```

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.

---

b. Add the following lines to create a `NameVirtualHost` directive and a `VirtualHost` container for myapp.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>
```

c. Create a second `NameVirtualHost` directive and a `VirtualHost` container for webhost1.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 the components on APPHOST1 using these commands in *WEBHOST1_ORACLE_HOME*`/opmn/bin`:

```
opmnctl stopall

opmnctl startall
```

3.6 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 3–1:

Example 3–1 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>
```

Follow these steps in APPHOST1 (the configuration will be replicated in APPHOST2, because the instances are clustered):

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

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

2. Click the link for the OC4J instance to configure.

   The page for the OC4J instance appears.

3. Click **Administration**.

4. Click **Advanced Properties**.

5. Click the **mod_oc4j.conf** link.

   The **Edit mod_oc4j.conf** screen appears.

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

   ```
   Oc4jConnTimeout 10
   ```

7. Add an `Oc4JMount` directive to specify the cluster to which requests should be load balanced. For example:

   ```
   Oc4jMount path cluster:
   //appcluster:OC4J1,appcluster:OC4J2,appcluster:OC4J3,appcluster:OC4J4...
   ```

   In the preceding example, *path* specifies the URI pattern of the request (such as the context root or application directory, that is, /myapp/*), *appcluster* is the name of the cluster created on the application tier, and *OC4J1* through *4* are the OC4J instance names.

3.7 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 mod_plsql 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 10*g* 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:

1. Create a file named `jazn_config.properties` in the `$ORACLE_HOME/config` directory that contains the following two lines and for which the current user has write permission:

   ```
   DCMRESYNC=oracle.ias.configtool.configimpl.DcmResync
   JAZN=oracle.security.jazn.util.JAZNConfigTool
   ```

2. Ensure that the operating system-specific environment variable that controls the loading of dynamic libraries is set. The library path should include `$ORACLE_HOME/lib`.

3. Issue the following command for the platform you are using (all on one line). Substitute values for the variables shown in bold. Table 3–2 describes the variables.

   Note: for the `-classpath` parameter, do not type any space characters after the colon (:) and semicolon (;) characters, as indicated by *<no spaces>*.

   On UNIX:

   ```
   $ORACLE_HOME/jdk/bin/java
   -classpath .:$ORACLE_HOME/sso/lib/ossoreg.jar:<no spaces>
   $ORACLE_HOME/jlib/ojmisc.jar:<no spaces>
   $ORACLE_HOME/jlib/repository.jar:<no spaces>
   $ORACLE_HOME/j2ee/home/jazn.jar:$ORACLE_HOME/jdk/lib/dt.jar:<no spaces>
   $ORACLE_HOME/jdk/lib/tools.jar:$ORACLE_HOME/jlib/infratool.jar
   oracle.ias.configtool.UseInfrastructure e
    -f $ORACLE_HOME/config/jazn_config.properties -h OID_HOST -p OID_PORT -u OID_
   ADMIN_NAME -w OID_PASSWORD
   -o ORACLE_HOME -m IAS_INFRA_INSTANCE_NAME
   -infra INFRASTRUCTURE_GLOBAL_DB_NAME  -mh MIDTIER_HOST
    -sslp SSL_PORT -sslf SSL_ONLY_FLAG
   ```

   On Windows:

   ```
   %ORACLE_HOME%\jdk\bin\java
   -classpath .;%ORACLE_HOME%\sso\lib\ossoreg.jar;<no spaces>
   %ORACLE_HOME%\jlib\ojmisc.jar;<no spaces>
   %ORACLE_HOME%\jlib\repository.jar;<no spaces>
   %ORACLE_HOME%\j2ee\home\jazn.jar;<no spaces>
   %ORACLE_HOME%\jdk\lib\dt.jar;<no spaces>
   %ORACLE_HOME%\jdk\lib\tools.jar;%ORACLE_HOME%\jlib\infratool.jar
   oracle.ias.configtool.UseInfrastructure e
    -f %ORACLE_HOME%\config\jazn_config.properties -h OID_HOST -p OID_PORT -u OID_
   ADMIN_NAME -w OID_PASSWORD
   -o ORACLE_HOME -m IAS_INFRA_INSTANCE_NAME
   -infra INFRASTRUCTURE_GLOBAL_DB_NAME  -mh MIDTIER_HOST
    -sslp SSL_PORT -sslf SSL_ONLY_FLAG
   ```

4. Verify that the command executed successfully by examining the *ORACLE_HOME*`/config/jazn_config.log` file.

5. Edit the *ORACLE_HOME*`/config/ias.properties` file to set the `OIDhost`, `OIDport` and `OIDsslport` values.

6. Verify that the provider was configured successfully using the JAZN administration tool in *ORACLE_HOME*`/j2ee/home`. Issue this command:

   ```
   $ORACLE_HOME/jdk/bin/java -jar jazn.jar -listrealms
   ```

Note: To enable the debug log for the administration tool, set the java option `"-Djazn.debug.log.enable=true"`

Table 3–2 Variables for the OracleAS JAAS Provider Configuration Command

Variable Name	Description	Example
ORACLE_HOME	Path to the Oracle home of the Oracle Application Server instance	/myj2eecompany/appserver
OID_HOST	Host name of the computer on which Oracle Internet Directory is installed	oidhost1.mycompany.com
OID_PORT	Oracle Internet Directory port number	3060
OID_ADMIN_NAME	Oracle Internet Directory administrator's distinguished name	cn=orcladmin
OID_PASSWORD	Oracle Internet Directory administrator's password	
IAS_INFRA_INSTANCE_NAME	Instance name of the Oracle Application Server Infrastructure instance	infradbhost1.mycompany.com
INFRASTRUCTURE_GLOBAL_DB_NAME	Global database name for the Infrastructure instance (as found in the tnsnames.ora file)	asdb
MIDTIER_HOST	Host name of the middle tier Oracle Application Server instance	apphost1.mycompany.com
SSL_PORT	SSL port for Oracle Internet Directory	3160
SSL_ONLY_FLAG	Enables or disables SSL communication for JAZN	false

3.8 Adding Administrative Users and Groups to Oracle Internet Directory for the Oracle Application Server Java Authentication and Authorization Service (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.

3.9 Configuring Secure Sockets Layer for the Oracle HTTP Server

To configure SSL on the connection path between external clients or the load balancer and Oracle HTTP Server, follow the instructions in the *Oracle HTTP Server Administrator's Guide*, section titled "Enabling SSL".

3.10 Configuring Secure Sockets Layer for OracleAS Web Cache

Depending on security needs, you may configure one or both of the following connection paths for OracleAS Web Cache:

- External Clients or Load Balancer to OracleAS Web Cache
- OracleAS Web Cache to Oracle HTTP Server

To configure OracleAS Web Cache for SSL, follow the instructions in "Configuring OracleAS Web Cache for HTTPS Requests" in the *Oracle Application Server Web Cache Administrator's Guide.*

3.11 Configuring Secure Sockets Layer for mod_oc4j and OC4J

To enable SSL communication between mod_oc4j and the OC4J instances, you must:

- Obtain an SSL certificate and place it in a wallet (see the *Oracle Application Server Administrator's Guide*).
- Enable SSL for mod_oc4j
- Enable SSL for OC4J

To enable SSL on mod_oc4j, use the Oracle Enterprise Manager 10*g* Application Server Control Console to edit the *ORACLE_HOME*`/Apache/Apache/conf/mod_oc4j.conf` file on WEBHOST1 and WEBHOST2:

1. On the **Oracle Enterprise Manager 10*g* Farm** page, select the WEBHOST1 instance.
2. Select **HTTP Server** from the **System Components** list.

   The **HTTP Server** page appears.
3. Click the **Administration** link.

   A list of links for configuration options appears.
4. Click **Advanced Server Properties**.

   The **Advanced Server Properties Configuration Files** page appears.
5. Click the **mod_oc4j.conf** link.

   The **Edit mod_oc4j.conf** screen appears.
6. Add this directive to enable SSL:

   ```
   Oc4JEnableSSL On
   ```
7. Add this directive to specify the location of the wallet (specify only the directory, not the file name, of the wallet):

   ```
   Oc4JSSLWalletFile path to file
   ```
8. Click **Apply**.

   The **Confirmation** screen appears.
9. Click **Yes** to restart the HTTP Server.

   The **Processing:Restart** screen appears, then the **Confirmation** screen appears with a message that the HTTP Server was restarted.
10. Click **OK**.

The **Edit mod_oc4j.conf** screen appears.

11. Enable the Auto Login feature in Oracle Wallet Manager to create an obfuscated copy of the wallet. Follow these steps:

    a. Start Oracle Wallet Manager with the command:

       (Windows) Select **Start > Programs > Oracle-HOME_NAME > Network Administration > Wallet Manager**

       (UNIX) Issue this command: `owm`.

    b. Choose **Wallet** from the menu bar.

    c. Check **Auto Login**. A message at the bottom of the window indicates that auto login is enabled.

To enable SSL for OC4J, specify the following settings in the *ORACLE_HOME*`/j2ee/home/config/default-web-site.xml` file, under the `<web-site>` element:

1. On the **Oracle Enterprise Manager 10*g* Farm** page, select the WEBHOST1 instance.
2. Select the OC4J instance from the **System Components** list.

   The OC4J instance page appears.
3. Click the **Administration** link.

   A list of links for configuration options appears.
4. Click **Advanced Server Properties**.

   The **Advanced Server Properties Configuration Files** page appears.
5. Click the **mod_oc4j.conf** link.

   The **Edit mod_oc4j.conf** screen appears.
6. Set `secure="true"` (in the `<web-site>` element) to direct the AJP protocol to use an SSL socket.
7. Specify the path and password for the keystore, as shown in the subsequent example.

