Oracle® Application Server Portal

Configuration Guide

10*g* Release 2 (10.1.4)

B19305-03

November 2005

ORACLE®

Oracle Application Server Portal Configuration Guide, 10*g* Release 2 (10.1.4)

B19305-03

Primary Author: Peter Lubbers

Contributing Authors: Pravin Prabhakar, Rosie Harvey, Lalithashree Rajesh, and Helen Grembowicz.

Contributors: Alistair Wilson, Andrew Wright, Arun Arat Tharakkal, Balakrishnan Jagdeesan, Balaravikumar Shanmugasundaram, Barry Hiern, Binodkumar Gupta, Chris van Es, Christopher Broadbent, Chung-Ho Chen, Dawn Tyler, Deborah Steiner, Demetris Christou, Dmitry Nonkin, Eddy Chee, Eric Lee, Greg Cook, Harry Wong, Jason Pepper, Joan Carter, John Bellemore, Madhu Muppagowni, Madhwa Chintarevula, Mahasweta Dey, Maheswaran Anantharaman, Marcie Caccamo, Mark Cann, Mark Clark, Mark Loper, Matthew Davidchuk, Michele Cyran, Mick Andrew, Nick Greenhalgh, Nick Pounder, P.V. Dharan, Panna Hegde, Pascal Gibert, Paul Encarnacion, Paul Spencer, Peter Moskovits, Preeti Yarashi, Pushkar Kapasi, Ramana Adusumilli, Ratna Bhavsar, Ravishankar Ramanathan, Richard Nessel, Rob Giljum, Ross Clewley, Sachin Parashar, Sanjay Khanna, Senthil Arunagirinathan, Sunil Marya, Susan Highmoor, Tiju Lei Oh, Tim Willard, Todd Vender, Venu Surakanti, Viswanath Dhulipala, and Yuhui Zhu.

Contents

5 Performing Advanced Configuration

6 Securing OracleAS Portal

7 Monitoring and Administering OracleAS Portal

8 Configuring the Search Features in OracleAS Portal

9 Tuning Performance in OracleAS Portal

10 Exporting and Importing Content

11 Using the Federated Portal Adapter

Part IV Appendixes

A Using the Portal Dependency Settings Tool and File

H Using TEXTTEST to Check Oracle Text Installation

I Configuring the Portal Tools Providers

J Setting Up and Maintaining a Virtual Private Portal

K Troubleshooting OracleAS Portal

Index

List of Examples

List of Figures

List of Tables

Preface

This manual describes how to configure Oracle Application Server Portal. This includes how to plan, upgrade, check pre-installation requirements, and perform post-installation tasks. This guide further explains some more advanced Portal deployments, and explains how to perform the advanced configuration required for these deployments. Finally, there is information about monitoring and troubleshooting.

> **Note:** For the portable document format (PDF) version of this manual, when a URL breaks onto two lines, the full URL data is not sent to the browser when you click it. To get to the correct target of any URL included in the PDF, copy and paste the URL into your browser's address field. In the HTML version of this manual, you can click a link to directly display its target in your browser.

Audience

This guide is intended for two kinds of users:

- **OracleAS Portal Administrators**, who are responsible for configuring and maintaining OracleAS Portal.
- **Oracle Application Server administrators**, who must configure OracleAS Portal to work with other Oracle Application Server components.

Documentation Accessibility

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

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

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an

otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

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

Related Documents

For more information, see the following manuals in the OracleAS Portal documentation set:

- Oracle Application Server Portal Release Notes
- *Oracle Application Server Portal User's Guide*
- *Oracle Application Server Portal Developer's Guide*

You can find all documentation related to OracleAS Portal, including the release notes, on the OracleAS Portal documentation page of the Oracle Technology Network (OTN): http://www.oracle.com/technology/products/ias/portal/documentation.html

---

Note: A complete glossary of OracleAS Portal-related terminology can be found in the *Oracle Application Server Portal User's Guide*.

You may also find the following manuals in the Oracle Application Server documentation set useful:

- *Oracle Application Server Concepts*
- *Oracle Application Server Security Guide*
- *Oracle Application Server Administrator's Guide*
- *Oracle HTTP Server Administrator's Guide*
- *Oracle Application Server Web Cache Administrator's Guide*
- *Oracle Application Server Wireless Administrator's Guide*
- *Oracle Application Server Single Sign-On Administrator's Guide*
- *Oracle Internet Directory Administrator's Guide*
- *Oracle Application Server Upgrade and Compatibility Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.
CAPITALIZED	Capitalized text indicates procedure names.
< >	Angle brackets enclose user-supplied information.
[ ]	Brackets enclose optional clauses from which you can choose one or none.
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.

What's New in Oracle Application Server Portal Configuration?

This chapter provides a brief description of new features introduced with the latest releases of Oracle Application Server Portal and provides pointers to additional information.

New Features Introduced with OracleAS Portal 10*g* Release 2 (10.1.4)

The new features of OracleAS Portal 10*g* Release 2 (10.1.4) include:

New Caching Architecture

In 10*g* Release 2 (10.1.4), OracleAS Portal introduces a major improvement in scaleability. In this release, OracleAS Web Cache, using Edge Side Includes (ESI) processing, is the entry point for page request processing rather than the Parallel Page Engine (PPE). This simplifies the page metadata (PMD) and it allows different types of metadata to be cached in OracleAS Web Cache at a more granular level, increasing the cache hit ratio and enabling a more granular invalidation of portal content. This new approach also provides support for secure full page caching in OracleAS Web Cache.

Query Path URL Supports SSL

OracleAS Portal maintains the URL prefix of OracleAS Single Sign-On, which accesses certain information through calls from the database using the UTL_HTTP package. These calls can now also be made using HTTPS. In previous releases, these calls were made using HTTP. As a result, even if OracleAS Portal and OracleAS Single Sign-On were configured to use HTTPS, you had to still use an HTTP port on OracleAS Single Sign-On to support these interfaces. If you are using HTTPS, then after configuring OracleAS Single Sign-On to use SSL, you must update the OracleAS Single Sign-On query path URL. See "Updating Wallet Path and Password in iasconfig.xml (HTTPS)" for information about the tasks to be performed.

New URL Format

The URL format in OracleAS Portal 10*g* Release 2 (10.1.4) has changed from `http://<host>:<port>/pls/<dad>` to `http://<host>:<port>/portal/pls/<dad>`. This change is to accommodate the availability of all necessary Portal Services running within OC4J_Portal. If URLs of the older format are accessed, then OracleAS Portal either automatically rewrites the URL to use the new format, or alerts you to change the bookmarked URL to the new format.

WSRP Support

Organizations engaged in enterprise portal projects have found application integration to be a major issue. Until now, users developed portlets using proprietary APIs for a single portal platform and often faced a shortage of available portlets from a particular portal vendor. All this changes with the introduction of Web Services for Remote Portlets (WSRP). WSRP is a Web services standard that allows the plug-and-play of visual, user-facing Web services with portals or other intermediary Web applications. Being a standard, WSRP enables interoperability between a standards-enabled container and any WSRP portal.

Search and Oracle Text Indexing Enhancements

Notable improvements to the search facilities in OracleAS Portal include:

- **Search Indexes Synchronize Automatically On Commit**

  If you are using Oracle Database 10*g*, you can now specify that Oracle Text indexes synchronize automatically whenever portal objects are added, modified, or deleted. This feature is useful for portal applications where newly added or altered content must be searchable immediately.

  To find out more, see Section 8.3.5.1, "Synchronizing Oracle Text Indexes". This feature is not available on databases earlier than Oracle Database 10*g*

- **Improvements to Document and URL Filtering**

  Oracle Text uses the AUTO_FILTER to convert documents and URL content into a plain text format that is suitable for indexing. Filtering content unnecessarily can impact the speed and efficiency of portal searches, so in this release OracleAS Portal introduces two special attributes for file- and URL- based item types: *MIME Type* and *Character Set*. These attributes enable portal users to classify portal content correctly when it is uploaded to the portal and this streamlines the filter process.

  For more information, see Section 8.3.3.7, "Maximizing AUTO_FILTER Performance".

Support for Enhanced Provider Message Authentication

In 10*g* Release 2 (10.1.4), OracleAS Portal introduces the support for Enhanced Provider Message Authentication. Enhanced message authentication secures the integrity of the headers that are used to propagate the user's authenticated identity to the Web provider. This enables you to leverage J2EE security in your provider code.

See Also:

- Section 6.3.1.3, "Configuring Provider Message Authentication" for information on how to configure Provider Message Authentication.
- *Oracle Application Server Portal Developer's Guide*

New Features Introduced with OracleAS Portal 10*g* Release 2 (10.1.2)

The new features of OracleAS Portal 10*g* Release 2 (10.1.2) include:

- Introduction of the Portal Services
- Support for Oracle Instant Portal
- ptlasst Command Line Utility Removed
- New ptllang Script to Install Languages
- Support for Global Inactivity Timeout in OracleAS Portal
- Support for User and Group LOVs in OracleAS Portal
- Configuring and Managing an Upgraded Oracle Application Server Portal Instance
- Automation of SSL Configuration Using the SSLConfigTool Tool

Introduction of the Portal Services

The OracleAS Portal OC4J instance now provides all the Portal Services used to assemble portal pages, access portal and page metadata, and so on.

The Parallel Page Engine (PPE) continues to be one of the Portal Services that assembles portal pages. Other services, like those previously provided by mod_plsql, are now incorporated in the Portal Services as well. Starting with this release, OracleAS Portal no longer depends on mod_plsql, but all previous functionality is still seamlessly provided by the Portal Services.

Although OracleAS Portal no longer has a runtime dependency on mod_plsql, it still has a dependency on the mod_plsql configuration files. These files are used by Portal Services in order to process requests for portal the same way that mod_plsql does.

Support for Oracle Instant Portal

Oracle Instant Portal, originally shipped as part of Oracle Application Server Standard Edition One, is shipped with OracleAS Portal 10*g* Release 2 (10.1.2). Oracle Instant Portal is a custom application built with OracleAS Portal for smaller enterprises to build simple portals in a short time. See *Oracle Instant Portal Getting Started* for more information.

ptlasst Command Line Utility Removed

The `ptlasst` command line utility has been removed. All of the functionality that was provided by `ptlasst` is now available through the Portal Dependency Settings tool, `ptlconfig`, and file, `iasconfig.xml`. See Appendix A, "Using the Portal Dependency Settings Tool and File" for details on using `ptlconfig` and `iasconfig.xml`.

New ptllang Script to Install Languages

OracleAS Portal is configured with the languages that are selected in the Oracle Universal Installer (OUI) during the Oracle Application Server middle-tier installation. To install languages, after you have installed OracleAS Portal, run `ptllang`.

In OracleAS Portal 10.1.2, the `ptllang` script completely replaces the `ptlasst.csh -mode LANGUAGE`. You must run `ptllang` for each language that you want OracleAS Portal to support. Refer to Section 4.9, "Configuring Language Support" for details.

Support for Global Inactivity Timeout in OracleAS Portal

A Global Inactivity Timeout can be configured for the OracleAS Single Sign-On (SSO) Server. This feature is now supported in OracleAS Portal 10*g* Release 2 (10.1.2). Refer to Section 6.1.6.1.3, "Support for Global Inactivity Timeout in OracleAS Portal" for details.

Support for User and Group LOVs in OracleAS Portal

User and Group LOVs work properly in OracleAS Portal 10*g* Release 2 (10.1.2) through the implementation of a new callback method. Oracle Delegated Administration Services posts the selected values to the callback method in OracleAS Portal's domain to avoid the cross-domain JavaScript issues. However, the releases of OracleAS Portal and Oracle Delegated Administration Services that you are using must support this callback method. 10*g* Release 2 (10.1.2) releases of both these components have this support. If OracleAS Portal 10*g* Release 2 (10.1.2) is used against an older release of Oracle Delegated Administration Services that does not support the callback method, then you must run the script `secjsdom.sql`, to set up a common domain. Refer to Section 6.1.6.2.5, "User and Group Lists of Values in OracleAS Portal" for details.

Configuring and Managing an Upgraded Oracle Application Server Portal Instance

A new appendix has been added that discusses how to configure a portal whose schema is not located in an OracleAS Metadata Repository, to take advantage of some of the new management services for Oracle Application Server 10*g* Release 3 (10.1.3). Refer to Appendix B, "Configuring and Managing an Upgraded Oracle Application Server Portal Instance" for details.

Automation of SSL Configuration Using the SSLConfigTool Tool

The SSL Configuration Tool (`SSLConfigTool`) simplifies and automates SSL configuration for common Oracle Application Server and OracleAS Portal configurations. It is designed to automate many of the manual steps currently required for configuring SSL. Refer to Section 6.3.2.1, "Configuring SSL for OracleAS Portal" for details.

Part I

Concepts

Part one contains the following chapters:

- Chapter 1, "Understanding the OracleAS Portal Architecture"
- Chapter 2, "Planning Your Portal"

1

Understanding the OracleAS Portal Architecture

This chapter introduces Oracle Application Server Portal and explains how it fits in the Oracle Application Server architecture.

This chapter contains the following sections:

- What Is the Oracle Application Server?, which provides you with a basic understanding of the solutions and components comprising Oracle Application Server so you can better understand how they work in concert with OracleAS Portal.
- Understanding the OracleAS Portal Architecture, which describes how OracleAS Portal and relevant pieces of Oracle Application Server work together.
- Understanding Caching in OracleAS Portal, which describes the caching configurations you can implement to increase the availability and scalability of medium to large deployments.

> **Note:** OracleAS Portal cannot be installed standalone, but must be installed as part of Oracle Application Server.

- Understanding WSRP and JPS, which provides an introduction to the Web Services for Remote Portlets (WSRP) specifications and Java Portlet Specification (JPS). These two standards enable the development of portlets that interoperate with different portal products, and therefore widen the availability of portlets within an organization.

1.1 What Is the Oracle Application Server?

Oracle Application Server is a completely standards-based application server that provides a comprehensive and fully integrated platform for running Web sites, J2EE applications, and Web services. It addresses all the challenges that you face as you refine your business processes to become an e-business.

Oracle Application Server provides full support for the J2EE platform, XML, and emerging Web services standards. With Oracle Application Server, you can simplify information access for your customers and trading partners by delivering *enterprise portals* that can be customized and accessed from a network browser or from wireless devices. It enables you to redefine your business processes and integrate your applications and data sources with those from your customers or partners. You can deliver tailored customer experiences through real-time personalization, and assess and correlate customer navigation, purchasing, ratings, and demographic data.

You can also implement a centralized management, security, and directory framework to manage and monitor all of your distributed systems and diverse user communities. Oracle Application Server maximizes your Web site infrastructure by deploying your fast, scalable Internet applications through built-in Web caching, load balancing, and clustering capabilities.

1.1.1 What Are the Oracle Application Server Solutions and Components?

Oracle Application Server is actually a set of Oracle Application Server *solutions*. Each solution contains one or more *components*. A component can be a service, an API, or an application. For detailed information, see the *Oracle Application Server Concepts* guide.

1.1.2 Overview of the Oracle Application Server Architecture

It is important to understand a bit about the overall Oracle Application Server architecture so you can more fully understand how your OracleAS Portal configuration fits within that structure. The next few sections provide some key concepts and terms you will need as you plan your configuration strategy.

The Oracle Application Server architecture consists of three basic tiers:

- Client Tier
- Middle Tier
- Infrastructure Tier

Client Tier

From the client computer, a user can connect to the middle tier and the infrastructure tier to access the self-service tools for publishing information, build applications, deploy content management, and administer enterprise portal environment.

Middle Tier

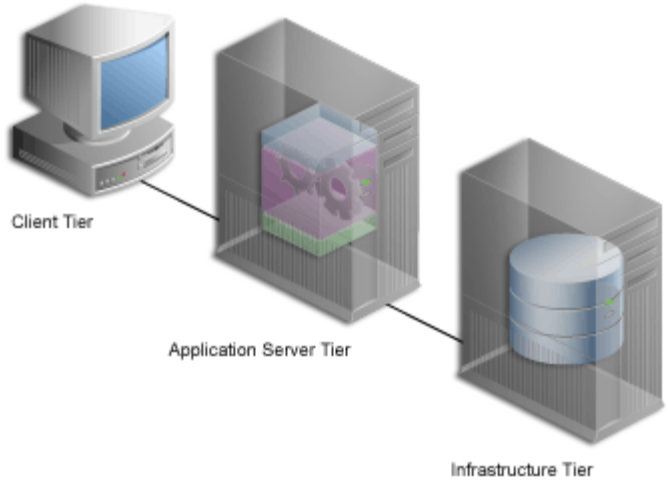
The middle tier, or *application server* tier, is a set of Oracle Application Server components typically installed into a single Oracle home. A single enterprise can have one or more application server installations, either residing on one host or, for more complex installations, distributed across multiple hosts.

Infrastructure Tier

The infrastructure installation consists of several components that help authenticate users, store access control information, and pass on the required content to the user based on the privileges the user has on OracleAS Portal. Like the middle-tier components, infrastructure components can be distributed across multiple hosts to enable scalability and high availability.

Figure 1–1 shows the three parts of the Oracle Application Server architecture.

Figure 1–1 Components of the Oracle Application Server Architecture



1.1.2.1 What Are the Middle-Tier Components?

The middle tier is the part of an Oracle Application Server architecture that contains several components responsible for accepting requests from clients, validating the requests, and providing content, while using intelligent data caching for faster and reliable performance.

For OracleAS Portal, the middle tier handles all Web requests by forwarding them to the appropriate provider. This is also where portal pages are assembled, and where the caching of portal content is managed. The middle tier also provides other functions for other Oracle Application Server components.

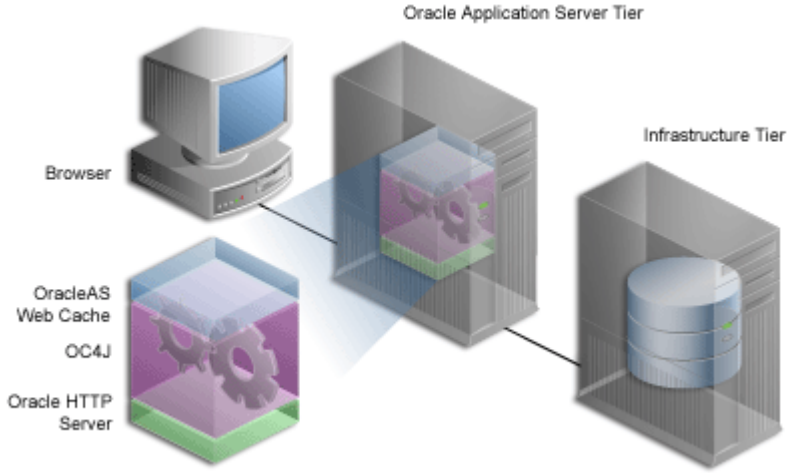
Some of the key components for OracleAS Portal in the Oracle Application Server middle tier are:

- **Oracle Containers for J2EE**. Oracle Containers for J2EE (OC4J) are fast, lightweight, and scalable J2EE containers that are written in Java and run on a standard Java Virtual Machine (JVM). OracleAS Portal's Parallel Page Engine (PPE), for example, is a servlet that assembles portal pages, and runs in the Oracle Containers for J2EE. OC4J have been designed for ease of use and to support standard APIs.
- **Oracle HTTP Server**. Oracle HTTP Server is the underlying deployment platform for all programming languages and technologies Oracle Application Server supports. Providing a Web listener for OC4J and the framework for hosting static and dynamic pages and applications over the Web, Oracle HTTP Server includes significant features that facilitate load balancing, administration, and configuration.
- **Portal Services**. For OracleAS Portal, Oracle HTTP Server handles all incoming HTTP requests to OracleAS Portal, by forwarding them to the OC4J_Portal instance, which provides all the Portal Services. The Parallel Page Engine (PPE) is one of the Portal Services that assembles portal pages. Other services, like those previously provided by mod_plsql, are now incorporated in the Portal Services as well.
- **Oracle Application Server Web Cache**. Works together with OracleAS Portal's own file-based caching to cache page definitions and content in memory, to boost performance. OracleAS Portal is closely integrated with OracleAS Web Cache to improve OracleAS Portal's overall availability, scalability, and performance.

OracleAS Web Cache combines caching and compression technologies to accelerate the delivery of both static and dynamically generated portal content.

- **Application Server Control Console**. This administration console for the Oracle Application Server enables you to administer clusters, start and stop services, enable and disable components, view logs and ports, and monitor servers in real-time.