```
<web-site ... secure="true" ... >
   ...
   <ssl-config keystore="path and file" keystore-password="password" />
</web-site>
```

> **Note:** The `<ssl-config>` element is required when the `secure` flag is set to `true`. The *path and file* value can indicate either an absolute or relative directory path, and includes the file name. A relative path is relative to the location of the Web site XML file.

8. (Optional) To specify that client authentication is required, set the `needs-client-auth` flag to `true`, as shown in the subsequent example.

```
<web-site ... secure="true" ... >
   ...
   <ssl-config keystore="path_and_file" keystore-password="pwd"
   needs-client-auth="true" />
</web-site>
```

When the `needs-client-auth` flag is set to true, OC4J accepts or rejects a client entity, such asOracle HTTP Server, for secure communication depending on its identity. The `needs-client-auth` flag instructs OC4J to request the client certificate chain upon connection. If OC4J recognizes the root certificate of the client, then the client is accepted.

The keystore that is specified in the `<ssl-config>` element must contain the certificates of any clients that are authorized to connect to OC4J through secure AJP and SSL.

Example 3–2 shows a sample configuration of secure AJP communication with client authentication. The settings pertinent to security are shown in bold text.

Example 3–2 Configuration for Secure AJP Communication with Client Authentication in default-web-site.xml File