Figure 1–2 The Middle-Tier Components



There are three types of middle-tier installations:

1. **Oracle Containers for J2EE and OracleAS Web Cache**, which is the simplest configuration and does not contain any of the OracleAS Portal Solution components.
2. **OracleAS Portal and OracleAS Wireless**, which adds the Portal and Wireless solutions to those provided by Oracle Containers for J2EE and OracleAS Web Cache.
3. **Business Intelligence and Forms**, which contains all of the middle-tier components, including OracleAS Portal.

To use OracleAS Portal, you must choose Option 2 or Option 3.

Refer to the following sections for more information:

- Section 2.1.7, "How Should I Configure My Hardware and Software?"
- Section 5.3, "Configuring Multiple Middle Tiers with a Load Balancing Router"

1.1.2.2 What Are the Infrastructure Components?

By default, the infrastructure tier handles all authentication requests and hosts the Oracle Application Server Metadata Repository, which contains schemas and business logic used by application server components (including OracleAS Portal) and other pieces of the infrastructure.

For the OracleAS Portal middle-tier installation, the infrastructure tier is a prerequisite.

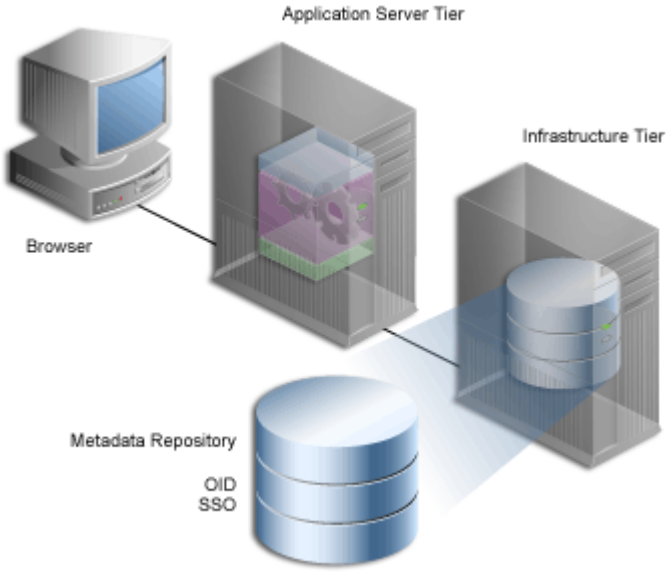
The Oracle Application Server Infrastructure contains:

- **Application Server Control Console**. This administration console for the Oracle Application Server enables you to administer clusters, start and stop services,

enable and disable components, view logs and ports, and monitor servers in real-time.

- **Oracle Internet Directory**. An LDAP version 3 compliant repository for storing user credentials and group memberships for OracleAS Portal and other Oracle products.
- **Oracle Application Server Single Sign-On (SSO)**. Authenticates user credentials against Oracle Internet Directory for OracleAS Portal and other applications, thus enabling users to log on once to the Web portal to access multiple accounts and applications with a single user name and password.
- **Oracle Application Server Metadata Repository**. The repository is installed in an Oracle Database and consists of a collection of schemas that contain product metadata for Oracle Application Server components. Some middle-tier components, such as OracleAS Portal, store their metadata in this repository and need access to that metadata during run time.

Figure 1–3 The Infrastructure Tier Components



You can install multiple instances of any of these components on multiple servers, and then connect the servers to suit your needs. Deployment configuration options for OracleAS Portal range from installing everything on a single server to multitier configurations in which the pieces comprising OracleAS Portal are located across multiple servers.

There are three types of OracleAS Infrastructure installations:

1. **Oracle Identity Management**, which installs and configures Oracle Identity Management services (Oracle Internet Directory, OracleAS Single Sign-On, Oracle Delegated Administration Services, Oracle Directory Integration and Provisioning, OracleAS Certificate Authority).
2. **OracleAS Metadata Repository**, which installs a new Oracle Database 10*g* containing the OracleAS Metadata Repository, and also stores the database objects that comprise OracleAS Portal, Oracle Internet Directory, and OracleAS Single Sign-On.
3. **Oracle Identity Management components and OracleAS Metadata Repository**, which consists of all the components listed in the preceding two installation types.

Note: Throughout this guide, you will see references to *ORACLE_HOME*. *ORACLE_HOME*, represents the full path of the Oracle home, and is used in cases where it is easy to determine which Oracle home is referenced. Oracle home contains all Oracle components selected for an installation type. You are prompted to enter an Oracle home in the **Path** field of the **Oracle Universal Installer File Locations** window.

The following conventions are used in procedures where it is necessary to distinguish between the middle tier, OracleAS Infrastructure, or OracleAS Metadata Repository Oracle home:

- *MID_TIER_ORACLE_HOME*, represents the full path of the middle-tier Oracle home.
- *INFRA_ORACLE_HOME*, represents the full path of the OracleAS Infrastructure Oracle home.
- *METADATA_REP_ORACLE_HOME*, represents the full path of the OracleAS Infrastructure home containing the OracleAS Metadata Repository.

1.2 Understanding the OracleAS Portal Architecture

After your development team builds your Web portal, the next step is to deploy a production version of it. Successful deployment means that end users are able to access content in a timely manner, without delays, errors, or server downtime. Because OracleAS Portal can be installed in a variety of configurations on different machines, a successful deployment ultimately depends how you configure portal to address the requirements of your site. This section provides some background information that should be useful to you as you plan your configuration.

1.2.1 How Does OracleAS Portal Integrate with Other Components?

Some Oracle Application Server components serve as *portlet providers*[1] for OracleAS Portal, which means you can easily integrate information from various components into a single portal page. Other components provide essential services to OracleAS Portal, as described in the following list.

- **Oracle Reports**. OracleAS Portal includes a simple report building facility. However, as your reports become more complex, you may want to import the report into OracleAS Reports Services to take full advantage of the functionality it offers. You can deploy any OracleAS Reports Services report as a portlet.

  See Also: *Oracle Application Server Reports Services Publishing Reports to the Web*

- **Oracle Business Intelligence Discoverer**. As a portlet provider, OracleBI Discoverer offers Worksheet portlets and List of Workbooks portlets to OracleAS Portal users. A Worksheet portlet contains information from a single Discoverer worksheet. The portlet displays this information in a table, a graph, or both. The List of Workbooks portlet presents a list of available workbooks.

[1] Applications and information sources, represented as portlets, communicate with the portal through a provider. Each portlet only has one provider, and a provider can have one or more portlets that expose an underlying application or information source.

See Also:

- The chapter titled "*Publishing workbooks to OracleAS Portal*" in the *Oracle Business Intelligence Discoverer Plus User's Guide* describes how to add a discoverer portlet.
- The chapter titled "*Using OracleAS Discoverer with OracleAS Portal*" in the *Oracle Business Intelligence Discoverer Configuration Guide* describes how to register the OracleBI Discoverer portlet provider with OracleAS Portal.

- **Oracle Ultra Search**. Oracle Ultra Search, which is integrated with OracleAS Portal, enables portal users to add a powerful search capability to their portal pages, and can be used to perform a search over a variety of content repositories and data sources. It also has the capability to crawl the OracleAS Portal Repository and search *public* content. Refer to Chapter 8, "Configuring the Search Features in OracleAS Portal" for more information about Oracle Ultra Search.
- **Oracle Application Server Wireless**. Working with OracleAS Wireless, OracleAS Portal automatically transforms the portal page structure to a format appropriate for the smaller screens of most wireless devices. Only portlets generating OracleAS Wireless XML content can display on a wireless device.

  OracleAS Portal developers also have access to a set of page design tools that help in creating portal pages that optimize the wireless experience. With these tools, developers can build a distinct portal structure for their wireless users. The wireless pages and portal pages can share portlet instances, which enables clients to reuse portlets on browser and wireless clients without reconfiguring each portlet.

  Refer to Section 4.6, "Configuring Mobile Support in OracleAS Portal" for more information.
- **Oracle Enterprise Manager 10*g***. Oracle Enterprise Manager 10*g* provides the Application Server Control Console, and the Grid Control Console. Oracle Enterprise Manager 10*g* Application Server Control Console can be used for monitoring, diagnostics, and for the configuration of OracleAS Portal-specific integration and performance settings. The Oracle Enterprise Manager 10*g* Grid Control Console can be used for monitoring OracleAS Portal, and tracking historical trends, but not for configuration. Refer to Chapter 7, "Monitoring and Administering OracleAS Portal" for more information about monitoring OracleAS Portal.
- **Oracle Application Server Forms Services**. Oracle Forms applications combine interactive, graphical interfaces with strong support for data validation. Forms developers can quickly create applications with powerful data manipulation features. OracleAS Forms Services deploys Forms applications to Java clients in a Web environment. OracleAS Forms Services automatically optimizes class downloads, network traffic, and interactions with the Oracle Database. OracleAS Forms Services applications are secured by the OracleAS Single Sign-On, and accessed from an OracleAS Portal environment provided by Oracle Application Server.
- **Oracle Application Server Single Sign-On**. OracleAS Single Sign-On authenticates users who are attempting to gain access to non-public areas of your portal. Refer to Section 6.1.6.1, "Relationship Between OracleAS Portal and OracleAS Single Sign-On" for more information.

- **Oracle Internet Directory**. Oracle Internet Directory is Oracle's highly scalable, LDAP version 3 service and hosts the Oracle common user identity. OracleAS Portal queries the directory to determine a user's privileges and what they are entitled to see and do in the portal. In particular, OracleAS Portal retrieves the group memberships of the user from the directory to determine what they may access and change. Refer to Section 6.1.6.2, "Relationship Between OracleAS Portal and Oracle Internet Directory" for more information.
- **Oracle Delegated Administration**. In addition to querying Oracle Internet Directory for user and group information, OracleAS Portal must provide users with a user interface to add and modify user and group information. To change information in the directory, you use the Oracle Delegated Administration Services user interface. OracleAS Portal provides links to the Oracle Delegated Administration Services for users with sufficient privileges to add and change users and groups. Refer to Section 6.1.6.4, "Relationship Between OracleAS Portal and Oracle Delegated Administration Services" for more information.
- **Oracle Directory Integration and Provisioning**. Oracle Directory Integration Platform notifies OracleAS Portal upon the occurrence of any directory events (for example, user deletions) to which OracleAS Portal subscribes. In essence, the directory integration server informs OracleAS Portal when a change occurs in the directory that requires a change in OracleAS Portal. Refer to Section 6.1.6.3, "Relationship Between OracleAS Portal and Oracle Directory Integration Platform" for more information.
- **Oracle Application Server Metadata Repository**. The repository is installed in an Oracle Database 10*g* and consists of a collection of schemas that contain product metadata for Oracle Application Server components. Some middle-tier components, such as OracleAS Portal, store their metadata in this repository and need access to that metadata during run time.

1.2.2 How Do the Pieces Fit Together?

A portal is comprised of groups of pages, each page divided into regions. The regions specify how space on a given page is allotted to that page's items and portlets.

1.2.2.1 How Are Pages Assembled in OracleAS Portal?

Each time a user requests an OracleAS Portal page, the page is dynamically assembled and formatted according to the portlets and layout chosen for that page. Keep in mind that the parts that comprise the page are typically drawn from a variety of sources. For example, the page's layout, look and feel, and user personalizations are stored in the database as part of the overall page definition, completely separate from any portlet content. This information may, in turn, be cached by the middle tier.

The fully assembled page may be cached in OracleAS Web Cache based on the page's caching properties. However, if full-page caching is used, pages are not re-assembled with each request, because they are served directly out of OracleAS Web Cache.

The portlets that appear on the page can be written in PL/SQL or Java. For PL/SQL portlets, the source is an OracleAS Metadata Repository database. This could be the database where the current instance of OracleAS Portal is installed, or some other OracleAS Metadata Repository database located on a remote server, which is accessed through the Federated Portal Adapter. If written in Java, a Web provider provides the portlet from any location accessible from the network, either Internet or intranet. For example, you could create a portal page that displays content from many different providers. These providers can be database providers, Web providers, or WSRP producers.

In 10*g* Release 2 (10.1.4), OracleAS Portal introduces a noticeable improvement in scalability, which is best described by looking at the way page requests are processed. Figure 1–4 shows how a typical page is assembled. In this release, OracleAS Web Cache, using Edge Side Includes (ESI) processing, is the entry point for page request processing rather than the PPE. The various pieces of metadata involved in a page request are cached at a more granular level, increasing the cache hit ratio and enabling a more granular invalidation of portal content. This new approach also provides support for secure full page caching in OracleAS Web Cache.

Figure 1–4 Portal Page Request Flow

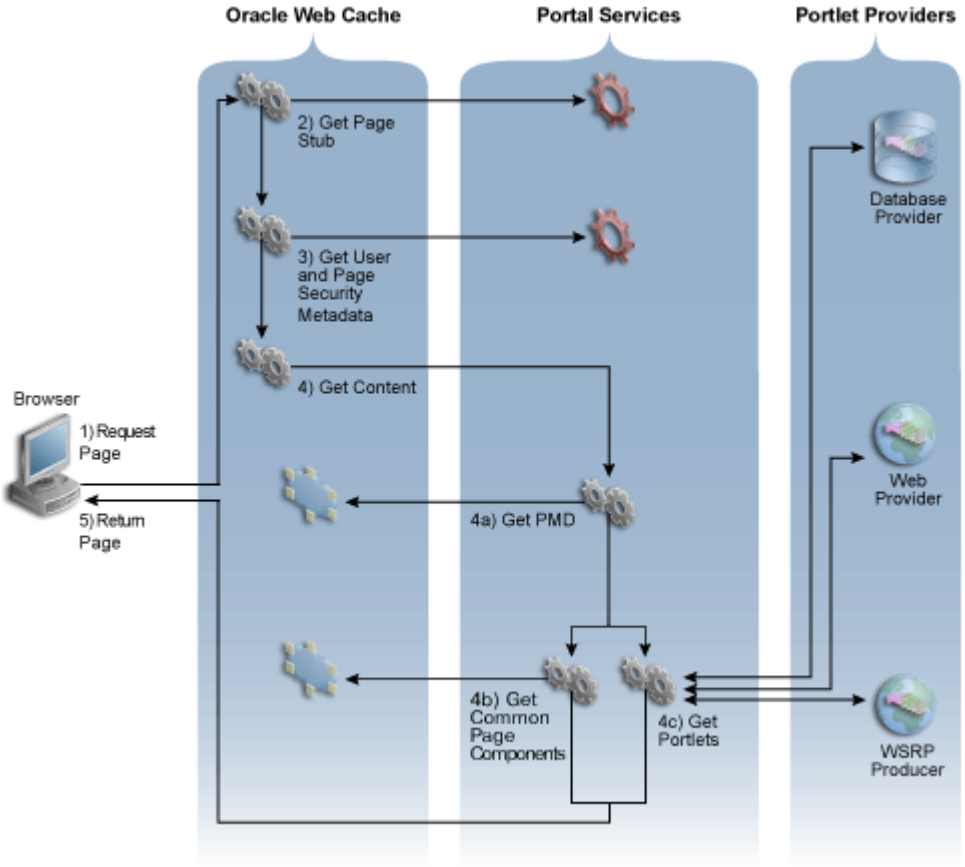


Figure 1–4 shows the steps performed when a client requests an OracleAS Portal page.

1. The client browser requests a portal page. OracleAS Web Cache receives this request.
2. OracleAS Web Cache retrieves the page stub. You can think of the page stub as a blueprint for the page that is to be assembled. If the page stub is not already cached in OracleAS Web Cache, then it is generated by the Portal Services running in the OracleAS Portal OC4J instance

> **Note:** The Portal Services are used to assemble portal pages, access portal and page metadata, and so on. The Parallel Page Engine (PPE) continues to be one of the Portal Services that assembles portal pages. Other services, like those previously provided by mod_plsql, are now incorporated in the Portal Services as well.

3. OracleAS Web Cache parses the page stub and retrieves additional user and page security metadata. Examples of the User Metadata (UMD) are user name, device type, and language. This user information is gathered once per session per user. If the UMD is not already cached in OracleAS Web Cache, then it is generated by the Portal Services running in the OracleAS Portal OC4J instance. The page security metadata (SMD) contains information that is used to determine whether the user is authorized to see the content at a given URL. If the SMD is not already cached in Web Cache, a request is sent through Portal Services to the portal schema in the OracleAS Metadata Repository. If the user has not logged in and is requesting private data, then the user will be challenged to log in. An error page displays if it turns out that the user is not authorized to view the page.
4. OracleAS Web Cache returns the already fully assembled copy of the page if it is found in the cache. Otherwise, it requests the content from Portal Services. The content of the page is assembled as follows:
   a. Portal Services tries to get the cached copy of the page metadata (PMD) from OracleAS Web Cache. The PMD, or the page definition, contains information about the portlets on a page and their layout. If there is a cache miss in OracleAS Web Cache, then it checks if the portal cache has a valid cached copy. Finally, if no cached copy of the PMD exists, then the portal schema in the OracleAS Metadata Repository generates the PMD.
   b. Portal Services retrieves common page components, such as navigation portlets, banners, tabs, and subpage regions, from OracleAS Web Cache if they exist. These requests are performed in parallel. A request is sent to the portal schema in the OracleAS Metadata Repository if no valid cached copies exist.
   c. For each portlet on the page, Portal Services checks if a cached copy of the portlet's content exists in the Portal Cache. If there is a cache hit, the cached content is used. If there is a cache miss, Portal Services fetches the content from the appropriate provider. These requests are performed in parallel along with the requests described in the preceding step. Each provider returns content for the portlet. Content can be requested from Web providers, WSRP producers, or Database (DB) providers.
5. OracleAS Web Cache returns the fully assembled page to the client browser.

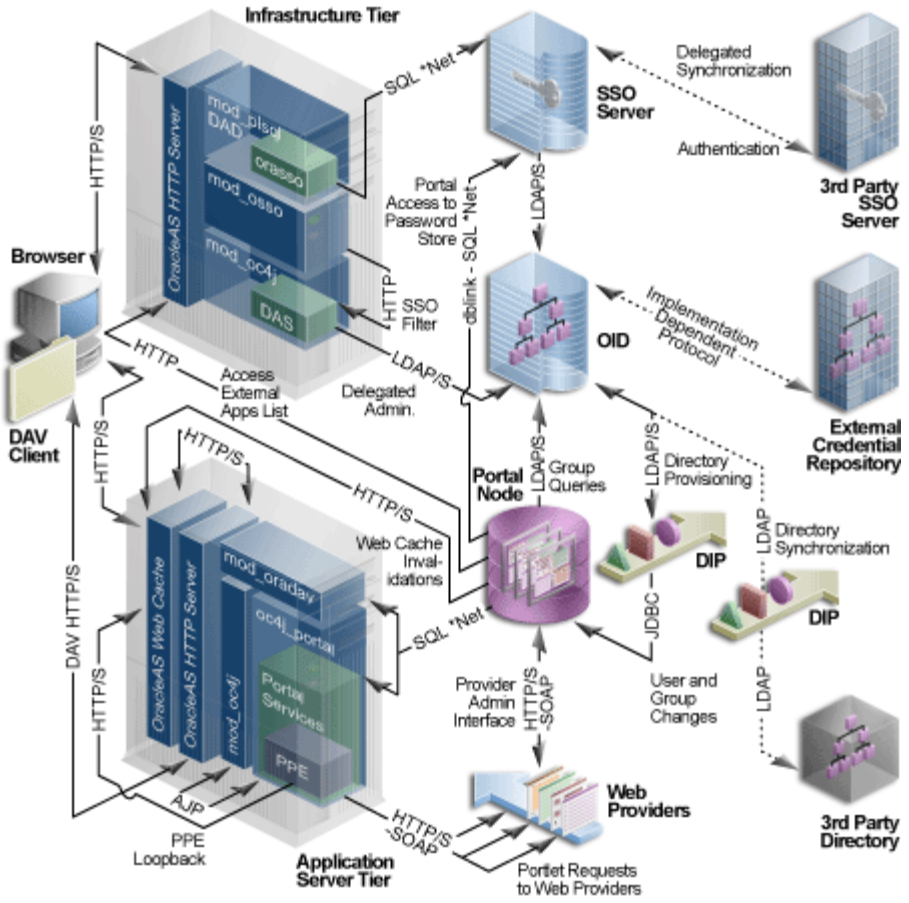
1.2.2.2 How Does Communication Flow in OracleAS Portal?

The OracleAS Portal implements a distributed architecture consisting of multiple communication points and protocols. For complex configurations including the introduction of firewalls and proxies, you need to understand the communication points, and how the various components of OracleAS Portal integrate together. Likewise, to allow for the distribution of the various functions across multiple servers, it is necessary to be aware of the network protocols that are used in the internode communication.

The OracleAS Portal architecture consists of three basic tiers: the client browser (pictured at the far left in Figure 1–5) the middle-tier server (pictured on the bottom left), and the infrastructure server and repositories (pictured on the top left). Although the default installation places all servers and repositories on the same host, it is recommended that you install these functions on separate servers, for increased performance and high availability.

Figure 1–5 shows in detail the communication flow between the various components of OracleAS Portal and Oracle Application Server.

Figure 1–5 Communication Flow and Protocols



The three tiers and the communication protocols used between them is described next:

- Client
- Infrastructure Tier
- Middle Tier

Client

- The client sends a request to OracleAS Portal, which is part of the middle tier, using the HTTP or HTTPS protocols. The use of firewalls and proxies is supported between the client and the middle tier.
- If the user needs to be authenticated, the client browser is redirected to the Oracle HTTP Server in the infrastructure tier. This connection is through HTTP or HTTPS and supports the implementation of both firewalls and reverse proxies in the network environment.

Infrastructure Tier

The infrastructure tier consists of the Oracle HTTP Server, OracleAS Single Sign-On, Oracle Internet Directory, and OracleAS Metadata Repository.

- If the requested page requires authentication, the user is prompted for a user name and password. This function is carried out by the Portal Services, through a redirection to OracleAS Single Sign-On for authentication. All authentication requests are communicated using the SQL*Net protocol.

- OracleAS Single Sign-On verifies user credentials against the Oracle Internet Directory through LDAP/S. The credentials are compared to those found within the Directory (LDAP compare) and the result returned to OracleAS Single Sign-On. Upon successful authentication, OracleAS Single Sign-On creates a single sign-on cookie. Once the user is authenticated and an appropriate OracleAS Portal session created, the user may access pages and other objects.
- As the access control lists (ACL) for all portal objects are held in the OracleAS Metadata Repository, the OracleAS Portal uses an LDAP/S request to communicate with the Oracle Internet Directory to query the appropriate user and group membership information defined in the Directory. When a user first logs in to OracleAS Portal, the group memberships of the user are copied to the portal node and cached on that tier. This process allows for fast lookup of object privileges. Once the object and page privileges of the user are known, the Parallel Page Engine goes on to generate the page from the appropriate pieces.
- All user provisioning is performed against the Oracle Internet Directory. The interface between the Infrastructure tier's Oracle HTTP Server and the LDAP server is through the Oracle Delegated Administration Services servlet. The Oracle Delegated Administration Services interface uses the LDAP/S protocol to communicate with the Oracle Internet Directory.
- The OracleAS Single Sign-On model includes the addition of mod_osso, which allows any URL to be protected within the OracleAS Single Sign-On environment. Calls to the Delegated Administration Services servlet are protected by the mod_osso plug-in. This verifies that the user has been properly authenticated before providing access to the Oracle Internet Directory. In effect, mod_osso filters the URL and forwards the HTTPS-based request, only if the user has previously been authenticated.
- The Oracle Directory Integration Platform automatically keeps the locally cached information up to date with changes in the Oracle Internet Directory. Just as the Oracle Directory Integration Platform keeps the local cache synchronized with the Oracle Internet Directory, it also keeps the Oracle Internet Directory synchronized with any external repository. The Oracle Directory Integration Platform communicates with the Oracle Internet Directory through LDAP/S.

Middle Tier

The middle tier consists of the OracleAS Web Cache, Oracle HTTP Server, Oracle Containers for J2EE, and other Oracle Application Server components.

> **Note:** OracleAS Web Cache and Oracle HTTP Server can be installed on different hosts to allow scalability and high availability.

- OracleAS Web Cache front ends the middle-tier components and thus optimizes the throughput of OracleAS Portal. When a page request comes from the browser, OracleAS Web Cache evaluates the URL and services the request from the cache if possible. If a requested page is not previously cached, the request is forwarded to its origin server (Oracle HTTP Server in this case) for generation. As a Web accelerator, OracleAS Web Cache allows the use of HTTP or HTTPS communication between itself and:
  - The client browser
  - The appropriate origin server
  - Both the origin server and the client browser

- The Parallel Page Engine (PPE) runs as a servlet within the Oracle Containers for J2EE. A URL request to the servlet is forwarded through the Oracle HTTP Server's plug-in, mod_oc4j. As a standards-based plug-in, mod_oc4j communicates with Oracle Containers for J2EE using the Apache Java Protocol (AJP).
- The WSRP container is a Java portlet container that implements the WSRP specification and the Java Specifications Request (JSR) 168 APIs.
- The PPE itself makes requests to database providers, Web providers, and Web Services for Remote Portlets (WSRP) producers through HTTPS-based communication. The render request to a database provider is through a URL loopback to the Portal Services, while the call to a Web provider is by use of a SOAP-based message protocol over HTTP or HTTPS, and the call to a WSRP producer is by use of the WSRP communication protocol using the Web Services Definition Language (WSDL) URL.

> **Note:** In the current release, use of the HTTPS protocol for communication between OracleAS Portal and WSRP producers is not supported.

- If any Web providers require information from the OracleAS Metadata Repository, they issue the appropriate call through the PDK using a SOAP-based message protocol over HTTP or HTTPS.
- The OracleAS Web Cache component uses an invalidation-based cache methodology. If a requested URL can be serviced from the cache, it is assumed to be correct until the specified URL is invalidated. If a user customizes their OracleAS Portal experience, or if the privileges configured to use the user changes, the OracleAS Portal invalidates the appropriate cached objects within OracleAS Web Cache. To do this, the OracleAS Portal issues a HTTPS-based request directly from the OracleAS Metadata Repository to the invalidation port of the OracleAS Web Cache.

1.3 Understanding Caching in OracleAS Portal

OracleAS Portal caches data in the following locations:

- **Browser** – Data that does not change between requests can be cached in the browser, for example, expiry-based pages.
- **OracleAS Web Cache** – Many types of portal data are stored in this in-memory caching system. See Section 1.3.1, "Understanding OracleAS Web Cache".
- **Portal Cache** – Many types of portal data are also stored in this persistent file system-based cache. See Section 1.3.2, "Understanding Portal Cache".

OracleAS Portal uses three methods to cache Web pages and content:

- **Invalidation-based caching** is used by OracleAS Web Cache. An item remains in the cache until it is explicitly invalidated. For example, a user may update some item, requiring the cache to be invalidated. As part of the update, the OracleAS Metadata Repository or a Provider sends an invalidation message to OracleAS Web Cache. The next time there is a request for the invalidated item, it is refreshed in the cache. You can set the expiry time for invalidation-based caching. See Section 5.8.3.3, "Setting the Expiry Time for Invalidation-based Caching" for more information.

- **Validation-based caching** is used by the portal cache. Before an item in the portal cache is used, Portal Services contacts the OracleAS Metadata Repository or a Provider to determine if the cached item is still valid.
- **Expiry-based caching** is used by the portal cache, OracleAS Web Cache, and browsers. Expiry-based portlets are cached in the portal cache, whereas, expiry-based assembled pages are cached in OracleAS Web Cache. A retention period for the item specifies how long it is valid in the cache, before a refresh is required. Pages that use expiry-based caching may also be cached in the user's browser.

Caching can be done at the following levels:

- **User** - A separate copy of the data is cached for each user.
- **System** – A single copy of the data is cached for all users.

1.3.1 Understanding OracleAS Web Cache

OracleAS Web Cache is a powerful server acceleration and load balancing solution. OracleAS Web Cache is required for running OracleAS Portal. OracleAS Web Cache offers intelligent caching, page assembly, and compression features. OracleAS Web Cache accelerates the delivery of both static and dynamic Web content, and provides load balancing and failover features for Oracle Application Server.

To increase the availability and scalability of medium to large deployments, consider configuring multiple instances of OracleAS Web Cache to run as members of a cache cluster. A cluster is a collection of cooperating OracleAS Web Cache instances that work together to provide a single logical cache. Cache clusters provide failure detection and failover, increasing the availability of your Web site. If an OracleAS Web Cache instance fails, other members of the cache cluster detect the failure and take ownership of the cached content of the failed cluster member. This is achieved because the nodes that receive requests hold the content, after forwarding the request to the owner cache node.

By distributing the Web site's content across multiple OracleAS Web Cache servers, more content can be cached and more client connections can be supported, expanding the capacity of your Web site. You make use of the processing power of more CPUs and, because multiple requests are executed in parallel, you increase the number of requests that are served concurrently.

OracleAS Portal functions as a Web Cache origin server to take advantage of the following Web Cache features:

- Caching dynamically generated, user-specific page structure and portlet content
- Fine-grained cache control
- Invalidation-based caching
- Layer 7 load balancing and failover detection
- Performance assurance and surge protection

Portal sites can choose from the following deployment options:

- **Collocated:** Web Cache runs on the same physical server as the OracleAS Portal middle tier. This configuration is appropriate for smaller, low-volume sites where the scalability of the middle tier is not a concern.
- **Dedicated:** Web Cache is deployed on a dedicated server that sits in front of one or more OracleAS Portal middle-tier servers. Dedicated deployments are usually preferable to collocated deployments, as there is no risk of resource contention

with other server processes. OracleAS Web Cache performs well on commodity hardware, so a dedicated deployment does not have to be costly in terms of hardware expenditure.

For medium to large business Web sites with high volume, the dedicated topology is advantageous for the following reasons:

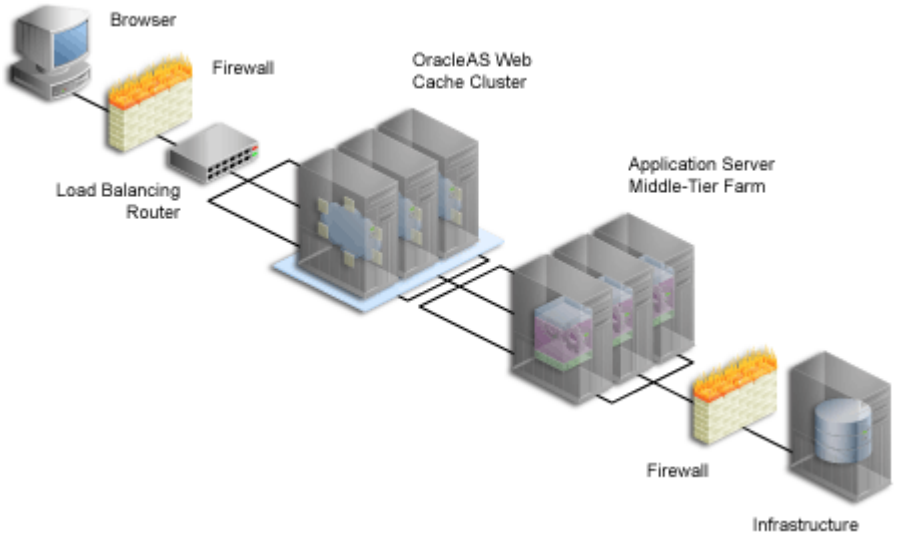
- *No resource contention*. Installing OracleAS Web Cache and OracleAS Portal middle tier on different servers will guarantee no competition among different services for hardware resources.
- *Performance assurance and surge protection*. By separating the middle-tier server and cache server, this topology minimizes compound failure rates. OracleAS Web Cache offers patent-pending techniques to guarantee site performance and scalability, even when Web server loads surpass capacity levels. A surge protection mechanism detects system overload conditions, providing a crucial buffer against traffic spikes and denial-of-service attacks.
- *Server affinity*. OracleAS Web Cache can be used to balance the load between multiple OracleAS Portal middle-tier servers and providers in a cluster. Cookies can be used to maintain persistent, or "sticky", connections to a specific server when necessary to preserve state.

See Also: Section 5.9, "Configuring OracleAS Portal to Use a Dedicated OracleAS Web Cache Instance"

To avoid a single point of failure in very high-volume sites, two or more nodes running OracleAS Web Cache may be deployed behind a Load Balancing Router (LBR). If you have multiple deployments of OracleAS Portal, each portal site can have its own Web Cache server. One or more sites can also share a single Web Cache server. Similarly, a provider can share a Web Cache with a portal site, or a dedicated Web Cache can be deployed in front of the Web server that hosts the provider. Refer to Section 5.8, "Managing OracleAS Portal Content Cached in OracleAS Web Cache" for more information about configuring OracleAS Web Cache.

In addition to providing failover, an OracleAS Web Cache cluster also balances the load it forwards to the middle tier.

Figure 1–6 Adding OracleAS Web Cache to a Medium to Large Portal Configuration



After the initial request to the owner node, the content is cached across all instances. In Figure 1–6, the LBR distributes incoming requests to the three OracleAS Web Cache instances. When the on-demand content is not available on the node receiving the request, the other instances are checked for the cached content, and the page matching the request is returned to the Browser.

To take advantage of OracleAS Web Cache's clustering capability, you must configure each instance as a member of a cache cluster. In this setup, there is no one-to-one relationship between an OracleAS Web Cache instance and a matching middle-tier instance. As shown in Figure 1–6, OracleAS Web Cache 1 provides load balancing between middle tiers 1, 2, and 3. OracleAS Web Cache 2 and 3 do the same.

See Also: *Oracle Application Server Web Cache Administrator's Guide*

You will find additional information about caching and performance on the Oracle Technology Network (OTN), http://www.oracle.com/technology/products/ias/portal/performance_10g1014.html.

1.3.2 Understanding Portal Cache

Portal cache is a file system-based cache for OracleAS Portal pages and portlets. Portal cache supports validation-based caching and expiry-based caching.

Portal cache consists of two kinds of caches:

- **Portal Content Cache**

  The content cache contains user level and system level content generated by OracleAS Portal, which includes page metadata, database portlets, Web portlets, documents, style sheets, images, and full-page caches.

- **Portal Session Cache**

  OracleAS Portal uses session cookies to maintain session details for each portal user. This session cookie is encrypted and contains important information like the database user name, lightweight user name, and Globalization Support characteristics of the session. In order for Portal Services to execute a portal request, it must get the database user name from the session cookie. To avoid an expensive decrypt operation with each user request, Portal Services decrypts the session cookie once and maintains the relevant cookie details in an in-memory session cache. The in-memory session cache may be used for garbage-collection by the JVM, and therefore, the session details are also cached in the file system.

Portal content and session cache content resides on the file system, typically under *ORACLE_HOME*/Apache/modplsql/cache, and is configured in the file *ORACLE_HOME*/Apache/modplsql/conf/cache.conf. You can specify content cache for OracleAS Portal from the Application Server Control Console. See Section 4.5.4, "Configuring the Portal Cache" for more information.

In multiple middle-tier configurations, you can set up the portal cache for each middle tier on a shared file system. This ensures that each middle tier can share cached content, rather than each drawing from its own independent cache.

For example, one middle tier might handle a request for an item by caching it in the portal cache. Because you typically use a load balancing router for configurations having multiple middle tiers, the next request for the item could be handled by a different middle tier. This middle tier could access the cached version if the portal caches for each middle tier are shared on a common file system.

Various parameters for configuring portal cache include:

- Cache location
- Total cache size
- Maximum cacheable file size
- Maximum time a cached file can be in the cache system
- Cleanup of the cache storage

See Also:

- *Oracle Application Server Performance Guide*
- **cache.conf** section in the *Oracle HTTP Server Administrator's Guide*.

1.3.3 Understanding Cache Invalidation in OracleAS Portal

OracleAS Portal makes use of two caching systems: OracleAS Web Cache, and portal cache. OracleAS Web Cache supports invalidation-based caching and expiry-based caching. The portal cache supports validation-based caching and expiry-based caching.

Cache invalidations can be classified into two groups:

- **Hard Invalidations**

  Hard invalidations are queued up over the duration of a single browser request and are then processed when the OracleAS Portal user interface action completes. The results will be seen immediately. Most page edits and all portlet customizations are treated as hard invalidations.

- **Soft Invalidations**

  Soft invalidations are queued up over many browser requests and are then processed later by the soft invalidation database job. Security related changes, for example, granting privileges on a page to a user or group, are treated as soft invalidations.

Cache Invalidation Resource Requirements

Invalidations are queued up based on edits and personalizations. With more such actions being performed, a greater number of invalidations are submitted. Individual actions that involve more portal objects or users will require more resources to process the corresponding invalidations. For example, changing the access privileges for a group of users will require that data for each user in the group be invalidated. Therefore, the larger the group the more invalidation resources will be needed. Consider another example, deleting a large number of pages as a bulk operation requires invalidation resources proportional to the number of pages being deleted. Processing of invalidations requires storage, CPU, and communication resources. Therefore, large numbers of cache invalidations may slow down the system. The reason for this could be any of the following:

- Communication with OracleAS Web Cache

  When either hard or soft invalidations are processed, a TCP/IP connection is established with the OracleAS Web Cache invalidation port from the OracleAS Metadata Repository, to send invalidation messages.

  For both hard and soft invalidations, all the messages queued are sent using a TCP/IP connection to OracleAS Web Cache. OracleAS Web Cache receives these

invalidation messages and attempts to invalidate cached data. This load may affect OracleAS Web Cache's ability to respond to requests for data.

- Cache invalidation queue storage

  Both hard and soft invalidation messages are queued into a database table in the OracleAS Metadata Repository. As the queue grows in size, more database resources are required to maintain the queue.

- Cache invalidation queue optimization

  During the processing of hard or soft invalidation messages, queue optimization removes duplicate or unnecessary invalidation messages. For example, if a page group is being invalidated, individual invalidation messages for pages in the page group are unnecessary. If a large number of invalidation messages have been queued up, the optimization process may take a long time.

1.4 Understanding WSRP and JPS

The WSRP specification is a Web services standard that allows the plug-and-play of visual, user-facing Web services with portals or other intermediary Web applications. Being a standard, WSRP enables interoperability between a standards-enabled container based on a particular language (such as JSR 168, .NET, Perl) and any WSRP portal. Therefore, a portlet (regardless of language) deployed to a WSRP-enabled container can be rendered on any portal that supports this standard. For more information about WSRP, see http://www.oasis-open.org/committees/download.php/2877/wsrp-specification-1.0-cs-1.0-rev2.pdf.

JPS is based on JSR 168 and defines a set of APIs for building standards-based portlets using Java. Portlets built to this specification can be rendered to a portal locally or deployed to a WSRP container for rendering portlets remotely. For more information about JSR 168, see http://jcp.org/aboutJava/communityprocess/first/jsr168/index.html.

See Also: *Oracle Application Server Portal Developer's Guide*

1.4.1 What's Next?

You are ready to move on to Chapter 2, "Planning Your Portal", now that you have a basic understanding of the Oracle Application Server architecture, how OracleAS Portal fits in, the working of caching in OracleAS Portal, and WSRP producers. By the end of that chapter, you should have a good idea of how you want to configure your installation.

2

Planning Your Portal

This chapter details the task flow involved in planning, installing, configuring, and administering Oracle Application Server Portal. After reading this chapter, you should understand how to plan the hardware and software you need to effectively build a portal.

This chapter contains the following sections:

- What Do I Need to Consider?
- What Do I Need to Do?

> **Note:** You may want to review Chapter 1, "Understanding the OracleAS Portal Architecture" if you are unfamiliar with the terms used in this chapter.

2.1 What Do I Need to Consider?

To develop a plan for configuring your portal, it is critical that you have a firm grasp of the goals you want your system to achieve. Take a look at the following sections to see what's involved in each of these crucial decision points:

- Which Topology Is Right for Me?
- How Much Hardware Do I Need?
- How Can I Maximize Performance?
- How Can I Make My Portal Scale?
- How Can I Make My Portal Highly Available?
- How Can I Secure My Portal?
- How Should I Configure My Hardware and Software?
- Getting the Most Out of Your Configuration

2.1.1 Which Topology Is Right for Me?

Oracle Application Server offers a variety of topology options. The Oracle Application Server recommended topologies range from small general development implementations to very large enterprise-wide implementations.

> **See Also:** Overview of the recommended topologies in the *Oracle Application Server Concepts* guide located in the Oracle Application Server 10*g* Documentation Library.

2.1.2 How Much Hardware Do I Need?

Servers, databases, and resources supporting your Web portal must handle wide variations in user traffic, especially during peak intervals.