```
<web-site display-name="OC4J Web Site" protocol="ajp13" secure="true" >
   <default-web-app application="default" name="defaultWebApp" root="/j2ee" />
   <access-log path="../log/default-web-access.log" />
   <ssl-config keystore="../keystore" keystore-password="welcome"
         needs-client-auth="true" />
</web-site>
```

4

Configuring the Application Infrastructure for myPortalCompany.com

This chapter provides instructions for creating the Application and Web Server tiers, distributing the software components into the DMZs shown in the Enterprise Deployment architecture depicted in Figure 1–2 on page 1-6.

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:

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

4.1 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 2.1.1.

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 2.1.2 or Section 2.1.3.
   - To install into a database using raw devices, follow the steps in Section 4.1.1, "Installing the Metadata Repository in a Database Using Raw Devices" on page 4-2.
   - To install into a database using Oracle Cluster File System, follow the steps in Section 4.1.2, "Installing the Metadata Repository in an Oracle Cluster File System (OCFS)" on page 4-4.
4. Perform the post-installation step described in Section 2.1.4.

4.1.1 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 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:

   - **`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** 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 Connection** 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.

4.1.2 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** 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 Connection** 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, writeable 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.

4.2 Installing the Application Tier

Follow these steps to install the Application Tier components (APPHOST1 and APPHOST2) into the Application tier.

4.2.1 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 = 4000
Web Cache Invalidation port = 4001
Web Cache Statistics port = 4002
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)
10. Click **Next**.

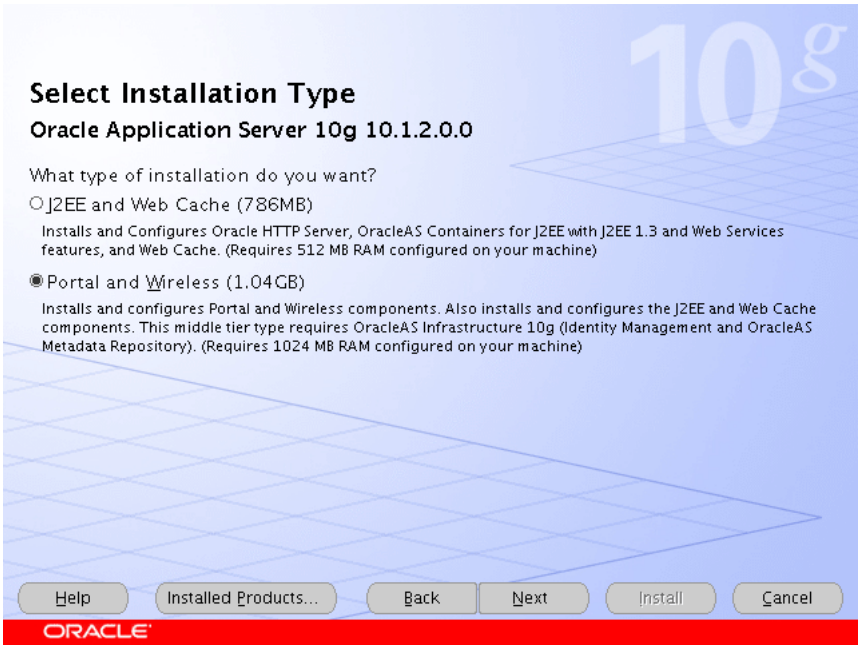
The **Select a Product to Install** screen appears.

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

11. Select **Oracle Application Server 10*g***, as shown in Figure 4–1, and click **Next**.

The **Select Installation Type** screen appears.

Figure 4–2 Oracle Universal Installer Select Installation Type Screen

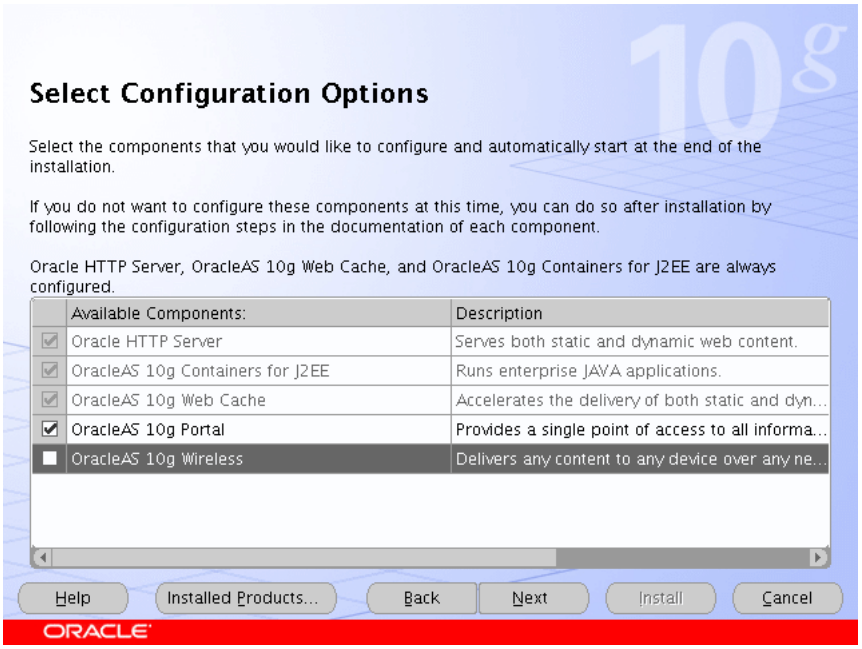


12. Select **Portal and Wireless**, as shown in Figure 4–2, and click **Next**.

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

13. Ensure that the requirements are met and click **Next**.
14. The **Select Configuration Options** screen appears.

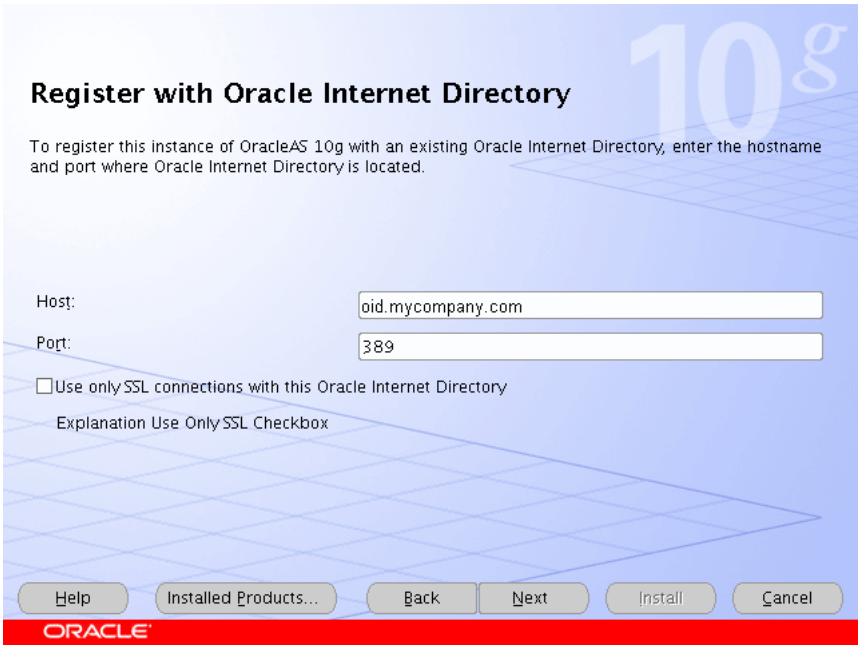
Figure 4–3 Oracle Universal Installer Select Configuration Options Screen



15. Select **OracleAS 10*g* Portal**, as shown in Figure 4–3, 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 4–4 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 10*g* 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 OracleAS 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:

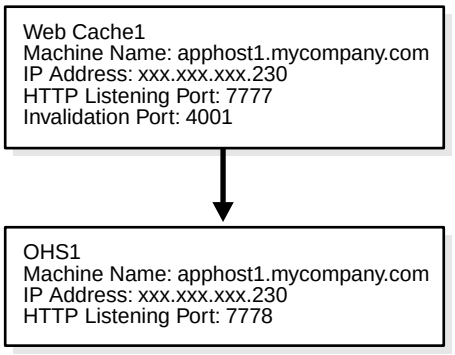
```
<IASConfig XSDVersion="1.0">
   <IASInstance Name="iAS-1.apphost1.mycompany.com"
Host="apphost1.mycompany.com">
       <WebCacheComponent ListenPort="7777" AdminPort="4000"
InvalidationPort="4001" InvalidationUsername="invalidator"
InvalidationPassword="@Bd4D+TnapIEqRc3/kle0A=" SSLEnabled="false"/>
<EMComponent ConsoleHTTPPort="1810" SSLEnabled="false"/>
   </IASInstance>

   <IASInstance Name="iAS.login.mycompany.com" Host="login.mycompany.com">
<OIDComponent AdminPassword="@BVs2Kn8lbTxUY=" SSLEnabled="false" LDAPPort="389"
AdminDN="cn=orcladmin"/>
   </IASInstance>

   <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="@BeyhA5zRtuYc=" ConnectString="cn=iasdb,cn=oraclecontext">
<WebCacheDependency ContainerType="IASInstance"
Name="iAS-1.apphost1.mycompany.com"/>
<OIDDependency ContainerType="IASInstance" Name="iAS.login.mycompany.com"/>
<EMDependency ContainerType="IASInstance" Name="iAS-1.apphost1.mycompany.com"/>
</PortalInstance>

</IASConfig>
```

Note: The value `iAS-1` in the `IASInstance` element is the instance name specified in step 22.

4.2.2 Configuring the First Application Server on APPHOST1

Upon installation of the first application server, the `iasconfig.xml` file shown above yields an OracleAS Web Cache configuration with the functionality shown in Figure 4–5.

Figure 4–5 Pre-Configuration Listener Setup on First Application Server



Before you begin this configuration, ensure that the following is configured:

- 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 4001 (an HTTP listening port), and balances them to the Application Tier OracleAS Web Cache on APPHOST1 port 4001 (an HTTP listening port). Port 4001 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 4001 for invalidation messages.
>
> For security reasons, port 4001 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 4001, use the following URL in the Load Balancing Router configuration:

  http://*hostname.domain*.com:4001

  For example:

  http://apphost1.mycompany.com:4001

  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.

The configuration of the OracleAS Portal application server tier on APPHOST1 consists of the following tasks:

- Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST1
- Configuring the Parallel Page Engine Loop-Back with the Load Balancing Router on APPHOST1
- Configuring the Event Servlet with the Load Balancing Router on APPHOST1
- Modifying the Portal Dependency Settings (iasconfig.xml) File on APPHOST1
- Registering the OracleAS Portal URLs with the Load Balancing Router on APPHOST1
- Re-Setting the Oracle Enterprise Manager 10g Link
- Configuring OracleAS Web Cache with the Load Balancing Router on APPHOST1

- 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 on APPHOST1
- Testing the Configuration on APPHOST1

Configuring the Oracle HTTP Server with the Load Balancing Router on APPHOST1

This step associates the components on which OracleAS Portal depends with the Load Balancing Router hostname and port: portal.mycompany.com:443.

1. Access the Oracle Enterprise Manager 10*g* 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
   ```

   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.

   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>
```

c. Create a second `VirtualHost` container for apphost1.mycompany.com and port 7777.

```
<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.

Configuring the Parallel Page Engine Loop-Back with the Load Balancing Router on APPHOST1

In this step, you configure (non-SSL) loop-back communication between the Load Balancing Router and the Parallel Page Engine on APPHOST1.

Before you start this configuration, ensure that:

- You are able to resolve portal.mycompany.com from APPHOST1, 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 APPHOST1:

  `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 APPHOST1. Issue this command on APPHOST1:

  `telnet portal.mycompany.com 7777`

  Verify that no connection failure message is returned.

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>
</servlet>
```

4. Save the `web.xml` file.

Configuring the Event Servlet with the Load Balancing Router on APPHOST1

Follow these steps to configure the event servlet:

1. Open the *APPHOST1_ORACLE_HOME*/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF/web.xml file.
2. Locate the Event servlet section.
3. Add the lines shown in bold:

```
<servlet>
 <servlet-name>event</servlet-name>
    <servlet-class>oracle.webdb.event.EventServlet</servlet-class>
            <init-param>
               <param-name>httpsports</param-name>
               <param-value>443</param-value>
            </init-param>
</servlet>
```

4. Save the web.xml file.
5. Issue this command in *ORACLE_HOME*/dcm/bin to update the DCM repository:

```
dcmctl updateconfig
```

6. Issue these commands in *ORACLE_HOME*/opmn/bin to restart the instance:

```
opmnctl stopall
opmnctl startall
```

Modifying the Portal Dependency Settings (iasconfig.xml) File on APPHOST1

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. Follow these steps to edit the file to include this information:

1. Create a backup copy of the *APPHOST1_ORACLE_HOME*/portal/conf/iasconfig.xml file.
2. Open the *APPHOST1_ORACLE_HOME*/portal/conf/iasconfig.xml file and perform the following steps:
   a. Make the additions and changes shown in bold in Example 4–1.

Example 4–1 Modifications to the iasconfig.xml File

```
<IASConfig XSDVersion="1.0">
 <IASFarm Name="Farm1.portal.mycompany.com" Host="portal.mycompany.com">
  <WebCacheComponent AdminPort="4000" ListenPort="443"
InvalidationPort="4001" InvalidationUsername="invalidator"
InvalidationPassword="@Beyh8p2bOWELQCsA5zRtuYc=" SSLEnabled="true"/>
 </IASFarm>
  <IASInstance Name="iAS-1.apphost1.mycompany.com" Host="apphost1.mycompany.com">
<WebCacheComponent AdminPort="4000" ListenPort="7777"
InvalidationPort="4001"
InvalidationUsername="invalidator"
InvalidationPassword="@BYgvINNtK1/ux15zmARPURHM2GMAkwK9UA=="
SSLEnabled="false"/>
 <EMComponent ConsoleHTTPPort="1810" SSLEnabled="false"/>
</IASInstance>
   <OIDComponent AdminPassword="@BVs2KPJEWC5a0l4n8lbTxUY=" SSLEnabled="false"
LDAPPort="389" AdminDN="cn=orcladmin"/>
</IASInstance>
```

```
<PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
 SchemaPassword="@BYgvINNtK1/uohU/kv+WvG21XiMAp6Wryw=="
ConnectString="cn=orcl1012,cn=oraclecontext">
<WebCacheDependency ContainerType="IASFarm"
 Name="Farm1.portal.mycompany.com"/>
<OIDDependency ContainerType="IASInstance" Name="iAS-1.iasclha3"/>
<EMDependency ContainerType="IASInstance"
 Name="apphost1.mycompany.com"/>
</PortalInstance>
</IASConfig>

<IASInstance Name="iAS-1.apphost1.mycompany.com" Host="apphost1.mycompany.com">
 <OIDComponent AdminPassword="@BYgvINNtK1/ux15zmARPURHM2GMAkwK9UA=="
AdminDN="cn=orcladmin" SSLEnabled="false" LDAPPort="389"/>
```

b. Save the `iasconfig.xml` file.

c. Encrypt any plain text passwords in the `iasconfig.xml` configuration file by setting the ORACLE_HOME environment variable, if necessary, and then issuing this command from *ORACLE_HOME*`/portal/conf`:

```
ptlconfig -encrypt
```

Registering the OracleAS Portal URLs with the Load Balancing Router on APPHOST1

In this step, you register the OracleAS Portal URLs using the Load Balancing Router hostname and port instead of the OracleAS Web Cache hostname and port. Follow the steps in this section to use the OracleAS Portal Configuration Assistant to register the URLs.

1. Ensure that the ORACLE_HOME environment variable is set.
2. Register the URLs using the Portal Dependency Settings tool (available in *APPHOST1_ORACLE_HOME*`/portal/conf`):

   ```
   ptlconfig -dad dadname -wc -site
   ```

   In the preceding command, *dadname* is the name of the OracleAS Portal Database Access Descriptor.

Re-Setting the Oracle Enterprise Manager 10*g* Link

To prevent access to Oracle Enterprise Manager 10*g* 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`:

```
ptlconfig -dad portal -em
```

Configuring OracleAS Web Cache with the Load Balancing Router 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 Load Balancing Router.

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:4000/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 so that it recognizes 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**.

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 these commands in SQL*Plus:

      ```
      create user prefstore identified by welcome;

      grant connect, resource to prefstore;
      ```

   c. 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="welcome"
  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"
/>
```

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>
```

2. Update the trusted certificates file used by OmniPortlet with the certificate of the Web site's certificate authority:

a. Follow step 1 in "Enabling Monitoring of the Load Balancing Router's OracleAS Portal Host and Port Settings on APPHOST1" on page 4-20. The HTTPS URL you can use to obtain the certificate from is https://portal.mycompany.com/pls/portal.

At the end of this step, you will have a certificate file named ias_certificate.cer.

b. Locate the provider.xml file in the *ORACLE_HOME*/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet/ directory. Look in this file to find the location of the trusted certificates file, specified in the trustedCertificateLocation tag.

If the tag is not specified or is commented out, the default is *ORACLE_HOME*/portal/conf/ca-bundle.crt. This file contains a list of Base64 certificates trusted by the OmniPortlet Provider.

c. Edit the OmniPortlet trusted certificates file by adding the contents of the certificate file you exported in step **a** (ias_certificate.cer) to the end of the file. Be sure to include all of the Base64 text from the certificate, including the BEGIN and END lines.

d. Restart the OC4J_Portal instance.

3. 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`

4. 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`

   Web Clipping Provider:

   `http://portal.mycompany.com:7777/portalTools/webClipping/providers/webClipping`

5. 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 on the **Navigator** link.

   c. Click on the **Providers** tab.

   d. Click on the **Registered Providers** link.

   e. Click on the **Edit Registration** link.

   f. Click on 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 **d** and **e** for both the OmniPortlet Provider and the Web Clipping Provider.

6. Refresh the Portlet Repository so that the Portal Tools portlets appear in the Portlet Builders folder in the Portlet Repository:

   a. Log in as the portal administrator, and click on the **Builder** link.

   b. Click on the **Administrator** tab.

   c. Click on the **Portlets** sub-tab.

   d. Click on the **Refresh Portlet Repository** link in the Portlet Repository portlet.

   e. The refresh operation continues in the background.

Note: Running ptlconfig again at any time after you have completed the steps in "Configuring the Portal Tools Providers on APPHOST1" will require you to repeat steps 4 and 5 in this section.

Re-registering mod_osso on APPHOST1

1. Set the ORACLE_HOME environment variable to the current Oracle home.
2. Execute the SSO registration script *ORACLE_HOME*/sso/bin/ssoreg. Example 4–2 shows the usage of ssoreg.sh on UNIX. (On Windows, the script name is ssoreg.bat.)

> **Note:** The script shown in Example 4–2 has multiple lines for readability only. When you execute the script, all parameters are on a single continuous line.

Example 4–2 ssoreg Usage

```
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. Access the following URL:

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

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**.