As with any Web portal, the server and database capacity with which you will need to deploy a portal largely depends on the number of user requests that you anticipate. Displaying a single page to a user may require many separate transactions, from verifying whether the user has permission to view the page, to loading the images that appear on the page, to calling a style sheet that contains formatting information for the page.

The upper and lower limits of what you will need are determined by how you expect your users to use the portal. At a minimum, you will need enough server capacity to satisfy the average load during a work day, with response times that are acceptable to your user base. If possible, you should strive to satisfy the volume of page requests you anticipate during peak intervals of high user activity. Hardware resources such as CPU, memory, I/O capacity, and network bandwidth are key to reducing response times. You must install OracleAS Portal on a server or group of servers that can handle a large number of transactions, or your users will experience slow response times.

Adding more servers and database capacity will certainly improve your Web portal's performance, but unless you have unlimited funds at your disposal, you will need to balance good performance against the costs configured to use each new piece of hardware and software.

See Also: *Oracle Application Server Administrator's Guide*

2.1.3 How Can I Maximize Performance?

Response time is the time between the receipt of a user request and the completion of the response to the request. Your Web portal should respond as quickly as possible with the least amount of software and hardware overhead. Some performance considerations are:

- **Distributing the load**

  If you anticipate a heavy volume of traffic on your Web portal, you can distribute the load across multiple servers, each with its own middle-tier instance. If one server is overloaded with too much traffic, a second server can handle the overflow. See Section 2.1.8.1, "Load Balancing" for more information.

- **Protecting against failures**

  A distributed OracleAS Portal configuration offers improved performance over a single server configuration because you are making more software and hardware resources available to the Web portal. You can use additional servers and software to provide *failover*, thus ensuring system stability. See Section 2.1.8.2, "Failover and Redundancy" for more information.

- **Implementing cache clusters**

  To increase the availability and scalability of medium to large deployments, you can configure *cache clusters*. Cache clusters provide failure detection and failover, increasing the availability of your Web site. See Section 1.3, "Understanding Caching in OracleAS Portal" for more information.

  See Also: *Oracle Application Server Performance Guide*

- **Choosing optimal cache options for pages, portlet instances, and templates**

To improve performance and scalability, OracleAS Portal provides several caching options. For more information, refer to the chapter about improving page performance in the *Oracle Application Server Portal User's Guide*.

2.1.4 How Can I Make My Portal Scale?

Clustering enables you to scale your system beyond the limitations of a single application server instance on a single host. A cluster unifies multiple application server instances spread over multiple hosts to collectively serve a single group of applications. In this way, clustering makes it possible to serve increasing numbers of concurrent users after the capacity of a single piece of hardware is exhausted.

In addition, you can improve scalability by choosing optimal cache options for pages, portlet instances, and templates. OracleAS Portal provides several caching options. For more information, refer to the chapter about improving page performance in the *Oracle Application Server Portal User's Guide*.

Refer to Section 2.1.8.3, "Scalability" and Section 1.3, "Understanding Caching in OracleAS Portal" for more information.

2.1.5 How Can I Make My Portal Highly Available?

Clustering also enables you to achieve a higher level of system availability than is possible with only a single application server instance. An application running on a single instance of an application server is dependent on the operating system and host on which the server is running. In this case, the host poses as a single point of failure because if the host goes down, the application becomes unavailable.

Application server clusters enable higher availability by providing redundancy and backup and eliminating a single point of failure. Clients access the cluster through a load balancing router that can send requests to any application server instance in the cluster. In the case that an application server instance becomes unavailable, the load balancing router can continue forwarding requests to the remaining application server instances, as any instance can service any request.

See Also: *Oracle Application Server High Availability Guide*

2.1.6 How Can I Secure My Portal?

Sensitive data should be secured without affecting content that you want to make available to all users.

To support a flexible approach to controlling access to Web content, OracleAS Portal leverages other components of Oracle Application Server and Oracle Database 10*g* to provide strong protection for your portal. OracleAS Portal interacts with all of the following components to implement its security model:

- Oracle Application Server Single Sign-On
- mod_osso, an Oracle HTTP Server listener module, which facilitates Single Sign-On for J2EE web applications.
- Oracle Application Server Web Cache
- Oracle Internet Directory
- Oracle Delegated Administration Services
- Oracle Directory Integration Platform

See Chapter 6, "Securing OracleAS Portal" for more information.

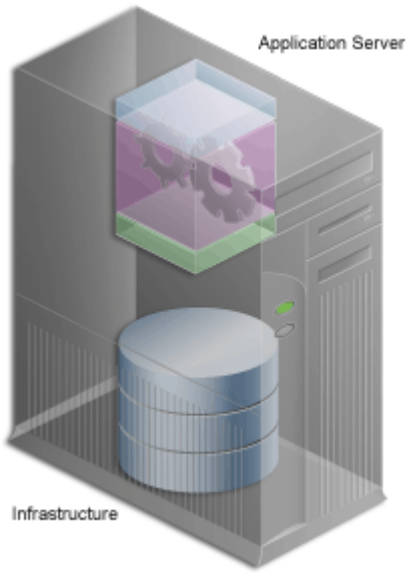
2.1.7 How Should I Configure My Hardware and Software?

This section discusses how you should configure your hardware and software installations for optimal use of OracleAS Portal and all related Oracle Application Server components. This section explains how you can configure your hardware to set up a small development environment, and deploy larger sites serving many users.

2.1.7.1 Using a Single Computer

In the simplest configuration, all of the component pieces (application server and infrastructure) are installed on a single computer as shown in Figure 2–1. In fact, a single database could also reside on the computer, containing separate schemas for OracleAS Portal, Oracle Internet Directory, and OracleAS Single Sign-On.

Figure 2–1 OracleAS Portal Single Computer Configuration



This configuration works nicely in a small development environment in which your developers are using OracleAS Portal's declarative interface to build pages, portlets and applications. It also easily supports a small deployment of the finished Web portal. If you expect to deploy a larger site that delivers more content to more users, you will need more than a single server or the simple configuration shown in Figure 2–1.

2.1.7.2 Using Multiple Computers

If a single computer configuration does not suit your needs, consider moving the various pieces of the OracleAS Portal architecture to other computers. When configuring your Web portal, you will require more servers depending on the size of your site, where each server performs specialized work. Adding extra hardware increases performance, and adding more software instances supports *redundancy*.

Deployment options for configuring larger Web portal sites include:

- Separating the Middle Tier from the OracleAS Metadata Repository
- Installing Oracle Identity Management Separately
- Adding Middle-Tier Instances
- Installing OracleAS Web Cache Separately from the Middle Tier

- Configuring High Availability for the Infrastructure

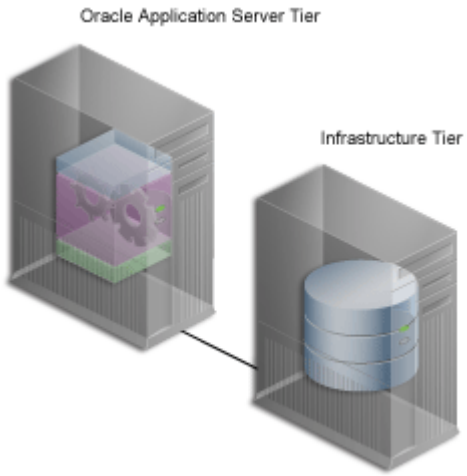
These tasks should be performed in the order they appear in on this list until you are satisfied that your configuration can handle the demands of your deployed Web portal. If your site must handle only a moderate workload, you could first separate the middle tier from the database, then consider moving Oracle Identity Management to another server. You probably will not need to perform all of these configuration tasks. But as the site grows, you should expand its underlying configuration by following the sequence shown in this list.

> **Note:** Before you go online with your Web portal, it's a good idea to set up and test a small pilot system. This enables you to gather valuable configuration and tuning information based on real usage patterns, without affecting the users you plan to serve.

2.1.7.2.1 Separating the Middle Tier from the OracleAS Metadata Repository The first thing you should consider when configuring a larger system is installing the middle tier separately, as shown in Figure 2–2.

> **See Also:** *Oracle Application Server Performance Guide*

Figure 2–2 Separating the Application Server Middle Tier from the Infrastructure



This frees the OracleAS Metadata Repository and the middle tier from having to compete for hardware resources, such as I/O, memory, and disk space. Installing them on separate computers also gives you more flexibility in performance tuning. This is important for sites that plan on storing a lot of content in the OracleAS Metadata Repository. Tuning parameters, such as those for an operating system, are different from those for middle-tier components such as the HTTP server. Setting a performance parameter for one may not provide optimal performance for another.

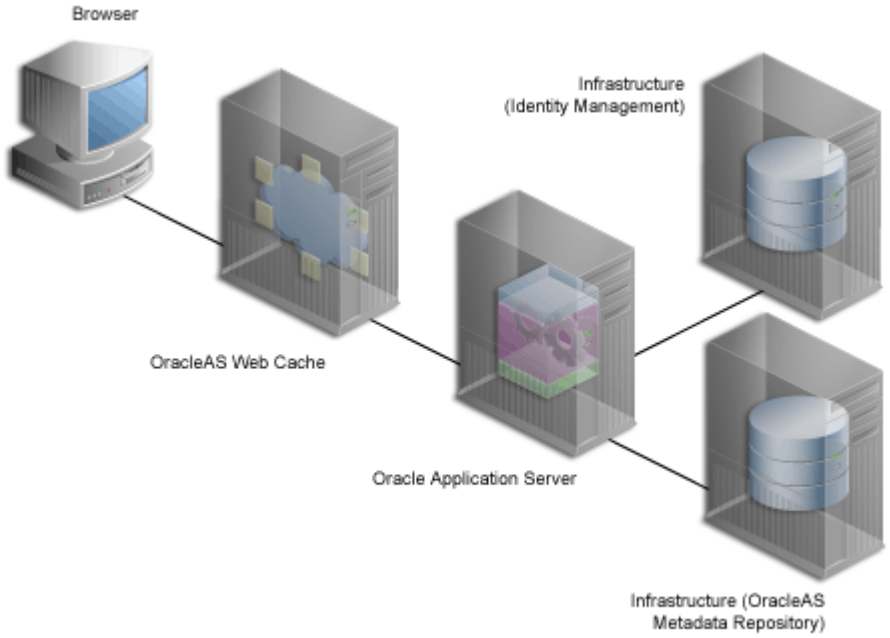
In 10*g* Release 2 (10.1.4), the Oracle Universal installer can install a new database seeded with the OracleAS Metadata Repository, or it can use an existing database (customer database). If you want to use an existing database, you need to run the new Oracle Application Server Repository Creation Assistant tool, available on the OracleAS RepCA CD-ROM, to populate the existing database with the OracleAS Metadata Repository. You do this before running the installer to install other Oracle Application Server components.

2.1.7.2.2 Installing Oracle Identity Management Separately OracleAS Single Sign-On authenticates user credentials against Oracle Internet Directory for OracleAS Portal and other applications, thus requiring users to log on to the Web portal only once with a single user name and password, to enable access to multiple accounts and applications.

Once users have logged in to a deployed OracleAS Portal site, they can access any other OracleAS Single Sign-On secured application from portlets within the portal.

As shown in Figure 2–3, Oracle Identity Management is located on a different computer from the OracleAS Metadata Repository. A single instance of Oracle Identity Management can be configured to work with multiple Oracle products, including multiple instances of the OracleAS Portal middle tier.

Figure 2–3 Oracle Identity Management Installed on a Separate Computer



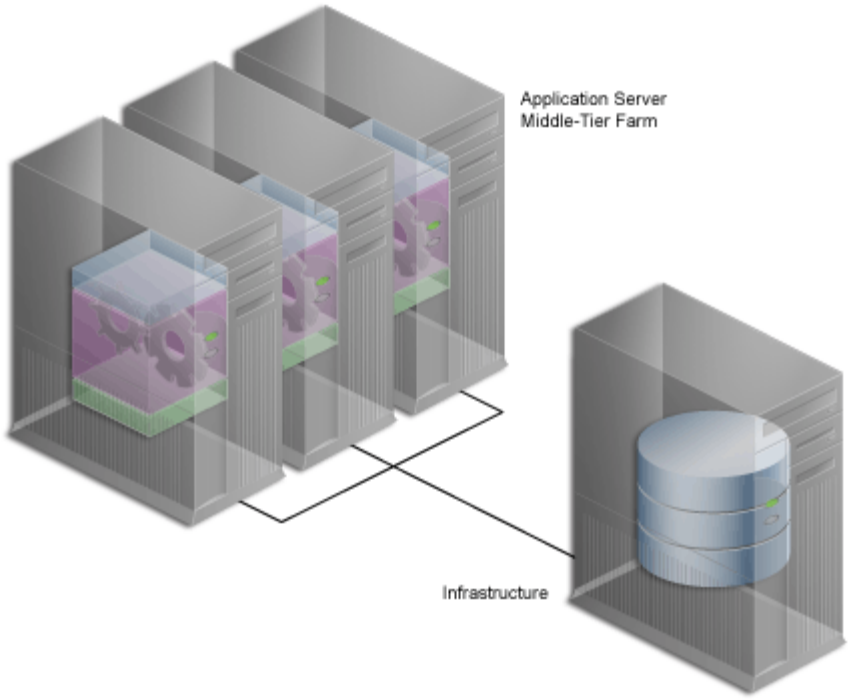
The system shown in Figure 2–3 is an example of a *distributed configuration.* The configuration includes a centralized Oracle Identity Management server that could support multiple middle-tier instances. Moving Oracle Identity Management to its own server gives you the flexibility to tune its performance independently of the database and middle tier.

In addition, isolating Oracle Identity Management from middle-tier installations ensures greater stability for the entire distributed system. If the computer where a middle tier is installed fails, OracleAS Single Sign-On and other middle-tier instances that rely on it to validate logins are not affected. Additionally, different security policies can be used to manage the various computers in the configuration.

See Also: *Oracle Application Server Installation Guide*

2.1.7.2.3 Adding Middle-Tier Instances You can add redundant middle-tier instances, each with identical configuration settings, to support the largest Web portals. The added middle-tier instances are shown in Figure 2–4. It is a good idea to install each middle-tier instance on its own computer to isolate any hardware failures.

Figure 2–4 Multiple Middle Tiers



The middle tier forwards user requests for portal pages to a provider, then assembles the pages with the returned content. As you add more middle-tier instances to your OracleAS Portal configuration, you increase the capacity for user requests and improve the overall performance of your portal. In addition, because the middle tier performs some processing before forwarding a request, less time is spent sending and receiving data over the network. Database and network resources are used more efficiently.

> **Note:** For this configuration, you must use a Load Balancing Router (LBR). See Section 2.1.8.1, "Load Balancing" for more information.

2.1.7.2.4 Installing OracleAS Web Cache Separately from the Middle Tier You can also separate the OracleAS Web Cache server from the middle tier to enable better caching of data, faster request times, and reduction in the load on the middle tier. This also improves the performance of OracleAS Portal.

2.1.7.2.5 Configuring High Availability for the Infrastructure In Oracle Application Server 10*g*, all Oracle High Availability (HA) solutions, including Cold Failover Cluster, Data Guard, and Real Application Clusters (RAC), are supported for the OracleAS Infrastructure.

> **See Also:** *Oracle Application Server High Availability Guide*

2.1.8 Getting the Most Out of Your Configuration

A distributed OracleAS Portal configuration offers improved performance over a single computer configuration because you are making more software and hardware resources available to the Web portal. But there are other benefits. You can use additional servers and software to provide *failover*, thus ensuring system stability. And you can deal with wide fluctuations in the amount of work your Web portal is

expected to perform over the course of a day using *load balancing* between multiple servers. Finally, you can add more servers to a distributed configuration to support more users, thus providing *scalability*.

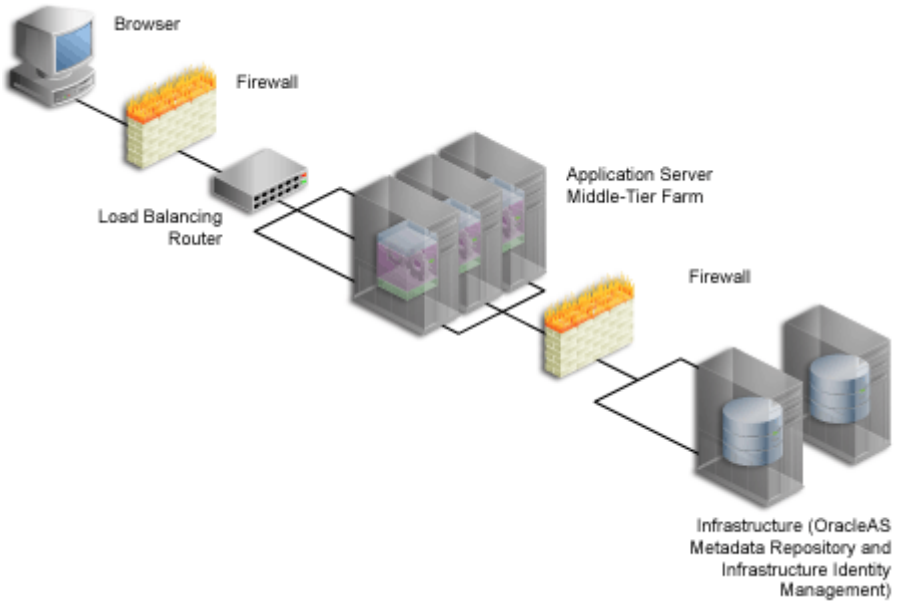
2.1.8.1 Load Balancing

If you anticipate a heavy volume of traffic on your Web portal, you can distribute the load across multiple servers, each with its own middle-tier instance. If one server is overloaded with too much traffic, a second server can handle the overflow.

Oracle Application Server provides its own load balancing capability by pooling server instances to service incoming requests. If one instance does not respond, then the request is forwarded to another instance. This ensures that content and applications are always available to users of your deployed site.

For very large sites, you can add a Load Balancing Router (LBR) to distribute incoming requests to the middle-tier servers, as shown in Figure 2–5. An LBR is a very fast network device that distributes network requests across a large number of servers. It provides users of your portal with a single published address, instead of them having to send each request to a particular middle-tier server.

Figure 2–5 Multiple Server Configuration Using a Load Balancing Router



See Section 5.3, "Configuring Multiple Middle Tiers with a Load Balancing Router" for more information on adding an LBR to distribute incoming requests to middle-tier servers.

As an example, the high traffic personal site, My.Oracle.com (MOC), uses an LBR to sort requests. Because the software logic for distributing loads is contained in the LBR itself rather than installed separately on each individual middle-tier server, an LBR lowers the overall administrative costs of your configuration. MOC is both an intranet and extranet Web site. It provides Oracle employees with a single customizable entry point to all of Oracle's online services as well business information from external providers.

Adding an LBR can also help your configuration deal with load variations. Users may access your site, use its applications, and request content at a much higher frequency during certain peak intervals, for example, between 9 a.m. and 10 a.m. when most

users log on to begin their work day. During these periods of heavy traffic, the LBR can distribute page requests among the various middle-tier instances to ensure quick response times.

If your peak load occurs on a regular basis, consider a configuration that specifically addresses the need to handle peak load requirements. If your peak load is infrequent, you may be willing to tolerate slower response times at peak intervals rather than spend additional money on hardware.

Note that the LBR itself can be configured to support failover. The My.Oracle.com configuration in Figure 2–6 could add a second LBR, which would be available in case the primary router fails.