Verifying Connectivity for Invalidation Messages from the Database to the OracleAS Web Cache on APPHOST1 through the Load Balancing Router

When an object is changed in the database, the application metadata repository database sends an invalidation message to Webcache to invalidate that object if it exists in the cache. 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 4001`

Verify that no connection failure message is returned.

Enabling Monitoring of the Load Balancing Router's OracleAS Portal Host and Port Settings on APPHOST1

You must first configure a certificate in Oracle Enterprise Manager 10*g* in order to successfully monitor the OracleAS Portal metrics using the Oracle Enterprise Manager 10*g* Application Server Control Console. 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.

e. Restart the Application Server Control Console by issuing the following commands in *`ORACLE_HOME`*`/bin`:

`emctl stop iasconsole`

`emctl start iasconsole`

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
```

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/htp.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.

   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.

4.2.3 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.

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.

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 = 4000
Web Cache Invalidation port = 4001
Web Cache Statistics port = 4002
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`

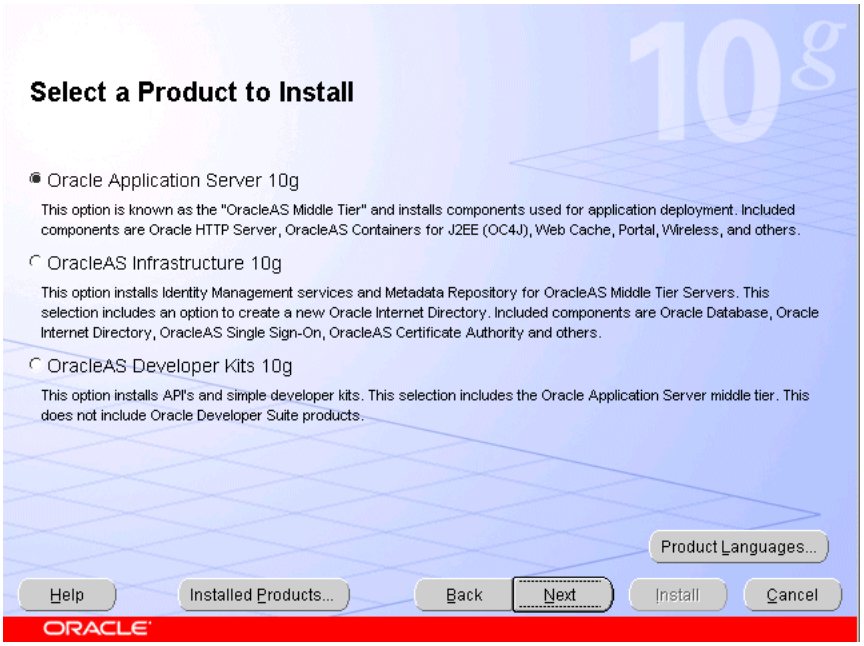
   All instructions for copying files from one computer to another assume this convention.

   ---

10. Specify the path and click **Next**.

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

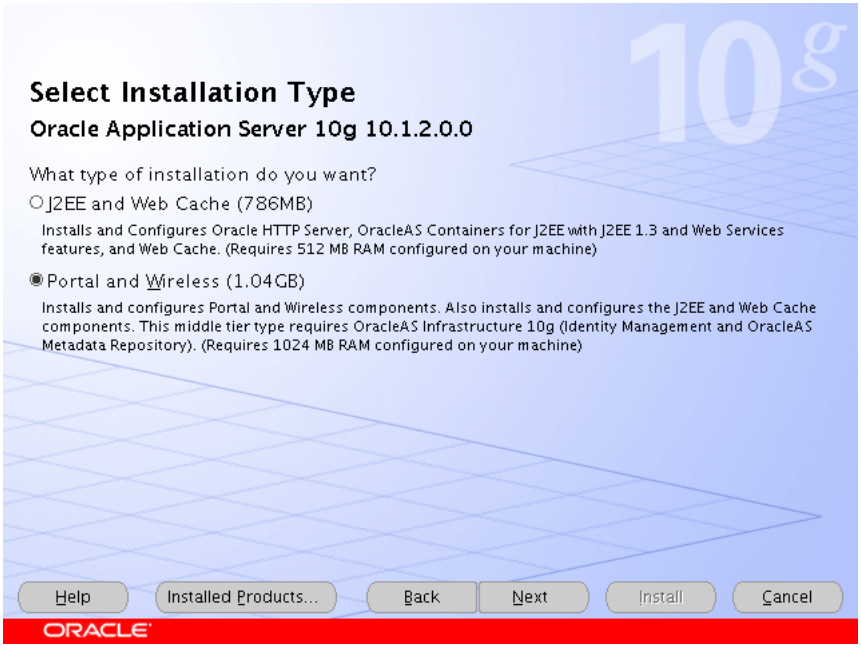
Figure 4–6 Oracle Universal Installer Select a Product to Install Screen



11. Select **Oracle Application Server 10*g***, as shown in Figure 4–6, and click **Next**.

    The **Select Installation Type** screen appears.

Figure 4–7 Oracle Universal Installer Select Installation Type Screen



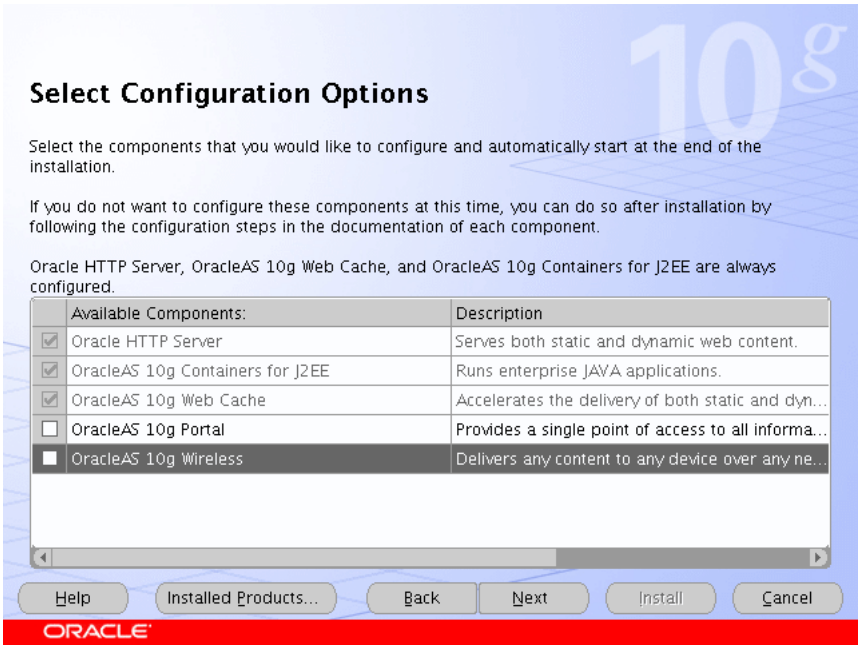
12. Select **Portal and Wireless**, as shown in Figure 4–7, and click **Next**.

    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 4–8 Oracle Universal Installer Select Configuration Options Screen



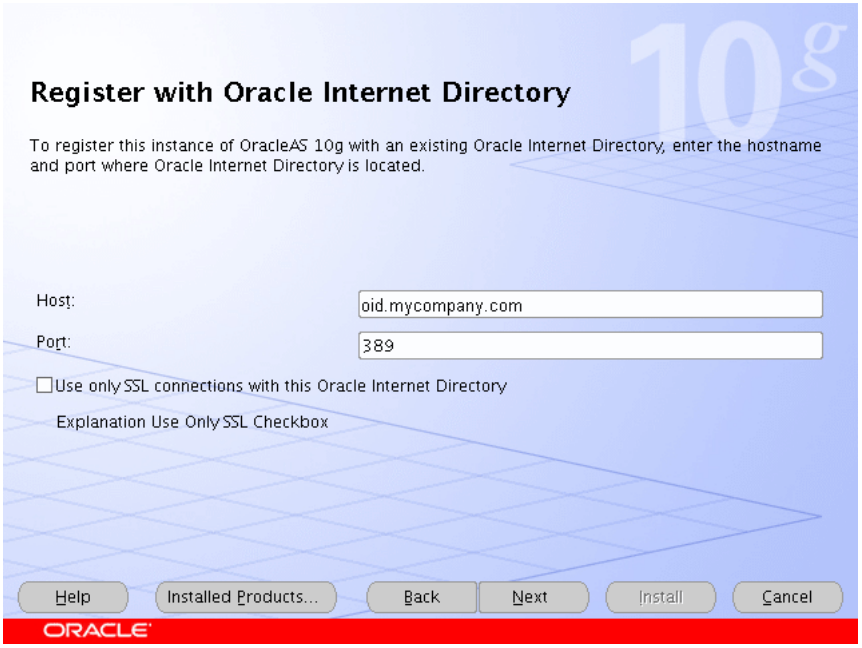
15. Do not select any configuration options, as shown in Figure 4–8, and click **Next**.

Note: Selecting the Oracle Application Server 10*g* 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 4–9 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 10*g* 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 OracleAS 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.

4.2.4 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
- Configuring the Event Servlet 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
- Enabling Monitoring of the Load Balancing Router's OracleAS Portal Host and Port Settings on APPHOST2

Enabling Portal on APPHOST2

The first task is to configure OracleAS Portal, using the Oracle Enterprise Manager 10*g* 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 the following is configured:

- 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.

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 10*g* 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
   ```

   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.

   ---

   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>
   ```

   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/`.

Configuring the Parallel Page Engine Loop-Back with the Load Balancing Router on APPHOST2

In this step, you provide (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 mod_plsql. 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.
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>
</servlet>
```

4. 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 mod_plsql.

5. 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
   ```

6. Restart all components on APPHOST2 by issuing the following command in *ORACLE_HOME*`/opmn/bin`:

   ```
   opmnctl stopall
   ```

   ```
   opmnctl startall
   ```

Configuring the Event Servlet with the Load Balancing Router on APPHOST2

Follow these steps to configure the event servlet:

1. Open the *APPHOST2_ORACLE_HOME*/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF/web.xml file.
2. Locate the Event servlet section.
3. Add the lines shown in bold:

```
<servlet>
 <servlet-name>event</servlet-name>
    <servlet-class>oracle.webdb.event.EventServlet</servlet-class>
            <init-param>
               <param-name>httpsports</param-name>
               <param-value>443</param-value>
            </init-param>
</servlet>
```

4. Save the web.xml file.
5. Issue this command in *ORACLE_HOME*/dcm/bin to update the DCM repository:

   dcmctl updateconfig

6. Issue these commands in *ORACLE_HOME*/opmn/bin to restart the instance:

   opmnctl stopall

   opmnctl startall

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.

1. Copy the *APPHOST1_ORACLE_HOME*/portal/conf/iasconfig.xml file to *APPHOST2_ORACLE_HOME*/portal/conf/.
2. Overwrite the file on APPHOST2 when prompted.

Configuring the Portal Tools Providers on APPHOST2

You must propagate the configuration made to Portal Tools providers on APPHOST1 to APPHOST2 by following these steps:

1. 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

2. 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

3. Copy the *APPHOST1_ORACLE_HOME*/j2ee/OC4J_Portal/config/data-sources.xml file to:

APPHOST2_ORACLE_HOME/j2ee/OC4J_Portal/config/data-sources.xml.

4. Copy the OmniPortlet trusted certificates file that you updated in APPHOST1 to APPHOST2. If you are using the default location, copy *APPHOST1_ORACLE_HOME*/portal/conf/ca-bundle.crt to *APPHOST2_ORACLE_HOME*/portal/conf/ca-bundle.crt.
5. Restart the OC4J_Portal instance.

Re-registering mod_osso on APPHOST2

1. Back up the *APPHOST2_ORACLE_HOME*/Apache/Apache/conf/osso.conf file.
2. Use FTP binary mode to copy the *APPHOST1_ORACLE_HOME*/Apache/Apache/conf/osso.conf file to *APPHOST2_ORACLE_HOME*/Apache/Apache/conf.
3. Synchronize the DCM repository with the FTP 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** forAPPHOST1 and APPHOST2.

4. Restart the components on APPHOST2 by issuing these commands in *APPHOST2_ORACLE_HOME*/opmn/bin:

   opmnctl stopall

   opmnctl startall

5. Access the following URL:

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

6. 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**.

4.2.5 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:4000/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: **4000**
   - Protocol for Admin. Port: **HTTP**
   - Cache Manager: **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.mycopany.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 "Testing the Configuration on APPHOST1" on page 4-22 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 on APPHOST2

You must first configure a certificate in Oracle Enterprise Manager 10*g* in order to successfully monitor the OracleAS Portal metrics using the Oracle Enterprise Manager 10*g* Application Server Control Console. 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 **Details** tab in the **Certificate** window.

   d. Click **Copy to File** to display the **Certificate Manager Export** wizard.

e. 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`.

f. 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.

e. Restart the Application Server Control Console by issuing the following commands in *`ORACLE_HOME`*`/bin`:

`emctl stop iasconsole`

`emctl start iasconsole`

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`

4.2.6 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 portal.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).
- Balance requests to portal.mycompany.com on port 7777 (an HTTP listening port) to the Application tier OracleAS Web Cache on APPHOST2 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 APPHOST2. 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 4001 (an HTTP listening port) to the Application Tier OracleAS Web Cache on APPHOST2 port 4001 (an HTTP listening port). Port 4001 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://apphost2.mycompany.com:7777/_oracle_http_server_webcache_static_.html

  To monitor invalidation port 4001, use this URL:

  http://apphost2.mycompany.com:4001/_oracle_http_server_webcache_static_.html

4.2.7 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 given 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:4000`