2.1.8.2 Failover and Redundancy

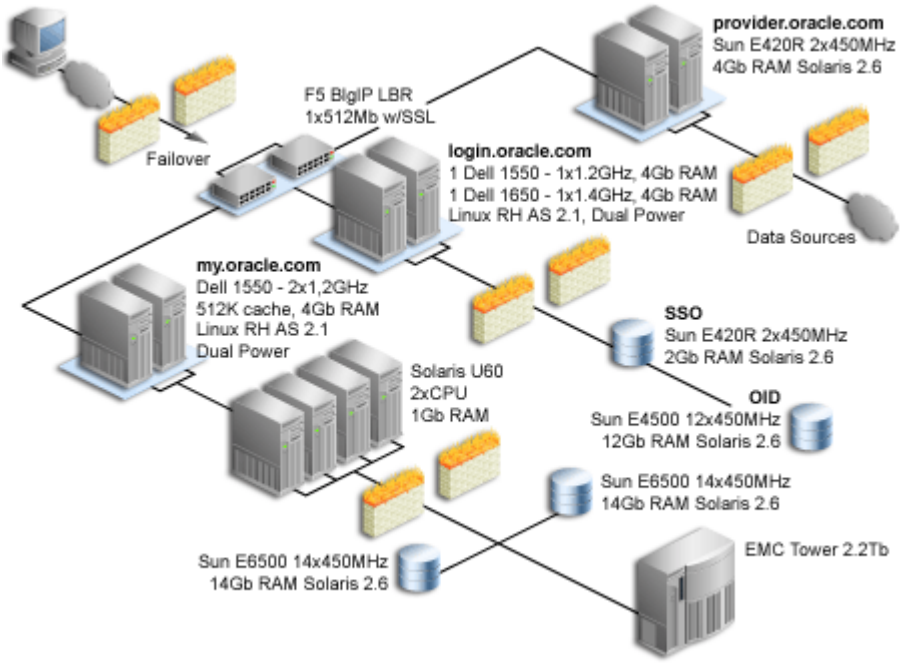
Failover is the ability to switch to a backup when part of your system fails, such as a server or database. When an Oracle Database 10*g* fails, for example, it restarts using any preserved state information from the backup.

Redundancy is the technique of providing duplicate computers configured identically. The redundant computers provide enough capacity to service requests, and provide backups in case of failures and errors. You implement redundancy by increasing the number of computers in your configuration. One server is typically active while the other monitors the first server's activity, ready to take over if it fails.

As shown in Figure 2–6, My Oracle.com provides for failover using an additional middle-tier server that can take over if any of the other servers encounter problems that cause them to fail.

Note: The components depicted in Figure 2–6 represent only one of many possible configurations. Oracle does not expressly recommend or endorse these specific vendors, or components, or both.

Figure 2–6 My Oracle.com Middle-Tier Configuration



To set up redundant middle-tier instances, you configure the original and each redundant instance with identical server name and server port entries, for example, `my.oracle.com` and port `5000`.

One alternative to redundancy is to set up failover by using any excess capacity that you have in your overall configuration. For example, you might have four middle-tier servers, each running at 75% capacity. If one server fails, the other three can take over the workload of the fourth (25% X 3 = 75%, which is the capacity of the failing server).

2.1.8.3 Scalability

Scalability is the ability of a Web portal to handle more requests as the number of users and the volume of content increases over time. As the portal handles more traffic, users should not notice any change in performance, as measured by response intervals and frequency of errors. If scalability is your goal, you need a flexible configuration that will enable you to add database capacity and servers incrementally as needed without adversely affecting the rest of your configuration.

When My.Oracle.com (MOC) was set up, for example, it was initially expected to serve approximately 40,000 Oracle employees. The user base is anticipated to expand eventually to a million and a half, most of them users of the Oracle Technology Network (OTN), each automatically provided with an MOC account.

2.2 What Do I Need to Do?

This section describes the task flow involved in planning, installing, configuring, and administering OracleAS Portal.

Successfully deploying OracleAS Portal consists of the following steps:

1. Planning Your Portal
2. Upgrading OracleAS Portal (if necessary)
3. Verifying Pre-Installation Requirements
4. Installing Oracle Application Server
5. Performing Post-Installation Configuration (basic configuration and administration)
6. Performing Advanced Configuration
7. Securing OracleAS Portal
8. Monitoring OracleAS Portal
9. Troubleshooting OracleAS Portal

The following sections provide high-level descriptions of each step and point to more detailed information in various locations, including this configuration guide, other Oracle Application Server 10*g* Documentation Library books, technical white papers, and OTN, `http://www.oracle.com/technology/`.

2.2.1 Planning Your Portal

If you are new to OracleAS Portal, you may benefit from reading Chapter 1, "Understanding the OracleAS Portal Architecture" to understand how OracleAS Portal fits into the Oracle Application Server architecture.

You can find more information about planning your OracleAS Portal configuration on Portal Center at `http://portalcenter.oracle.com`.

2.2.2 Upgrading OracleAS Portal

You will find the latest information on upgrading from an earlier release of OracleAS Portal on OTN, http://www.oracle.com/technology/products/ias/portal/upgrade.html. On the Upgrade page, you will find:

- Instructions for downloading the upgrade scripts.
- Online upgrade documentation.

2.2.3 Verifying Pre-Installation Requirements

To ensure a smooth installation, you must verify that you have fulfilled all prerequisites and have performed all pre-installation steps. The *Oracle Application Server Installation Guide* contains the general Oracle Application Server requirements, while Chapter 3, "Installing OracleAS Portal" discusses the portal-specific steps.

2.2.4 Installing Oracle Application Server

The *Oracle Application Server Installation Guide* contains the steps for installing the Oracle Application Server middle tier and infrastructure required to run OracleAS Portal. Refer to Chapter 3, "Installing OracleAS Portal" for additional information.

2.2.5 Performing Post-Installation Configuration

Chapter 4, "Performing Basic Configuration and Administration" contains information about all the post-configuration tasks that can be performed by the OracleAS Portal administrator.

You can find additional information on Portal Center at, http://portalcenter.oracle.com.

2.2.6 Performing Advanced Configuration

Part III, "Advanced Configuration Topics" is targeted at the Oracle Application Server administrator. Chapter 5, "Performing Advanced Configuration" provides instructions on how to perform more advanced OracleAS Portal configuration and integration configuration, including virtual hosts, load balancing routers, proxy server, OracleAS Web Cache, and OracleAS Single Sign-On configuration. Other chapters in Part III deal with setting up features such as search, import and export, and more.

2.2.7 Securing OracleAS Portal

Chapter 6, "Securing OracleAS Portal" contains in-depth information on how to configure the security features in OracleAS Portal.

2.2.8 Monitoring OracleAS Portal

You can monitor OracleAS Portal through the Oracle Enterprise Manager 10*g* Application Server Control Console. For details, see Chapter 7, "Monitoring and Administering OracleAS Portal".

In addition, you can generate performance reports to monitor portal performance. See Section 9.5, "Generating Performance Reports". This performance information will be useful later on, for tuning OracleAS Portal performance. See Chapter 9, "Tuning Performance in OracleAS Portal".

2.2.9 Troubleshooting OracleAS Portal

Appendix K, "Troubleshooting OracleAS Portal" discusses various issues and ways for resolving and diagnosing problems.

Refer to the *Oracle Application Server Portal Error Messages Guide*, for more information on error messages.

Part II

Installation and Basic Configuration

Part two contains the following chapters:

3

Installing OracleAS Portal

This chapter provides a brief overview of what is installed and configured, by default, in the process of installing OracleAS Portal. For complete instructions on how to install and configure the infrastructure and the middle tier in different topologies, refer to the *Oracle Application Server Installation Guide*.

This chapter contains the following sections:

If you are planning to upgrade OracleAS Portal from a previous release, you will need to refer to the Upgrade documentation on the Oracle Technology Network (OTN), `http://www.oracle.com/technology/products/ias/portal/upgrade.html`.

3.1 What Is Installed and Configured By Default?

A new installation of OracleAS Portal 10*g* Release 2 (10.1.4) consists of three phases:

- Oracle Application Server Infrastructure Release 10.1.2.0.2 Installation
- Oracle Application Server Middle-tier Release 10.1.2.0.2 Installation
- Oracle Application Server Portal Upgrade to 10*g* Release 2 (10.1.4)

Oracle Application Server Infrastructure Release 10.1.2.0.2 Installation

The OracleAS Infrastructure installation consists of the Oracle Application Server Single Sign-On, Oracle Internet Directory, Oracle Delegated Administration Services, and the Oracle Application Server Metadata Repository.

The OracleAS Metadata Repository component of Oracle Application Server 10*g* creates a new database and populates it with a collection of schemas used by Oracle Application Server components, such as the OracleAS Portal metadata schema during an infrastructure installation.

If you plan to install the OracleAS Metadata Repository, including the OracleAS Portal schema, in a customer database, you need to run the Oracle Application Server Repository Creation Assistant tool, available on the OracleAS RepCA CD-ROM, to populate the existing database with the OracleAS Metadata Repository. You must do this before running the installer to install other Oracle Application Server components. Refer to the *Oracle Application Server Installation Guide* for more information.

When you install OracleAS Portal, some default database schemas and user accounts are also installed. Refer to Section 6.3.1, "Configuring OracleAS Portal Security Options" in Chapter 6, "Securing OracleAS Portal" for a description of the default database schemas.

Oracle Application Server Middle-tier Release 10.1.2.0.2 Installation

During the Oracle Application Server middle-tier installation, OracleAS Portal is configured to use the infrastructure services. The deployment of the portal applications in the middle tier also occurs at this time. The following steps are performed at this time:

1. OracleAS Portal is added as a partner application to OracleAS Single Sign-On.
2. OracleAS Web Cache configuration information is stored in the OracleAS Portal schema in the OracleAS Metadata Repository.
3. User and Group information is created in Oracle Internet Directory.
4. The Oracle Application Server Provider Group is added to OracleAS Portal.
5. OracleAS Portal Service Monitoring is configured.
6. The Provider user interface is configured to work with OracleAS Portal. The middle-tier URL, used to access the provider user interface framework from OracleAS Portal is added to the global settings page.
7. The default Web providers, OmniPortlet and Web Clipping, are registered.
8. The Web Services for Remote Portlets (WSRP) container is installed.

   The portal repository contains the default preference store of this WSRP container. Refer to the *Oracle Application Server Portal Developer's Guide* for more information.

   Note: OracleAS Portal supports communication with any WSRP producer.

9. A sample JSR 168 application is installed to run the WSRP container.
10. The Wireless configuration information is configured.
11. Oracle Text and Oracle Ultra Search are configured.
12. An entry for OracleAS Portal is created in the Oracle Enterprise Manager 10*g* `targets.xml` file.
13. The configuration file *`ORACLE_HOME`*`/config/ias.properties` is updated.
14. The OracleAS Portal DAD is created in the configuration file, *`ORACLE_HOME`*`/Apache/modplsql/conf/dads.conf`, using the parameters provided at installation time.
15. The Portal Dependency Settings file `iasconfig.xml` is created. See Section A, "Using the Portal Dependency Settings Tool and File" for details.
16. The file `cache.xml` is created.

    `cache.xml` stores OracleAS Web Cache invalidation settings and it is used by Web providers such as OmniPortlet and Web Clipping. See the *Oracle Application Server Portal Developer's Guide* for more information.

The details for these steps are available, after the installation, in the file *`MID_TIER_ORACLE_HOME`*`/assistants/opca/install.log`.

Refer to Section 6.1.2.1, "OracleAS Portal Default, Seeded User Accounts" and Section 6.1.2.2, "OracleAS Portal Default, Seeded Groups" for a description of the OracleAS Portal default user accounts and groups.

Out of the box, the initialization parameters for this new database are suitable for a very small OracleAS Portal configuration with few users. If you plan to use OracleAS Portal, it is recommended that you modify the initialization parameters for the database based on the requirements for installing the OracleAS Metadata Repository in an existing database, using the settings specified in the *Oracle Application Server Installation Guide*. As you make changes in your configuration, you may need to further tune the initialization parameters based on the size of your configuration, and the number of simultaneous users of OracleAS Portal.

Oracle Application Server Portal Upgrade to 10*g* Release 2 (10.1.4)

When you upgrade to OracleAS Portal 10*g* Release 2 (10.1.4), only the portal schema in the OracleAS Metadata Repository gets upgraded. Refer to the *Oracle Application Server Portal Installation and Upgrade Guide* for more information about the steps performed during an upgrade. This guide is classified into two parts and depending on whether you are installing OracleAS Portal for the first time, or, are upgrading from an earlier release of OracleAS Portal, refer to the relevant part in the guide as follows:

- Refer to the part titled "Installation" if you are installing an OracleAS Portal instance for the first time.
- Refer to the part titled "Upgrade" if you are upgrading from an earlier release of OracleAS Portal.

3.2 Accessing OracleAS Portal After Installation

This section details the steps you should take to access OracleAS Portal after installation:

1. After you install Oracle Application Server, go to the Oracle Application Server page at `http://hostname.domain:port`. Here you can view the documentation library, take the Quick Tour, and run demos. To run the demos, click the **Demonstrations** tab and then select **Portal and Wireless** from the Navigation panel.
2. Access OracleAS Portal by entering the following URL in your browser:

   `http://<host>:<port>/portal/pls/<dad>`

   For example:

   `http://portal.mycompany.com:7779/portal/pls/portal`

   The **Portal Builder** page is displayed.

   Table 3–1 explains the components that make up the URL used to access OracleAS Portal.

Table 3–1 Portal URL Descriptions

Parameter	Description
`host`	Defines the computer on which you installed OracleAS Portal. Enter both the hostname and the fully qualified domain name. For example, enter `host.domain.com`. This name must also match the `ServerName` parameter in the configuration file, `httpd.conf`, located in: *ORACLE_HOME*`/Apache/Apache/conf`
`port`	Defines the port number to access OracleAS Portal.
`portal`	Specifies that the request should be routed to the Portal Services running inside OracleAS Portal OC4J. **Note:** In earlier versions, the OracleAS Portal URL was of the format `http://<host>:<port>/pls/<dad>`. Backward compatibility is provided for such URLs using rewrite rules in the Oracle HTTP Server configuration, so that URLs of the `/pls/<dad>` format are rewritten to `/portal/pls/<dad>`. When URLs of the older format are accessed, OracleAS Portal either services the URL directly or alerts you to change the bookmarked URL to the new format.
`pls`	Specifies that the request is for a PL/SQL procedure.
`dad`	Defines the Database Access Descriptor (DAD) you specified earlier for your OracleAS Portal installation. The DAD contains information on how to connect to the database. In a typical default installation, the DAD is 'portal'.

3. Click the **Login** link, located in the top right corner as shown in Figure 3–1:

Figure 3–1 Login Link

4. Log in as the *portal* user, using the ias_admin password.

> **Note:** OracleAS Portal creates its users in Oracle Internet Directory only once, based on the OracleAS Portal schema in the Application Server Metadata Repository. Subsequent middle-tier installations that use the services of the same portal schema in the Application Server Metadata Repository do not create or update users in Oracle Internet Directory. Therefore, the portal password is the ias_admin password of the first middle tier that uses the services of the Application Server Metadata Repository.

5. After you have verified that OracleAS Portal is up and running, by logging in, you can run the OracleAS Portal Diagnostic Assistant (PDA) and view the generated reports for additional verification. Refer to Section K.2.5, "Using OracleAS Portal Diagnostics Assistant" for instructions on how to run the PDA.

3.3 Configuring OracleAS Portal During and After Installation

During a middle-tier installation that includes OracleAS Portal, you can specify if you want to configure, and automatically start OracleAS Portal at the end of the installation. If you select that option, Oracle Universal Installer (OUI) will configure OracleAS Portal in two phases:

1. OracleAS Portal middle-tier deployment
2. OracleAS Portal schema configuration in the OracleAS Metadata Repository

If you choose not to configure OracleAS Portal, and want to do this later, you need to:

- Use Oracle Enterprise Manager 10*g* Application Server Control Console to deploy OracleAS Portal on the middle tier. Refer to Section 7.2.2, "Using Application Server Control Console to Configure OracleAS Portal" for more information.
- Use the Portal Dependency Settings file and tool, to perform the OracleAS Portal schema configuration in the OracleAS Metadata Repository. This step is necessary because, when you use Application Server Control Console, the OracleAS Metadata Repository is not automatically configured and existing configuration entries on the OracleAS Metadata Repository are overwritten. Refer to Appendix A, "Using the Portal Dependency Settings Tool and File" for more information.

You can update the OracleAS Metadata Repository with any changes made to the Portal Dependency Settings file `iasconfig.xml`, related to middle-tier component properties, such as OracleAS Web Cache, and Oracle Enterprise Manager 10*g*.

OracleAS Portal does not support serving two middle tiers from a single repository, unless it is front-ended by a load balancing router (LBR). Refer to Section 5.3, "Configuring Multiple Middle Tiers with a Load Balancing Router" for instructions on how to set up OracleAS Portal with an LBR. If you want to add additional middle tiers to a farm that is already using OracleAS Infrastructure Services, you do not want to overwrite the existing configuration entries during the deployment. In this case, you would install the additional middle tier without configuring OracleAS Portal, then configure OracleAS Portal, using Application Server Control Console, and finally update the Portal Dependency Settings file.

Note: By default, `iasconfig.xml` resides in *ORACLE_HOME*`/portal/conf`. If the Portal Dependency Settings file is accessible over a network file system, you can share the file across multiple hosts, avoiding the need to manually replicate it every time the file is modified. If the installation is running on an operating system that supports symbolic links, it is recommended that you use this mechanism to reference a shared file. If, however, the Portal Dependency Settings file is not accessible over the network, you must ensure that the file is kept up-to-date with changes to your site topology. Refer to Section A.2.4, "Updating the Portal Dependency Settings File" for more information.

Follow the steps outlined in Section 7.2.2, "Using Application Server Control Console to Configure OracleAS Portal" to use Application Server Control Console to deploy OracleAS Portal on the middle tier.

At this point, your OracleAS Portal middle-tier components are deployed and configured. The DAD has been created, and the Portal Dependency Settings file `iasconfig.xml` has been updated.

To update the OracleAS Metadata Repository with any changes made to the Portal Dependency Settings file `iasconfig.xml`, run the script `ptlconfig`, located in the directory *`ORACLE_HOME`*`/portal/conf`, as follows:

```
ptlconfig -dad portal
```

Additional middle tiers are often added to production sites, to improve scalability. The two-phased process described in the preceding text allows the flexibility of adding additional middle tiers, without system downtime.

4

Performing Basic Configuration and Administration

This chapter assumes that OracleAS Portal has been installed as part of the Oracle Application Server and addresses the basic tasks that the portal administrator can perform after installation is complete.

This chapter contains the following sections:

4.1 Getting Started with OracleAS Portal Administration

Basic OracleAS Portal configuration can be performed on the **Administer** tab available from OracleAS Portal. Additionally, there are other administrative tools available to configure OracleAS Portal and its related components.

This section will introduce you to the various different administrative tools:

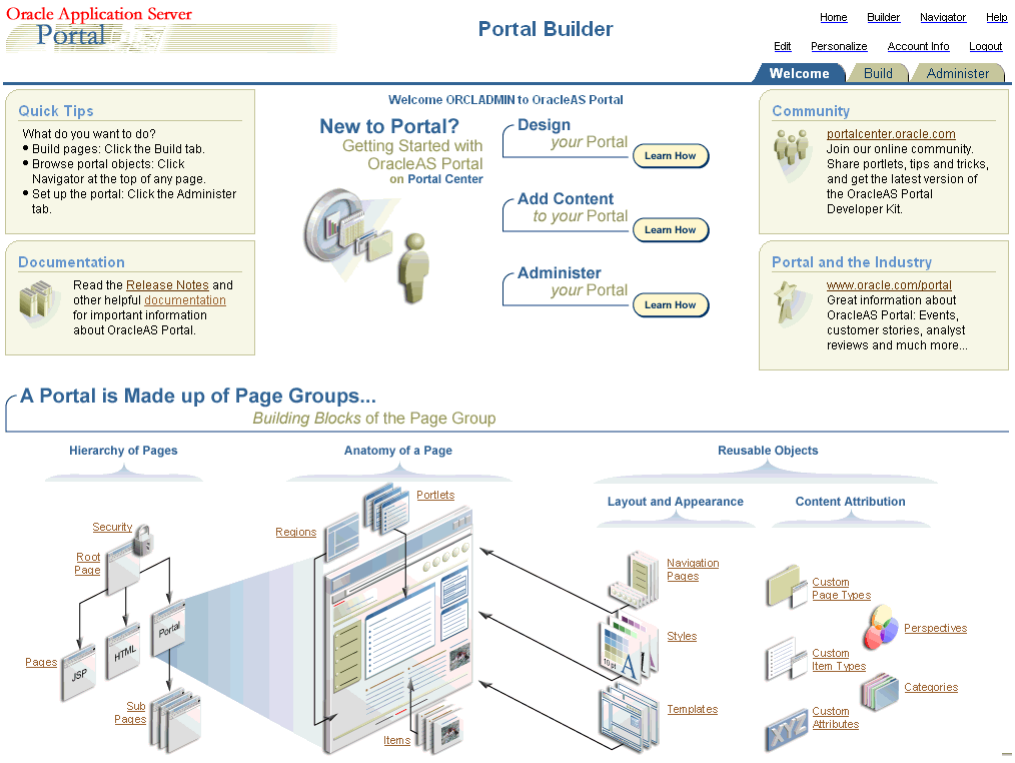
4.1.1 Using the OracleAS Portal Administer Tab

The OracleAS Portal framework provides administrative services, such as access to monitoring and configuration tools, single sign-on, directory integration, caching, and security. A lot of the features needed to manage users and groups, to set up security and search features, and to administer the portal and database are incorporated into a series of dialog boxes accessed through portlets on a portal page.