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**.

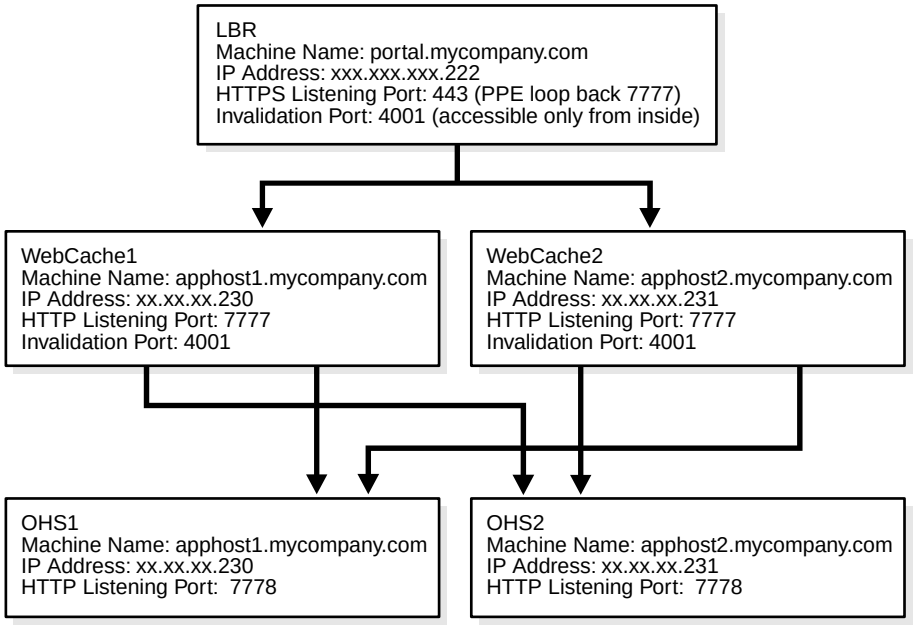
    The changes are propagated to the OracleAS Web Cache instance on the other computer.
11. Click **Restart**.

    OracleAS Web Cache is restarted on APPHOST1 and APPHOST2.

4.3 Testing the Application Server Tier

The complete configuration is shown in Figure 4–10.

Figure 4–10 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 the Oracle HTTP Server *APPHOST2_ORACLE_HOME*/Apache/Apache/logs/error_log file for information about how to resolve the error.

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.)

4.4 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 aLoad 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.

4.4.1 Deploying Custom JPDK Providers

Follow these steps to deploy custom JPDK providers:

1. Use the Oracle Enterprise Manager 10*g* Application Server Control Console to create a new OC4J instance named OC4J_JPDK on each middle tier instance.

2. Use the Application Server Control Console to deploy the custom providers in the OC4J_JPDK instances.

3. Use the Application Server Control Console to start the OC4J_JPDK on each middle tier instance.

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:

   `https://portal.mycompany.com:7777/<webApp>/providers/<provider name>`

4.5 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). Consesquently, 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:7777pls/orasso

A

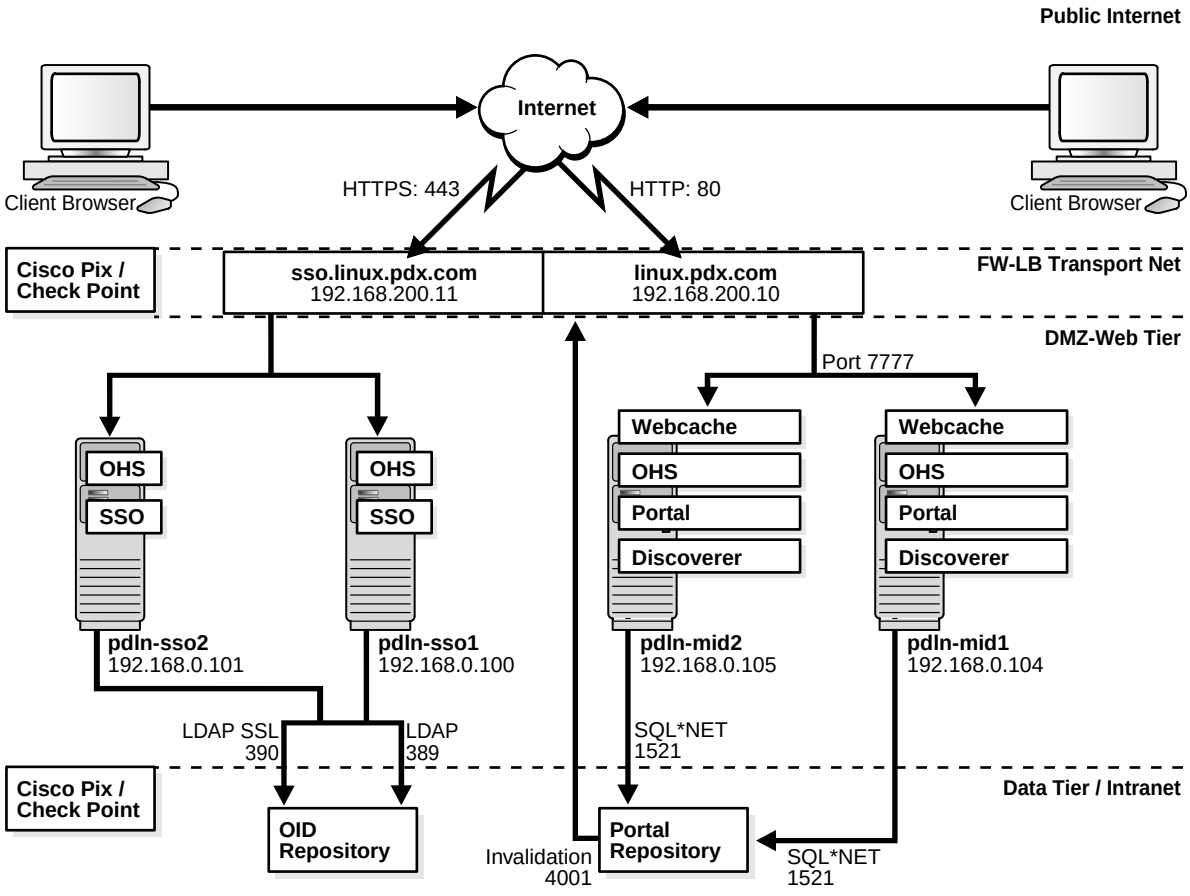
Sample Configurations for Certified Load Balancers

This appendix provides sample configurations for load balancers certified for use with Oracle Application Server. It contains these sections:

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.

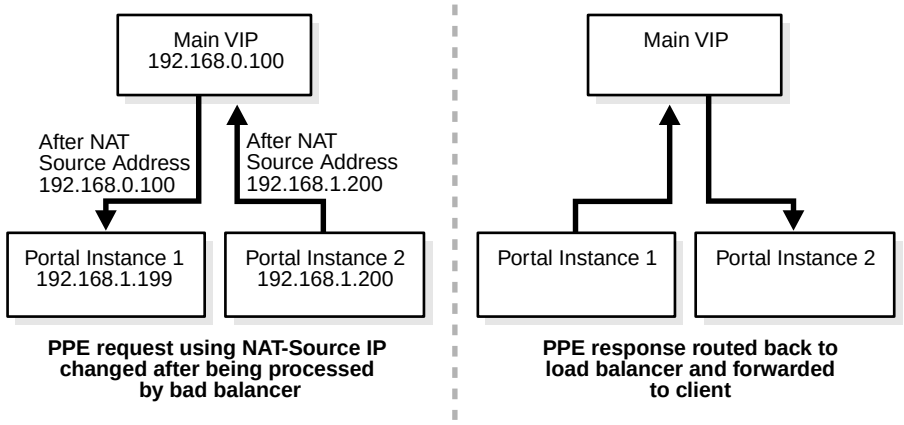
This section describes the communication in general terms and identifies the network configuration that enables it. For specific instructions on configuring a particular load balancer, refer to the section for that load balancer.

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.

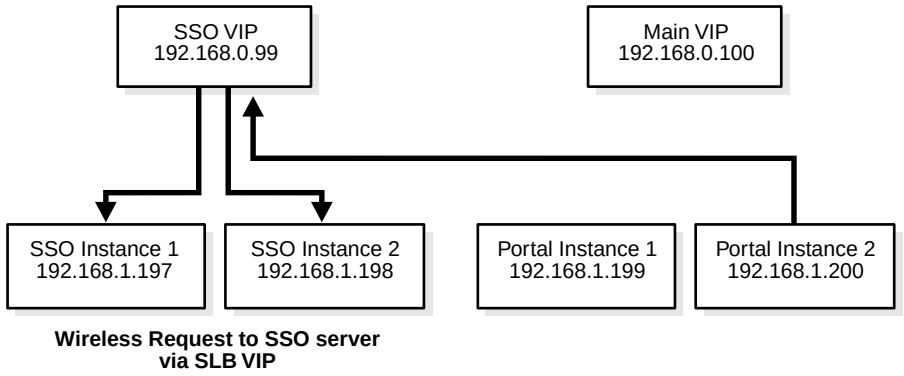
Figure A–2 OracleAS Portal Parallel Page Engine Network Address Translation



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 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 4001 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 10*g* 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

Pool 3: **OracleAS Web Cache Invalidation**

- pdln-mid1.pdx.com (Port 4001)
- pdln-mid2.pdx.com (Port 4001)
- 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
VIP3	192.168.200.10	4001	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 though 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 10*g* 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_INVAL (Web Cache Invalidation)

  NAT server

  NAT client WEBCACHE (for NAT of invalidation requests)

  Real 192.168.0.101 4001

  Real 192.168.0.105 4001

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

  Peristent 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 4001

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_INVAL
  nat server
  nat client WEBCACHE
  real 192.168.0.104 4001
   inservice
  real 192.168.0.105 4001
   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 SSO2
  virtual 192.168.200.11 tcp https
```

```
  vlan 200
  serverfarm SSO_FARM2
  persistent rebalance
  inservice
!
 vserver SSO3
  virtual 192.168.200.11 tcp https
  vlan 2
  serverfarm SSO_SSL-A
  persistent rebalance
  inservice
!
 vserver WEBCACHE_INVAL
  virtual 192.168.200.10 tcp 4001
  vlan 200
  serverfarm WC_INVAL
  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 10*g* 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 4001

- Server102 192.168.0.104 (OracleAS Portal on pdln-mid2)

  Source-NAT

  Port 7777

  Port 4001

- 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 though 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 4001
!
server real server102 192.168.0.104
 source-nat
 port 7777
 port 4001
 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 4001
 port 7778
 port ssl sticky
 bind http server102 7777 server103 7777
 bind 4001 server102 4001 server103 4001
 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 10*g* 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	4001	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 though 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 4001
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 4000
        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
        addclicert off
        addbeassl off
        addbeacli 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 4000
        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 10*g* 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–2 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	4001	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 though 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_INVAL -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_INVAL -as enable -dm cyclic
wsd farm table cdbset 192.168.0.151 WCACHE_INVAL -as enable -dm cyclic
wsd farm table cdbset 192.168.0.151 WCACHE_INVAL -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 4001 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#
```

B

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	*dbname*_raw_portal_128m
PORTAL_DOC	64	*dbname*_raw_portaldoc_64m
PORTAL_IDX	64	*dbname*_raw_portalidx_64m
PORTAL_LOG	64	*dbname*_raw_portallog_64m
DCM	256	*dbname*_raw_dcm_256m
OCATS	64	*dbname*_raw_ocats_64m
DISCO_PTM5_CACHE	64	*dbname*_raw_discoptm5cache_64m
DISCO_PTM5_META	64	*dbname*_raw_discoptm5meta_64m
WCRSYS_TS	64	*dbname*_raw_wcrsysts_64m
UDDISYS_TS	64	*dbname*_raw_uddisysts_64m
OLTS_ATTRSTORE	128	*dbname*_raw_oltsattrstore_128m
OLTS_BTTRSTORE	64	*dbname*_raw_oltsbttrstore_128m
OLTS_CT_STORE	256	*dbname*_raw_oltsctstore_256m
OLTS_DEFAULT	128	*dbname*_raw_oltsdefault_128m
OLTS_SVRMGSTORE	64	*dbname*_raw_oltssvrmgstore_64m
IAS_META	256	*dbname*_raw_iasmeta1_128m
DSGATEWAY_TAB	64	*dbname*_raw_dsgatewaytab_64m

B.2 Tablespace Mapping to Raw Devices Sample File

Example B–1shows 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/rdsk/oracle/mydb_raw_portal_128m
PORTAL_DOC1=/dev/vx/rdsk/oracle/mydb_raw_portal_doc_64m
PORTAL_IDX1=/dev/vx/rdsk/oracle/mydb_raw_portal_idx_64m
PORTAL_LOG1=/dev/vx/rdsk/oracle/mydb_raw_portal_log_64m
IAS_META1=/dev/vx/rdsk/oracle/mydb_raw_ias_meta_256m
DISCO_PTM5_META1=/dev/vx/rdsk/oracle/mydb_raw_disco_meta_64m
DISCO_PTM5_CACHE1=/dev/vx/rdsk/oracle/mydb_raw_disco_cache_64m
DCM1=/dev/vx/rdsk/oracle/mydb_raw_dcm_256m
WCRSYS_TS1=/dev/vx/rdsk/oracle/mydb_raw_clip_64m
OCATS1=/dev/vx/rdsk/oracle/mydb_raw_oca_64m
UDDISYS_TS1=/dev/vx/rdsk/oracle/mydb_raw_uddi_64m
OLTS_ATTRSTORE1=/dev/vx/rdsk/oracle/mydb_raw_olts_attr_128m
OLTS_BATTRSTORE1=/dev/vx/rdsk/oracle/mydb_raw_olts_battr_64m
OLTS_CT_STORE1=/dev/vx/rdsk/oracle/mydb_raw_olts_ct_store_256m
OLTS_DEFAULT1=/dev/vx/rdsk/oracle/mydb_raw_olts_default_128m
OLTS_SVRMGSTORE1=/dev/vx/rdsk/oracle/mydb_raw_olts_svrmgstore_64m
DSGATEWAY_TAB1=/dev/vx/rdsk/oracle/mydb_raw_synd_64m
b2b_dt1=/dev/vx/rdsk/oracle/mydb_raw_b2b_dt_256m
b2b_rt1=/dev/vx/rdsk/oracle/mydb_raw_b2b_rt_256m
b2b_lob1=/dev/vx/rdsk/oracle/mydb_raw_b2b_lob_256m
b2b_idx1=/dev/vx/rdsk/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 OracleAS 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 Databse 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 @BVXkuI3MPM1yWJArZp1kz4M4RP7rzEr/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

I

J

K

L

M

N

O

P

R

S

T

U

V

W