After you have installed OracleAS Portal, you need to log in as an administrator, to perform various administrative functions.

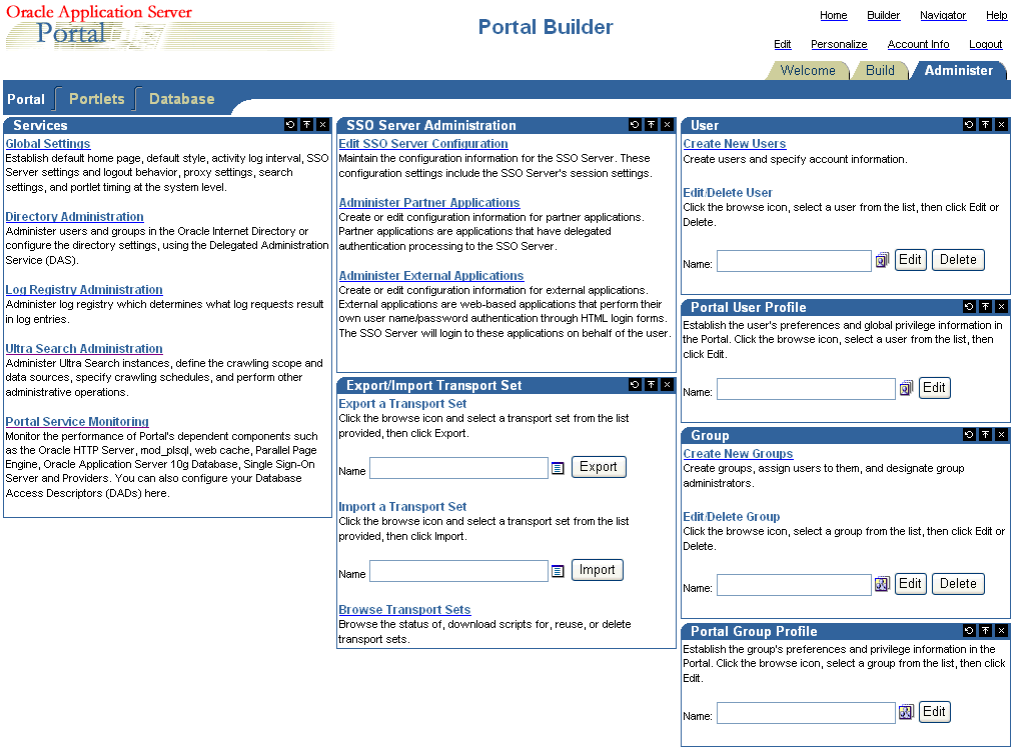
After you have logged in to OracleAS Portal, the **Portal Builder** page is displayed to you, as shown in Figure 4–1:

Figure 4–1 The Portal Builder Page



Click the **Administer** tab to view all the subtabs and portlets that help you administer the portal. The **Administer** tab is shown in Figure 4–2.

Figure 4–2 The Administer Tab on the Portal Builder Page



You will see the following subtabs in the **Administer** tab screen:

- Portal - This subtab enables you to create users and groups, administer the OracleAS Single Sign-On (SSO) server, and administer other services including Oracle Internet Directory, Oracle Ultra Search, Oracle Application Server Web Cache, proxy settings, and so on.
- Portlets - This subtab enables you to view the Portlet Repository and its refresh log, and register remote providers and provider groups.
- Database - This subtab enables you to create and edit database schemas, create and edit database roles, and monitor database information like database parameters, memory consumption, and database storage details.

Portal

This subtab under the **Administer** tab in the **Portal Builder** page contains the portlets shown in Table 4–1. This subtab is displayed by default when you click the **Administer** tab.

Table 4–1 Portlets in the Portal Subtab

Portlet Name	Enables You to
Services	■ Specify default home page, default style, and so on. ■ Administer users and groups in the Oracle Internet Directory or configure the directory settings. ■ Administer log registry. ■ Set up basic and advanced search features. ■ Specify Proxy Server settings. ■ Administer and monitor the performance of OracleAS Portal and its dependent components such as the Oracle HTTP Server, Portal Services, OracleAS Metadata Repository, Oracle Ultra Search, and providers using the Oracle Enterprise Manager 10*g* Application Server Control Console. See Chapter 7, "Monitoring and Administering OracleAS Portal" for more information on administering the log registry and monitoring OracleAS Portal performance.
SSO Server Administration	■ Edit OracleAS Single Sign-On (SSO) Server configuration. ■ Create or edit configuration information for partner applications. ■ Create or edit configuration information for external applications. See Chapter 6, "Securing OracleAS Portal" for more information. **Note:** You will need to log in as an Oracle Application Server administrator, such as **orcladmin**, to change SSO settings. The portal administrator (**portal**) does not have sufficient privileges to edit components other than OracleAS Portal.
Export/Import Transport Set	■ Export a transport set. ■ Import a transport set. ■ Browse the status of, download scripts for, reuse, or delete transport sets. See Chapter 10, "Exporting and Importing Content" for more information.
User	■ Create new users and specify account information. ■ Edit or delete users.
Portal User Profile	■ Establish the user's preferences and global privilege information in the portal.
Group	■ Create groups, assign users to them, and designate group administrators. ■ Edit or delete groups.
Portal Group Profile	■ Establish the group's preferences and privilege information in the portal.

Portlets

This subtab under the **Administer** tab in the **Portal Builder** page contains the portlets shown in Table 4–2.

Table 4–2 Portlets in the Portlets Subtab

Portlet Name	Enables You To
Portlet Repository	■ View all local and remote portlets. ■ Refresh information about all the portlets in the repository. ■ View Portlet Repository refresh log.
Remote Providers	■ Add a provider to the portlet repository. ■ Change configuration and access information about a provider.
Remote Provider Group	■ Register multiple providers with a single URL. ■ Edit a Provider Group registration.

Database

This subtab under the **Administer** tab in the **Portal Builder** page contains the portlets shown in Table 4–3.

Table 4–3 Portlets in the Database Subtab

Portlet Name	Enables You To
Schemas	■ Create new database schemas, or edit existing schemas.
Roles	■ Create new database roles, or edit existing roles.
Database Information	■ Monitor and view various database related information and parameters.
Database Memory Consumption, Transactions and Locks	■ Monitor database jobs. ■ View reports and charts of memory consumption and transactions. ■ Monitor session and locks. ■ Terminate undesirable user sessions.
Database Storage	■ Monitor and view various database storage related information.

4.1.2 Using Additional Administrative Tools

For some administrative tasks that cannot be performed through the OracleAS Portal **Administer** tab, you may need to use one of the following tools:

- Oracle Enterprise Manager 10g Application Server Control Console
- Portal Dependency Settings File and Tool
- Portal Installation and Configuration Scripts

4.1.2.1 Oracle Enterprise Manager 10*g* Application Server Control Console

The Oracle Enterprise Manager 10*g* Application Server Control Console is included when you install Oracle Application Server. From OracleAS Portal's perspective, consider this to be the administration console for the Oracle Application Server. The Application Server Control Console enables you to perform the following administration and configuration operations:

- Enable and disable components
- Administer clusters
- Start and stop services

- View logs and ports
- Perform real-time monitoring
- Modify the OracleAS Infrastructure services used by an Oracle Application Server middle tier.

Refer to Chapter 7, "Monitoring and Administering OracleAS Portal" for a detailed description of these Application Server Control Console functions.

4.1.2.2 Portal Dependency Settings File and Tool

OracleAS Portal is dependent on the components Oracle Application Server Web Cache and Oracle Internet Directory. It may be necessary to fine tune or configure these components after Oracle Application Server is installed.

To simplify configuration changes, OracleAS Portal introduces the *Portal Dependency Settings File*. This file stores configuration data from all the dependent components in a central place, and the content of the file is updated when there are configuration changes.

You can use the Portal Dependency Settings file to:

- Check settings used by an OracleAS Portal instance.
- Update settings in the Oracle Application Server Metadata Repository.

Refer to Appendix A, "Using the Portal Dependency Settings Tool and File" for a description of the Portal Dependency Settings file.

4.1.2.3 Portal Installation and Configuration Scripts

There are also various scripts, copied to your *ORACLE_HOME* during the installation of OracleAS Portal. These scripts may be needed to perform administrative actions. Refer to Appendix C, "Using OracleAS Portal Installation and Configuration Scripts" for a description of these scripts.

4.2 Finding Out Information About OracleAS Portal

This section covers the following topics:

- Accessing OracleAS Portal in Your Browser
- Finding Your OracleAS Portal Version Number

4.2.1 Accessing OracleAS Portal in Your Browser

After OracleAS Portal is installed, access it by entering the following URL in your browser:

`http://<host>:<port>/portal/pls/<dad>`

See Table 3–1, " Portal URL Descriptions" for an explanation of the URL components.

For backward compatibility, the old URL syntax is supported in this release. For example, `http://<host>:<port>/pls/<dad>`.

See Also: Section 3.1, "What Is Installed and Configured By Default?"

4.2.2 Finding Your OracleAS Portal Version Number

To find your portal version number:

1. Log in as a portal administrator.
2. In the Portal Builder, click the **Administer** tab.
3. Click the **Portal** subtab.
4. In the **Services** portlet, click the **Global Settings** link.

   The version number for your OracleAS Portal is shown at the bottom of the page.

4.3 Performing Basic Page Administration

This section covers the following topics:

- Setting a Default Home Page
- Setting the System Default Style
- Creating Personal Pages
- Setting the Total Space Allocated for Uploaded Files
- Setting the Maximum File Size for Uploaded Files
- Changing the Page Group Quota
- Specifying an Error Message Page
- Setting the Default Page for Non-Authenticated Users
- Removing the Context-Sensitive Help Link

4.3.1 Setting a Default Home Page

The home page is the first page that is displayed to a user after logging in to OracleAS Portal. Here's how the logic works:

- If the user has specified a personal home page, that page is displayed when the user logs on.
- If the user has not selected a personal home page, but the portal administrator has set one for him or her, the default home page specified for that user is displayed.
- If the user has not selected a personal home page, but belongs to a default group, the default home page specified for that group is displayed.
- If there is no default home page for the user's default group, or if the user has no default group, then the system default home page is displayed.

If mobile support is enabled, you can specify a default mobile home page to display when a user accesses the portal from a mobile device.

Note: You must be a portal administrator to define a default home page for the system, a group, or a user.

4.3.1.1 Setting the System Default Home Page

If there is no default home page for the user's default group, the system default home page is displayed.

To set the system default home page:

1. In the **Services** portlet, click **Global Settings**.

By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Next to the **Default Home Page** field, click the **Browse Pages** icon to see a list of pages from which to choose.

   Note: You cannot enter a value in this field; you must select one from the list.

3. Click **Return Object** next to the page you want to make the system default home page.
4. Click **OK**.

   Note: To check that you set the system default home page correctly, log out of the portal and log back in again. When you log back in, you should be taken the page that you specified as the system default home page.

4.3.1.2 Setting a Group's Default Home Page

If the user has not selected a personal home page, but belongs to a default group, the default home page specified for that group is displayed.

To set a group's default home page:

1. In the **Portal Group Profile** portlet, in the **Name** field, enter the name of the group for which you want to assign a default home page.

   By default, the **Portal Group Profile** portlet is on the **Administer** tab of the **Portal Builder** page.

   Note: If you are not sure of the group name, click the **Browse Groups** icon and select from the list provided.

2. Click **Edit**.
3. Next to the **Default Home Page** field, click the **Browse Pages** icon to see a list of pages from which to choose.

   Note: You cannot enter a value in this field; you must select one from the list.

4. Click **Return Object** next to the page you want to make the default home page for this group.
5. Click **OK**.

   Note: Click **Reset** to remove the group's default home page.

4.3.1.3 Setting a User's Default Home Page

If the user has not selected a personal home page, but you have set one for him or her, the default home page specified for that user is displayed.

To set a user's default home page:

1. In the **Portal User Profile** portlet, in the **Name** field, enter the user name of the user for whom you want to assign a default home page.

   By default, the **Portal User Profile** portlet is on the **Administer** tab of the **Portal Builder** page.

   Note: If you are not sure of the user name, click the **Browse Users** icon and select from the list provided.

2. Click **Edit**.
3. Next to the **Default Home Page** field, click the **Browse Pages** icon to see a list of pages from which to choose.

   Note: You cannot enter a value in this field; you must select one from the list.

4. Click **Return Object** next to the page you want to make the default home page for this user.
5. Click **OK**.

   Note: Click **Reset** to reset the user's default home page to the system default home page.

4.3.2 Setting the System Default Style

If you are the portal administrator, you are responsible for selecting a style to serve as the system default.

When a style is deleted, all pages and item regions that used the style revert to the page group default style. If the page group default style is **<None>**, all pages and regions revert to the system default style.

Note: To set the system default style, you must be the portal administrator.

To set the system default style:

1. In the **Portal Builder**, click the **Administer** tab.
2. Click the **Portal** subtab.
3. In the **Services** portlet, click the **Global Settings** link.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

4. In the **Default Style** section, choose a style from the **Display Name** list.

   Note: The list includes all public styles in the **Shared Objects** page group.

5. Click **OK** to return to the **Portal Builder**.

4.3.3 Creating Personal Pages

A personal page provides an area within OracleAS Portal where authorized users can store and share their own content. Personal pages are located under the **Shared Objects** page group, and are organized alphabetically by user name.

> **Note:** To create personal pages for users, you must be the portal administrator.

This section covers the following topics:

- Automatically Creating a Personal Page for New Users
- Creating a Personal Page for an Existing User

4.3.3.1 Automatically Creating a Personal Page for New Users

To configure OracleAS Portal to automatically create a personal page for new users:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.
2. Ensure that you are on the **Main** tab.
3. Select **Create Personal Pages for New Users**.
4. Click **OK**.

Whenever a new user logs on for the first time, a personal page is automatically created for that user.

> **Note:** Personal pages are automatically created when new users log on for the first time (that is, when the user record is created for the user), not for users that already exist.

4.3.3.2 Creating a Personal Page for an Existing User

To configure OracleAS Portal to create a personal page for an existing user:

1. In the **Portal User Profile** portlet:
   a. In the **Name** field, enter the name of the user for whom you want to create a personal page.

      > **Note:** If you are not sure of the name of the user, click the **Browse Users** icon and select from the list provided.

   b. Click **Edit**.

   By default, the **Portal User Profile** portlet is on the **Administer** tab of the **Portal Builder** page.
2. Ensure that you are on the **Preferences** tab.
3. Select **Create Personal Page**.

Note: If you do not see this check box, the user already has a personal page.

4. Click **OK**.

Notes:

- Personal pages are accessible from the Navigator in the **Shared Objects** page group. Any authorized user can drill down into the **Personal Pages** area of the **Shared Objects** page group, but they can only view their own personal page, or those personal pages to which they have been granted access.
- Personal pages for users with user names that do not begin with an alphabetic character are located under the **Others** area of **Personal Pages**.
- Personal pages cannot be deleted.

4.3.4 Setting the Total Space Allocated for Uploaded Files

You can limit the amount of space provided in your database to store documents uploaded to page groups. See Section 4.3.6, "Changing the Page Group Quota" if you want to limit the amount of space provided for a single page group.

You can also limit the size of individual files that content contributors can upload to page groups. See Section 4.3.5, "Setting the Maximum File Size for Uploaded Files" for more information.

When a user uploads a file to the portal, the upload is monitored in the middle tier to detect whether the total space or maximum file size is exceeded. If either of these limits is exceeded, the upload is terminated and an error message is displayed.

Note: To set the total space allocated for uploaded files, you must be the portal administrator.

To set the total space allocated for uploaded files:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Make sure you are on the **Main** tab.
3. In the **Total Space Allocated** radio group, select **Limit To** to limit the amount of space provided to store files uploaded to the page groups in this portal.
4. In the field, enter the maximum amount of megabytes provided for uploaded files across the whole portal. When this limit is reached, users will no longer be able to upload files to page groups in the portal.

Notes:

- Select **No Limit** if you do not want to impose a limit for uploaded files.
- The **Used Space** field displays the amount of space currently used by documents uploaded to page groups in this portal.

5. Click **OK**.

4.3.5 Setting the Maximum File Size for Uploaded Files

You can limit the size of individual files that users can upload to the page groups in your portal.

You can also limit the total amount of space provided in your database to store documents uploaded to page groups. See Section 4.3.4, "Setting the Total Space Allocated for Uploaded Files" for more information.

When a user uploads a file to the portal, the upload is monitored in the middle tier to detect if the maximum file size or portal file quota is exceeded. If either of these limits is exceeded, the upload is terminated and an error message is displayed.

Note: To set the maximum file size for uploaded files, you must be the portal administrator.

To set the maximum file size for uploaded files:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Make sure you are on the **Main** tab.
3. In the **Maximum File Size** radio group, select **Limit To** to specify the maximum size allowed for individual files uploaded to the portal.
4. In the field, enter the maximum size (in MB) for each individual file uploaded to the portal. If a content contributor attempts to upload a file larger than this size, an error is displayed.

   Note: Select **No Limit** if you do not want to impose a maximum file size.

5. Click **OK**.

4.3.6 Changing the Page Group Quota

You can limit the amount of space provided in your page group to store uploaded documents.

Note: To change the page group quota, you must have at least one of the following privileges:

- Portal administrator
- Manage all privileges on the page group
- Manage all global privileges on all page groups

To change the page group quota:

1. In the **Portal Navigator** page, click the **Page Groups** tab.
2. Click **Properties** next to the page group with which you want to work.
3. In the **Page Group Quota** section, select **Limit to** to limit the amount of space provided to store uploaded documents.
4. In the field provided, enter the size limit (in MB) for uploaded documents in the page group. When this limit is reached, users will no longer be able to upload documents to the page group.

   Note: Select **No limit** if you do not want to impose a limit for uploaded documents.

5. Click **OK**.

4.3.7 Specifying an Error Message Page

OracleAS Portal enables you to choose the error message page that you want to display to users. You can choose the default system error page, or you can specify your own customized error page.

OracleAS Portal includes an error message page (called Sample Error Page) that you can edit to match the look and feel of the other pages in your portal. The Sample Error Page is available under the **Portal Design-Time** page group and includes a portlet that displays all the diagnostic information. Alternatively, you can create your own error message page in any of your page groups. To do this, you must include the Error Message Portlet on the page and turn caching off.

Note: By default, the Error Message Portlet is located under the Administration Portlets page of the Portlet Repository.

To specify an error message page:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. In the **Error Page** section, select one of the following:

   - **System Error Page** to use the system error page to display full-page error messages to users. The system error page automatically includes all the diagnostic information.

- **Error Page** to use your own page to display full-page error messages to users. Click the **Browse Pages** icon to select the error message page that you want to use.

3. Click **OK**.

4.3.8 Setting the Default Page for Non-Authenticated Users

You can specify the default page that is displayed to users after they have logged out, or when they initially access the portal site, by setting the default home page for the PUBLIC (that is, non-authenticated) user.

> **Note:** You must be a portal administrator to define a default home page.

To set the default page users see when they log out, or when they initially access the site, perform the following steps:

1. In the **Portal User Profile** portlet, in the **Name** field, enter **PUBLIC**.

   By default, the **Portal User Profile** portlet is on the **Administer** tab of the **Portal Builder** page.

2. Click **Edit**.
3. Next to the **Default Home Page** field, click the **Browse Pages** icon to see a list of pages from which to choose.

   > **Note:** You cannot enter a value in this field; you must select one from the list.

4. Click **Return Object** next to the page you want to be displayed when users log out.
5. Click **OK**.

   > **Note:** Click **Reset** to remove this setting.

4.3.9 Removing the Context-Sensitive Help Link

If you have access to SQL*Plus, you can suppress the **Context-sensitive Help** link that appears in the banner in OracleAS Portal wizards, dialog boxes, alerts, and so on. Note that you cannot suppress the "**?**" icon that appears in the blue bar of wizards, dialog boxes, and alerts.

You cannot perform this task through the user interface; it must be done programmatically through SQL*Plus.

> **Note:** You must make the following API calls in both the portal schema and in the portal SSO schema.

To remove the context-sensitive help link:

1. Access SQL*Plus.

2. Enter:

```
exec wwui_api_body.set_display_help (wwui_api_body.DISPLAY_HELP_OFF);
commit;
```

To reinstate the context-sensitive help link:

1. Access SQL*Plus.
2. Enter:

```
exec wwui_api_body.set_display_help (wwui_api_body.DISPLAY_HELP_ON);
commit;
```

4.4 Configuring Self-Registration

To enable users to create their own portal user accounts, you must configure the self-registration feature. After completing this process, the self-registration link is exposed in the **Login** portlet.

You can set up an approval process for self-registered users so that they cannot log in until their accounts have been approved. When the account has been approved or rejected, the user is notified by e-mail.

If you do not require approval for self-registered users, the user will be able to log in to the portal immediately after registering.

Note: To set up self-registration, you must be the portal administrator.

To set up self-registration:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.
2. In the **Self-Registration Options** section, select **Enable Self-Registration**.
3. Select **No Approval Required** if self-registered users can log on to the portal immediately after registering.
4. Select **Approval Required** if self-registered users need to be approved before they can log on to the portal.
   a. Click **Configure** to set up the approval process.
   b. In the **Approvers** field, enter the names of the users or groups that must approve self-registered users.

      Note: Use a semicolon (;) as the separator between multiple users or groups. Each step of the approval routing can include both users and groups.

      ---

   c. For **Routing Method for Approvers**, choose:
      – **One at a time, all must approve** if you want each user or group to be notified in turn and every user or group must approve self-registered users before they can log on.

- **All at the same time, all must approve** if you want all the users and groups to be notified at the same time and every user or group must approve self-register users before they can log on.
- **All at the same time, only one must approve** if you want all the users and groups to be notified at the same time, but only one user or group member must approve self-registered users before they can log on.

d. Click **Add Step**.

e. Repeat steps a to d to add more steps to the approval process.

Notes:

- You do not need to change any other settings on this tab, or any of the settings on the other tabs in this screen.
- The final approver in the approval chain must be privileged to approve the registration requests of users. To do this, grant the **Allow account management** privilege to the final approver. In the case of Oracle Internet Directory or Oracle Delegated Administration Services (DAS) releases older than 10*g* Release 2 (10.1.2), grant **Allow user editing** privilege to the final approver. These privileges can be assigned on the default DAS **Edit User** page.
- Each approver in the approval chain must have the **My Notifications** portlet on their page to see and act upon new user accounts that are waiting for approval. The **My Notifications** portlet can be found under **Portal Content Tools** in the portlet repository.

---

f. Click **OK** to return to the Global Settings screen.

g. In the **E-Mail (SMTP) Host** section, enter the **Host Name** and **Port** of your e-mail server so that self-registered users can be informed by e-mail when their accounts are accepted or rejected.

5. Click **OK**.
6. Go to the home page of your portal.
7. Switch to Edit mode.
8. If the home page of your portal does not already contain a **Login** portlet, add the **Login** portlet to the page.

   By default, the **Login** portlet can be found in the **SSO/OID** page under the **Administration** page in the Portlet Repository.
9. Next to the **Login** portlet, click the **Edit Defaults** icon.
10. Select **Enable Self-Registration**.
11. In the **Self-Registration Link Text** field, enter the text that you want users to click to register with the portal.
12. Leave the **Self-Registration URL** field blank to use OracleAS Portal's own self-registration screen.

    If you have created your own self-registration screen, enter the URL in this field.
13. Click **OK**.

4.5 Performing Basic Portal Administration

This section covers the following topics:

4.5.1 Simplifying the Full URL of an OracleAS Portal Instance

You can simplify the full URL created by the OracleAS Portal installation to a more memorable or meaningful URL using the Redirect directive. In this way, end users can access OracleAS Portal by entering a simple URL.

By default, the URL for a new OracleAS Portal installation requires you to enter:

`http://<host>:<port>/portal/pls/<dad>`

You can simplify this URL to:

`http://<host>/<redirectpath>`

Note: Do not simplify the OracleAS Portal URL to `http://<host>:<port>/portal`. This is because OracleAS Portal is already mounted on `/portal`.

---

1. Open the Oracle HTTP Server configuration file, `httpd.conf`, which is located in the following directory:

   ORACLE_HOME`/Apache/Apache/conf/`

2. Enter the redirect path as follows:

   `Redirect /<redirectpath> http://<host>:<port>/portal/pls/<dad>`

   For example:

   ```
   Redirect /portalhome http://mysite.oracle.com/portal/pls/portal
   ```

   In this example, end users can enter:

   ```
   http://mysite.oracle.com/portalhome
   ```

   to access the full URL, which is:

   ```
   http://mysite.oracle.com/portal/pls/portal
   ```

Notes:

- The example `http://mysite.oracle.com/portalhome` assumes that the default port `80` is being used. If the default port is not being used, then the user would have to enter the URL with the port number, `http://mysite.oracle.com:<port>/portalhome`.
- You can also edit the `httpd.conf` file using the Oracle Enterprise Manager 10*g* Application Server Control Console.

  If the `httpd.conf` file is updated manually, you must synchronize the manual configuration changes done on the middle tier by running *ORACLE_HOME*`/dcm/bin/dcmctl` as follows:

  ```
  dcmctl updateConfig -ct ohs
  ```

  Finally, restart Oracle HTTP Server, by running the following command from *ORACLE_HOME*`/opmn/bin`:

  ```
  opmnctl restartproc type=ohs
  ```

4.5.2 Configuring Oracle HTTP Server to Use the OracleAS Portal Home Page

To set the OracleAS Portal home page as the Oracle HTTP Server's default home page:

1. In the directory *ORACLE_HOME*`/Apache/Apache/htdocs/`, make a backup copy of the files `index.html` and `index.html.<lang>`, where `<lang>` is the language code. For example, `index.html.en` is the index HTML file for English.
2. Edit `index.html.<lang>` by replacing the entire contents of the file with the following HTML redirection code:

```
<HTML>
<SCRIPT LANGUAGE=JavaScript>
document.location="http://<host>.<domain>:<port>/portal/pls/<dad>"
</SCRIPT>
</HTML>
```

Notes:

- Do not specify a port number if you are running OracleAS Portal on port `80`.
- If you plan to support other languages, you need to have the language-specific index HTML files with the redirection code, for these languages.

4.5.3 Configuring a Portal DAD

A Database Access Descriptor (DAD) is a set of values that specify how an application connects to an Oracle Database to fulfill an HTTP request. The information in the DAD includes the user name (which also specifies the schema and the privileges), password, connect-string, and Globalization Support language of the database.

There are two types of DADs: general DADs and portal DADs. An OracleAS Portal middle tier uses a *portal DAD* to access the OracleAS Metadata Repository, and this section describes how you can configure portal DAD information. For information on general DADs, refer to the *Oracle HTTP Server Administrator's Guide*.

You can configure the portal DAD for a particular OracleAS Portal instance from the Oracle Enterprise Manager 10*g* Application Server Control Console:

1. Navigate to the Application Server Control Console.

   See Section 7.2.1, "Accessing the Application Server Control Console" for more information.

2. Navigate to the home page for your OracleAS Portal instance.

   See Section 7.3, "Using Application Server Control Console to Monitor and Administer OracleAS Portal" for more information.

3. Click **Portal DAD Settings**.

4. Edit the DAD for this OracleAS Portal instance as required. Table 7–3, " DAD Settings" has a description of all the options on this page.

5. Click **Apply**.

6. Restart Oracle HTTP Server and OC4J_Portal.

   Navigate to the Oracle Application Server home page. In the **System Components** table, select **HTTP_Server** and **OC4J_Portal**, and then click the **Restart** button.

4.5.4 Configuring the Portal Cache

Portal cache is a file system-based cache for OracleAS Portal pages and portlets. See Section 1.3.2, "Understanding Portal Cache" for more information.

You can configure the Portal cache in the Oracle Enterprise Manager 10*g* Application Server Control Console:

1. Navigate to the Application Server Control Console.

   See Section 7.2.1, "Accessing the Application Server Control Console" for more information.

2. Navigate to the home page for your OracleAS Portal instance.

   See Section 7.3, "Using Application Server Control Console to Monitor and Administer OracleAS Portal" for more information.

3. Click **Portal Cache Settings**.

4. Ensure that the **Caching** option is set to **On**.

5. Edit the cache settings for the OracleAS Portal instance as required. Table 7–2, " Portal Cache Settings" has a description of all the options on this page.

6. Click **Apply**.

7. Restart Oracle HTTP Server and OC4J_Portal.

   Navigate to the Oracle Application Server home page. In the **System Components** table, select **HTTP_Server** and **OC4J_Portal**, and then click the **Restart** button.

4.5.5 Clearing the Portal Cache

Sometimes you must clear the entire portal cache (the OracleAS Portal file system-based cache). For example, when you change the character set of the OracleAS Metadata Repository, you will need to clear the entire portal cache as the existing content will use the older character set.

To clear the portal cache:

1. Navigate to the portal cache directory. The default path is *ORACLE_HOME*/Apache/modplsql/cache.
2. Perform a recursive delete of all the files under this directory. For example, on UNIX platforms, issue the following command:

```
rm -rf *
```

Notes:

- Whenever you clear the portal cache, you may need to clear the OracleAS Web Cache content as well. Refer to Section 5.8.3, "Managing Portal Content Cached in OracleAS Web Cache" for information about clearing the OracleAS Web Cache content.
- You must clear the portal cache on all middle tiers.

WARNING: Ensure that you are in the correct directory before issuing this command. Do not delete the `cache` directory.

4.5.6 Using a Custom Image Directory

To avoid losing custom images stored in the OracleAS Portal images directory (which is *ORACLE_HOME*/portal/images by default) during a future upgrade, it is recommended that you create your own images directory and set up an appropriate Oracle HTTP Server alias for this directory.

For example, add an entry, similar to the one shown next, to the file *ORACLE_HOME*/portal/conf/portal.conf. It is recommended that you use the local Oracle Enterprise Manager 10*g* Application Server Control Console instance to make this change. For more information, refer to the *Oracle HTTP Server Administrator's Guide* or the *Oracle Application Server Web Cache Administrator's Guide*.

```
Alias /mycompany/images/ "/opt/app/myportal/images/"
<Directory "/opt/app/myportal/images/">
     AllowOverride None
     Order allow,deny
     Allow from all
     ExpiresActive on
     ExpiresDefault A2592000
  <Files *>
     Header set Surrogate-Control 'max-age=2592000'
  </Files>
</Directory>
```

You do not need to perform any specific OracleAS Web Cache configuration as OracleAS Web Cache is already configured to globally cache .bmp, .gif, .png, .jpg, and .jpeg files.

4.6 Configuring Mobile Support in OracleAS Portal

This section discusses how OracleAS Portal and Oracle Application Server Wireless are configured to operate together. OracleAS Portal pages can be viewed from a wide variety of devices including desktop browsers, mobile phones, and PDAs. OracleAS Portal uses OracleAS Wireless to provide wireless functionality to receive requests

from wireless devices, and transform content provided by the portal into an appropriate format.

This section describes the following:

- What Is Installed By Default?
- Configuring Mobile Settings in OracleAS Portal
- Manually Reconfiguring the Mobile Setup
- Changing the Mobile Device Component of the Cache Key

4.6.1 What Is Installed By Default?

Performing a standard Oracle Application Server installation of OracleAS Portal and Oracle Application Server Wireless configures mobile support in OracleAS Portal as follows:

- A service is created that provides mobile device access to the installed portal and this service refers to the portal home page URL. As mobile access to the portal is mediated by OracleAS Wireless, a mobile device must communicate with OracleAS Wireless to access content on OracleAS Portal.
- The OracleAS Wireless service's URL refers to OracleAS Portal. Users that access this OracleAS Wireless service are directed to the public home page of the portal. When a mobile browser contacts OracleAS Portal through the home page URL, the request is redirected to the OracleAS Wireless service.

4.6.2 Configuring Mobile Settings in OracleAS Portal

To configure mobile settings in OracleAS Portal:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Mobile** tab.

Most mobile-related settings for OracleAS Portal are found here. For more detail, see:

- Enabling Mobile Access
- Configuring Mobile Home Pages
- Displaying Page Titles in Mobile Banner Links
- Displaying Enhanced Page Layouts on PDAs
- Logging Mobile Responses
- Updating the OracleAS Wireless Portal Service URL Reference

---

Note: In a hosted environment, you can control each subscriber individually. The exception to this is the *OracleAS Wireless Service URL* setting. When OracleAS Portal is operating in hosted mode (with multiple subscribers), any change to the OracleAS Wireless Service URL must be made by the hosting administrator, using a command line script, as it affects all subscribers.

4.6.2.1 Enabling Mobile Access

The **Enable Mobile Access** option enables you to control how OracleAS Portal responds when a mobile client requests portal pages through OracleAS Wireless. If you want OracleAS Portal to return pages and portlets in response to mobile requests, you must select the **Enable Mobile Access** option.

If you do not select this option, OracleAS Portal responds to mobile requests with a message stating that it is not mobile enabled.

To enable mobile access:

1. In the **Services** portlet, click **Global Settings**.
2. Click the **Mobile** tab.
3. Select the **Enable Mobile Access** option.
4. Click **OK**.

4.6.2.2 Configuring Mobile Home Pages

Your mobile home page is the first page you see when you access OracleAS Portal from a mobile device. If mobile access is enabled, you may choose whether users may select a home page specifically for mobile devices and you can also determine whether all mobile home pages display a Login Link by default:

- Enabling Users to Select Mobile Home Pages
- Excluding Login Links from Mobile Home Pages

4.6.2.2.1 Enabling Users to Select Mobile Home Pages

The **Enable Mobile Home Page Selection** option enables you to control whether portal users may select separate home pages for mobile access. If you do not select this option, the home pages displayed on mobile devices is the same home page that is used for standard desktop access.

To allow mobile home pages:

1. In the **Services** portlet, click **Global Settings**.
2. Click the **Mobile** tab.
3. Select the **Enable Mobile Home Page Selection** option.
4. Click **OK**.

When you select this option, the **Default Mobile Home Page** field become available to users on the Account Info page. For more information, see the *Oracle Application Server Portal User's Guide*.

4.6.2.2.2 Excluding Login Links from Mobile Home Pages

The **Exclude Login Link from Mobile Home Page** option enables you to control whether a Login Link is displayed on mobile home pages. If mobile home pages are allowed, a Login Link is displayed on the mobile home page by default. Select this option if you do not want the default Login Link to be displayed.

To exclude Login Links from mobile home pages:

1. In the **Services** portlet, click **Global Settings**.
2. Click the **Mobile** tab.
3. Select the **Exclude Login Link from Mobile Home Page** option.

4. Click **OK**.

4.6.2.3 Displaying Page Titles in Mobile Banner Links

The **Use Page Titles in Mobile Banner Links** option enables you to choose what text is displayed in the navigation links that appear in the mobile banner. Select this option to use the titles of pages in navigation link text. To see an example, refer to Figure 4–3. If you do not select this option, the default text (*Home* and *Back*) is displayed instead.

To use page titles in navigation link text:

1. In the **Services** portlet, click **Global Settings**.
2. Click the **Mobile** tab.
3. Select the **Use Page Titles in Mobile Banner Links** option.
4. Click **OK**.

4.6.2.4 Displaying Enhanced Page Layouts on PDAs

The **Enhance Display for PDAs** option allows enhanced page layouts to be displayed on PDAs (Personal Digital Assistants). PDAs have better display capabilities than other, more simple mobile devices; therefore it is possible to enhance portal page display for PDAs.

If you select this option, default font and color settings on the PDA are used for the text, link text, the page list background, the banner background, and so on. By setting additional **PDA Display Options** you can override the default PDA display settings and include an image in the PDA page banner if you wish. See Figure 4–3, "Sample PDA Page Layout".

Figure 4–3 Sample PDA Page Layout

If you do not select this option, the same page layout is used for all mobile devices.

To display enhanced page layouts on PDAs and (optionally) customize PDA display options:

1. In the **Services** portlet, click **Global Settings**.
2. Click the **Mobile** tab.
3. Select the **Enhance Display for PDAs** option.
4. Click **Apply**.

   When you click Apply, a new section called **PDA Display Options** is displayed at the bottom of the page.
5. (Optional) Set **PDA Display Options** to control how portal pages are displayed on PDAs. Ensure that you use valid markup when specifying your font and color preferences.

   For more detail, see Table 4–4, " PDA Display Options".
6. Click **OK**.

Table 4–4 PDA Display Options

Option	Description
General Options	Override the default font and color for: ■ **Background Color** - Specify a background color for portal pages; for example, enter `#FF0000` or `red`. ■ **Font Name** - Specify the font used to display text on portal pages; for example, enter `arial`. ■ **Font Size** - Specify the font size used to display text on portal pages; for example, enter `-1`. ■ **Font Color** - Specify the font color used to display text on portal pages; for example, enter `#0000FF` or `blue`. To use the default font or color selected by the PDA, leave the appropriate field blank.
Basic Link Options	Override the default colors: ■ **Unvisited Link Color** - Specify a color for unvisited links on portal pages; for example, enter `#00FFFF` or `lightblue`. ■ **Selected Link Color** - Specify a color for selected links on portal pages; for example, enter `#FFFFFF` or `white`. ■ **Visited Link Color** - Specify a color for visited links on portal pages; for example, enter `#FF00FF` or `magenta`. To use the default link color selected by the PDA, leave the appropriate field blank.

Table 4–4 (Cont.) PDA Display Options

Option	Description
Banner Image Options	Use these options to specify an image (.GIF) for the PDA banner: ■ **Banner Image (File name or URL)** - If the image is located in the portal's default image directory, enter the name of the .GIF file only; for example, enter `mylogo`. Alternatively, enter the full URL to the image; for example, enter `http://www.mycompany/images/mylogo`. ■ **In Default Image Directory?** - Select this check box if the banner image you want to use is located in the portal's default image directory. Clear this check box if the image is accessible from a URL. ■ **Banner Background Color** - Specify a background banner color for portal pages; for example, enter `#00FFFF` or `lightblue`. Leave the field blank to use the default color selected by the PDA.
Page List (Breadcrumbs) Options	Override the default colors: ■ **Foreground Color** - Specify a foreground color for portal page breadcrumbs; for example, enter `#00FFFF` or `lightblue`. ■ **Background Color** - Specify a background color for portal page breadcrumbs; for example, enter `#0000FF` or `blue`. ■ **Link Color** - Specify a color for link text in portal page breadcrumbs; for example, enter `#000000` or `black`. To use the default color selected by the PDA, leave the appropriate field blank.
Login / Logout Link	Specify a color for the Login/Logout link displayed on portal pages; for example, enter `#000000` or `black`.

4.6.2.5 Logging Mobile Responses

The **Log Mobile Responses** option enables you to control whether portlet responses to mobile requests are logged. This feature is useful during development for portlet debugging purposes. When you select this option, the portal logs the content that mobile portlets generate when displayed on a page in response to a mobile device request.

For mobile devices, portal content is rendered in an Oracle specific markup language called MobileXML. This markup is transformed by OracleAS Wireless to the appropriate device markup that generated the request.

Portlet responses are logged when all the following conditions are met:

- The **Log Mobile Responses** option is selected.
- The user making the request is logged on.
- The request is either from a mobile device, or it is for a mobile page.

Notes:

- This option is intended for development purposes only. We do not recommend that you set this option in a production portal as mobile response logging will impact your portal performance.
- Whenever you enable or disable the **Log Mobile Responses** option, all currently cached page data is invalidated. Therefore, we recommended that you do not change this option frequently after your OracleAS Portal has been deployed for general access.

To log portlet responses to mobile requests:

1. In the **Services** portlet, click **Global Settings**.
2. Click the **Mobile** tab.
3. Select the **Log Mobile Responses** option.
4. Click **OK**.

OracleAS Portal comes with two built-in portlets for viewing the content that is logged:

- **Most recent mobile log entry** - Shows only the most recent record for a particular user, irrespective of the portlet from which the data was recorded.
- **Mobile log portlet** - Shows a list of all the portlets for which a user has content recorded, the user can select which portlet's content they wish to review

You will find additional information in the article *Provider Debugging Techniques: Using the Mobile Log Viewers*, on the Oracle Technology Network (OTN), http://www.oracle.com/technology/products/ias/portal/html/mobile_10g_debugging.with.logs.html.

4.6.3 Manually Reconfiguring the Mobile Setup

An Oracle Application Server reconfiguration that results in a change to the Oracle Application Server Wireless service URL or OracleAS Portal home page URL, requires the changes to be reflected in the stored information in OracleAS Portal, and the OracleAS Wireless service definition that refers to OracleAS Portal. You should reconfigure OracleAS Wireless and OracleAS Portal to ensure that the communication between them is not affected.

You have to manually reconfigure OracleAS Wireless and OracleAS Portal to update the values of the following referenced URLs. For more information, see:

- Updating the OracleAS Portal Home Page URL References
- Updating the OracleAS Wireless Portal Service URL Reference

4.6.3.1 Updating the OracleAS Portal Home Page URL References

The OracleAS Portal home page URL is the address that OracleAS Wireless service definition refers to. If the home page URL changes, you need to update the following references to it:

- The Oracle Application Server Wireless Service Definition
- OracleAS Portal's Internal Reference to Itself

4.6.3.1.1 The Oracle Application Server Wireless Service Definition

To update the OracleAS Portal home page URL in the OracleAS Wireless server service definition:

1. Log in to OracleAS Wireless Tools by using the following URL:

   `http://<host>:<port>/webtool/login.uix`

2. Enter your user name and password.
3. Click the **Contents** tab.
4. In the Content Manager, select the portal service, and click **Edit**.
5. Click **Input Parameters** on the left side of the screen.
6. In the Input Parameters screen, change the URL as required.
7. Click **Apply** to save the changes.
8. Log out of the OracleAS Wireless Content Manager.

See Also: *Oracle Application Server Wireless Administrator's Guide*

4.6.3.1.2 OracleAS Portal's Internal Reference to Itself

To change OracleAS Portal's own reference to its home page URL, use the script `cfgiasw.pl` to manually update the value. The script files are located here:

ORACLE_HOME`/assistants/opca/`

To run the script, use the following command:

```
perl cfgiasw.pl -s portal -c portal_db -h "http://my_portal_server.com/portal/pls/portal/portal.home"
```

The preceding example is specific to a UNIX machine. See Section C.8, "Using the cfgiasw Script to Configure Mobile Settings" for more information on the `cfgiasw` script.

4.6.3.2 Updating the OracleAS Wireless Portal Service URL Reference

Oracle Application Server Wireless is used by OracleAS Portal as an intermediary in providing access to mobile devices. To provide this access, OracleAS Portal must know the URL to the OracleAS Wireless service on which the portal is registered. If the OracleAS Wireless service URL changes, its reference within OracleAS Portal must be updated. This reference can be updated in either of the following ways:

- Using the Global Settings Page to Update the OracleAS Wireless Portal Service URL
- Using the cfgiasw Script to Update the OracleAS Wireless Service URL Reference

4.6.3.2.1 Using the Global Settings Page to Update the OracleAS Wireless Portal Service URL

To update the OracleAS Wireless Portal Service URL using the Global Settings page:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Mobile** tab.
3. Enter the URL in the **OracleAS 10*g* Wireless Portal Service URL** field.

   The **Portal home page URL** is displayed for information only.

4. Click **OK**.

You can change the **OracleAS 10*g* Wireless Portal Service URL** setting only when OracleAS Portal is not operating with multiple subscribers. If OracleAS Portal is operating with multiple subscribers, only the hosting administrator should change the value of **OracleAS 10*g* Wireless Portal Service URL**.

4.6.3.2.2 Using the cfgiasw Script to Update the OracleAS Wireless Service URL Reference If you need to change OracleAS Portal's reference to the URL of Oracle Application Server Wireless Portal service, you can use the script `cfgiasw.pl` to manually set the value. The script files are located here:

ORACLE_HOME`/assistants/opca/`

To run the script, use the following command:

```
perl cfgiasw.pl -s portal -c portal_db -w "http://my_iasw_server.com/ptg/rm?PAoid=12345"
```

The preceding example is specific to a UNIX machine. See Section C.8, "Using the cfgiasw Script to Configure Mobile Settings" for more information on the `cfgiasw` script.

4.6.4 Changing the Mobile Device Component of the Cache Key

OracleAS Wireless is integrated with OracleAS Web Cache to improve page rendering performance and scalability. The cache is used as a repository for post-transformed content; the wireless runtime determines what content needs to be inserted into the cache and when to expire content in the cache. The cache key used by OracleAS Portal is composed of numerous components. One of these components is based on the OracleAS Wireless header, X-Oracle-Device.Class. This component allows portlet content to be cached based on the class of the mobile device used.

You can enable portlet content to be cached based on the name of a specific device rather than the device class. Refer to Section C.9, "Using the cfgxodnc.pl Script to Change the Mobile Device Component of the Cache Key" for more information.

4.7 Managing Users, Groups, and Passwords

Refer to Chapter 6, "Securing OracleAS Portal" for more information on managing users, groups, and passwords.

4.8 Configuring Browser Settings

Refer to **Browser Recommendations** in the Preface of the *Oracle Application Server Portal User's Guide*.

4.9 Configuring Language Support

OracleAS Portal allows application development and deployment in different languages. This allows developers to work in their own language when they build portals. In addition, the self-service content management supports multiple languages so that end users can provide documents and other content in different languages.

OracleAS Portal is configured with the languages that are selected in the Oracle Universal Installer (OUI) during the Oracle Application Server middle-tier installation. Languages that are configured show up in the **Set Language** portlet. You can use OracleAS Portal in the language that corresponds to the language setting in the browser, or to the language you have selected in the **Set Language** portlet. However, the language setting in the browser must correspond to an installed language in

OracleAS Portal. The **Set Language** portlet is not displayed by default, but you can add the portlet to the **Portal Builder** page or any other page that you create in OracleAS Portal. To configure additional languages after installation, you must use the `ptllang` tool.

> **See Also:** The section **Working with the Set Language Portlet** in the *Oracle Application Server Portal User's Guide*.

Table 4–5 shows the languages that are available for OracleAS Portal.

Table 4–5 OracleAS Portal Languages and Language Abbreviations

Language	Language Abbreviation
Arabic	ar
Brazilian-Portuguese	ptb
Canadian French	frc
Czech	cs
Danish	dk
Dutch	nl
English	us
Finnish	sf
French	f
German	d
Greek	el
Hebrew	iw
Hungarian	hu
Italian	i
Japanese	ja
Korean	ko
Latin American Spanish	esa
Norwegian	n
Polish	pl
Portuguese	pt
Romanian	ro
Russian	ru
Simplified Chinese	zhs
Slovak	sk
Spanish	e
Swedish	s
Thai	th
Traditional Chinese	zht
Turkish	tr

> **Note:** OracleAS Portal is not supported on the `ZHT32EUC` database character set. If your environment supports Traditional Chinese, then use the `AL32UTF8`, `ZHT16MSWIN950`, or `ZHT16BIG5` character set. For more information about selecting the character set in the Oracle Universal Installer, refer to the *Oracle Application Server Globalization Guide*.

This section describes the following:

- Installing Languages After Installation
- Enabling the Use of Territories

4.9.1 Installing Languages After Installation

To install languages after you have installed OracleAS Portal, run `ptllang`. You must run `ptllang` for each language that you want OracleAS Portal to support.

> **Caution:** During login operations, information is sent to Oracle Application Server Single Sign-On. The language used in the authentication request is sent back to OracleAS Portal. OracleAS Single Sign-On must have all languages installed that exist on the OracleAS Portal, so that the selected language is recognized. If OracleAS Single Sign-On does not have the selected language installed, it will default to **US English**. This is the language that would be asserted to any OracleAS Portal that requested authentication in a language that is not available on OracleAS Single Sign-On.
>
> The **Set Language** portlet in OracleAS Portal sets a language and a Persistent Language cookie on OracleAS Single Sign-On and OracleAS Portal.
>
> If there are multiple portals configured to use the same OracleAS Single Sign-On, and the portals have different languages installed, all the combined languages must exist on the OracleAS Single Sign-On to accommodate a Set Language request from any of the portals.

Environment

The language installation must be run from the Oracle Application Server Portal Upgrade CD–ROM.

- Mount the Oracle Application Server Portal Upgrade CD–ROM.
- Set the ORACLE_HOME environment variable to the Oracle home that contains the OracleAS Portal schema.

`ptllang` must be run from the Oracle Application Server Portal Upgrade CD–ROM in which OracleAS Portal is installed. `ptllang` is located in the *`CD_ROOT`*`/assistants/opca` directory.

Assumptions

The OracleAS Metadata Repository is already installed, and the respective databases are up.

Usage

On Windows:

```
ptllang.bat  -lang lang_code [ -s portal_schema] [-sp portal_schema_password] [-c
portal_db_connect_string] [-log log_file_directory]
```

On UNIX:

```
ptllang.sh  -lang lang_code [ -s portal_schema] [-sp portal_schema_password] [-c
portal_db_connect_string] [-log log_file_directory]
```

Table 4–6 lists and describes the parameters supported by `ptllang`.

Table 4–6 ptllang Parameters

Parameter	Definition
`-s`	OracleAS Portal schema name. Default: `portal`
`-sp`	OracleAS Portal schema password.
`-c`	Connect string to the target database where OracleAS Metadata Repository is installed. The format must be `DbHostName:DbPortNumber:DbServiceName`.
`-lang`	Abbreviation for the language to install. Refer to Table 4–5, " OracleAS Portal Languages and Language Abbreviations" for a list of all the supported abbreviations.
`-log`	The directory that the log file is written to.

Usage example

The following examples pass in the input provided on the command line. The examples load the Dutch language strings into the portal schema in the OracleAS Metadata Repository.

On Windows:

```
ptllang.bat -s portal -sp portal -c  myDBhost.domain.com:1521:dbServiceName  -lang
nl -log c:\temp
```

On UNIX:

```
ptllang.sh -s portal -sp portal -c  myDBhost.domain.com:1521:dbServiceName  -lang
nl  -log /oracle/log
```

See Also: *Oracle Application Server Globalization Guide*

4.9.2 Enabling the Use of Territories

Once a language is installed into OracleAS Portal, the end user can select the language to be used from the languages displayed in the **Set Language** portlet. For a given language, portal users may also select their geographic location (territory) so that localization settings such as date, currency, and decimal formats are displayed correctly. For example, if the portal language is set to English, portal users may select from territories such as, America, Australia, Canada, Ireland, United Kingdom, and so on.

Territory selection is not available on the **Set Language** portlet by default. If you want portal users to be able to specify their geographical location (territory), you must edit the **Set Language** portlet.

The **Set Language** portlet is not displayed by default. However, you can add it to the **Portal Builder** page or any other OracleAS Portal page.

Adding the Set Language Portlet to a Portal Page

To add the **Set Language** portlet to a portal page:

1. Log in to OracleAS Portal as the portal schema owner.
2. Display the page where you want to display the **Set Language** portlet. For example, you might want to add the **Set Language** portlet to the **Administrator** tab on the Portal Builder Page.
3. Click **Edit** on top of the page.
4. Click the **Add Portlet** icon in the region where you want to add the portlet.
5. In the Portlet Repository, click **Portal Content Tools**.
6. Click **Set Language** in the **Available Portlets** area, and click **OK**.

   The **Set Language** portlet is now available on the portal page.

> **Note:** If you add the **Set Language** portlet to a page and subsequently install another language, the new language is not displayed when you view the page. As a workaround, remove the portlet and add it to the page again.

Enabling the Use of Territories and Locales

To enable the use of territories and locales:

1. Log in to OracleAS Portal as the portal schema owner.
2. Click the **Edit Defaults** icon for the **Set Language** portlet.
3. In the **Edit Set Language Portlet Settings** screen shown, select the **Enable Territory Selection** option.
4. Click **OK**.

By selecting the **Enable Territory Selection** option, the appropriate locales for each registered language are displayed. The locales are listed after the languages in the Set Language portlet, as shown in Figure 4–4.

Figure 4–4 The Set Language Portlet

Note: The OracleAS Portal online Help system, which uses *Oracle Help for the Web*, relies on certain fonts to render the online Help user interface in different languages. To get the correct fonts installed, you must select all the languages in which you want to render the online Help, at the time of installation of the middle-tier server. To do this, click the **Product Languages** button, and select your languages on the **Select a Product to Install** screen, during the installation.

Additionally, you must make sure that the languages that are installed on the Application Server middle tier correspond with the languages that are installed on the Oracle Application Server Infrastructure, to avoid problems with the Set Language request issued to OracleAS Single Sign-On.

Installing all languages increases the time required for the middle-tier installation.

4.10 Configuring OracleAS Portal for WebDAV

WebDAV is a protocol extension to HTTP 1.1 that supports distributed authoring and versioning. With WebDAV, the Internet becomes a transparent read and write medium, where content can be checked out, edited, and checked in to a URL address. mod_dav is an implementation of the WebDAV specification. The standard mod_dav implementation supports read and write access to files.

The term OraDAV refers to the capabilities available through the mod_oradav module. mod_oradav is the Oracle module that is an extended implementation of mod_dav, and is integrated with the Oracle HTTP Server. mod_oradav can read and write to local files, but also to an Oracle Database. The Oracle Database must have an OraDAV driver installed. The OraDAV driver is installed by default on installation of OracleAS Portal. mod_oradav calls this driver to map WebDAV activity to database activity. mod_oradav enables WebDAV clients to connect to an Oracle Database, read and write content, and query and lock documents in various schemas.

See Also: *Oracle HTTP Server Administrator's Guide*

When Oracle Application Server is installed, all required OraDAV parameters are set with values that enable access to Oracle Database content through a Web browser or a WebDAV client. If necessary, you can modify parameter values if the default values do not meet your needs.

Similar to the portal DAD configuration file, WebDAV has it own configuration file (*ORACLE_HOME*`/Apache/oradav/conf/oradav.conf`) that contains the OraDAV parameters and start with *DAV* and *DAVParam*. These parameters are specified within a <*Location*> directive. The `oradav.conf` file is included in the `httpd.conf` file in an include statement.

See Also: *Oracle Application Server Portal User's Guide*

4.10.1 Performing Basic WebDAV Configuration

After OracleAS Portal has been installed as part of the Oracle Application Server installation, the `oradav.conf` file should be populated with a <*Location*> directive that points to the portal schema. In Example 4–1, the location `/dav_portal/portal` will be OraDAV-enabled and will (once populated with the correct values) connect to the portal schema so that users can use WebDAV clients to access portal data.

Example 4–1 Configuration Parameters for Portal Access

```
<Location /dav_portal/portal>
   DAV Oracle
   DAVParam ORACONNECT dbhost:dbport:dbsid
   DAVParam ORAUSER portal_schema
   DAVParam ORAPASSWORD portal_schema_password
   DAVParam ORAPACKAGENAME portal_schema.wwdav_api_driver
</Location>
```

By default, the OracleAS Portal DAV URL is:

```
http://<host>:<port>/dav_portal/portal/
```

For example:

```
http://mysite.oracle.com:7777/dav_portal/portal
```

The `dav_portal` part of the URL is the default name of a virtual directory used to differentiate between portal access through a WebDAV client and portal access that uses the `pls` virtual directory. `portal` is the DAD of the portal installation. You can also configure virtual hosts to provide a different, simpler, or easier to remember URL for WebDAV access, if need be.

Users connect to a portal in WebDAV clients using the same user name and password that they use to log in to the portal itself. If the portal is in a hosted environment, users also need to add their company information to their user name, as follows:

```
<username>@<company>
```

Authentication

Due to the way some WebDAV clients behave, users might experience authentication requests multiple times. To avoid this, the portal administrator can enable the cookie option by adding the following line to the `oradav.conf` file:

```
DAVParam ORACookieMaxAge <seconds>
```

where `seconds` is the amount of time in seconds before the cookie expires.

For example a value of 28800 is 8 hours and means that once a user has logged on through a WebDAV client, he or she will not be prompted for a user name and password again until 8 hours has passed.

> **Note:** Some WebDAV clients, for example, Dreamweaver, do not support cookies, so even if the cookie option is enabled, users may still be prompted for their passwords multiple times.

If you are using the SQL*Net Advanced Security Option (ASO), the `ORACONNECT` parameter in the `oradav.conf` file must be replaced with `ORASERVICE dbhost` as shown next:

```
<Location /dav_portal/portal>
  DAV Oracle
  DAVParam ORASERVICE dbhost
  DAVParam ORAUSER portal_schema
  DAVParam ORAPASSWORD portal_schema_password
  DAVParam ORAPACKAGENAME portal_schema.wwdav_api_driver
  Options Indexes
</Location>
```

This allows the database alias to be resolved by the `tnsnames.ora` file.

Notes:

- When you add a new DAD without specifying the user name and password, or if you change the portal database schema user name or password using SQL*Plus, you will need to update the `dads.conf` and `oradav.conf` files manually.
- Whenever you make changes to `dads.conf` or `oradav.conf`, Oracle HTTP Server and OC4J_Portal must be restarted before the new settings will take effect.

Default Time Limit for File Locks

The new `DEFAULTLOCKTIMEOUT` parameter provides information about the amount of time for which any single lock created by a DAV client will endure if the client does not actively maintain the lock. This is an optional parameter. By modifying this value, you can define the default amount of time beyond which the locks will expire.

The `DEFAULTLOCKTIMEOUT` parameter is available in the following format in the `oradav.conf` file:

```
DAVParam DEFAULTLOCKTIMEOUT 86400
```

The unit of measurement for this parameter is seconds. If the parameter is not specified in the configuration file, then OracleAS Portal will create locks that will expire in one day, that is, 86400 seconds.

If the time specified for a lock expires, then any temporary document related to that lock is removed. This is expected behavior, for example:

- If Microsoft Word crashes while you are updating a document, you will lose changes to the document if the lock time has expired.
- If you perform operations such as LOCK, PUT, PUT and then close a client without specifying UNLOCK, all data that was PUT will be lost if the lock time has expired.

4.10.2 Setting Up a WebDAV Client

The steps required to set up a WebDAV client to connect to OracleAS Portal varies depending on the client. All clients will eventually request a URL. The Portal DAV URL is very similar to the URL you use to access the portal itself in your Web browser, and uses the following format:

```
http://<host>:<port>/<dav_location>
```

If you experience problems while connecting to OracleAS Portal from a WebDAV client, refer to the WebDAV troubleshooting section in the *Oracle Application Server Portal Error Messages Guide*.

4.10.3 WebDAV Clients and SSL

Although OraDAV does support Secure Socket Layer (SSL), some WebDAV clients do not. Refer to the WebDAV client's documentation for details.

4.10.4 Checking the Version of OraDAV Drivers

You can check the version of the OraDAV drivers from any Web browser, as shown in the following example:

```
http://<computer>:<port>/<dav location>/~OraDAV-Version
```

The output will be like the following example:

```
Version 1.0.3.2.3-0030
Using Container Version 1.5
```

4.10.5 Checking the Version of mod_oradav.so

You can check the version of `mod_oradav.so` by running the `oversioncheck` binary and specifying `mod_oradav.so` as its argument, as shown subsequently:

```
ORACLE_HOME/Apache/Apache/bin/oversioncheck ORACLE_HOME/Apache/oradav/lib/mod_oradav.so
```

4.10.6 Viewing Errors

Any errors that occur when a user performs actions on a portal using a WebDAV client are recorded in an error log that is created in that user's personal page (as an item titled My Error Log) the first time an OracleAS Portal-related WebDAV error occurs. This can be very helpful for interpreting the error messages reported in WebDAV clients, such as the message 'An error has occurred while trying to complete this operation' that is often displayed in Web Folders, or HTTP error numbers reported in Cadaver.

All errors are also recorded in the Apache error log file (*ORACLE_HOME*`/Apache/Apache/logs`), so if the user does not have a personal page, or is a public user, the errors can still be examined.

The `OraTraceEvents` parameter in the `oradav.conf` file ensures that certain information about an error, such as Agent, User, ECID, URL, and Method, is logged in the Apache error log file. This information is helpful to portal administrators and Oracle Support Services in resolving the error. The `OraTraceEvents` parameter is available in the following format in the `oradav.conf` file:

```
DAVParam OraTraceEvents agent
```

The information logged in the Apache error log file will be in a format similar to the following example:

```
[Wed Sep 22 10:38:46 2004] [notice] OraDAV: Agent [Secret-Agent-Man] User
[Hanckel] ECID [Viscous] URI [/orddav_var2/images/var2] Method [MKCOL].
```

For more verbose error reporting in the Apache error log file, add the following parameter to the `oradav.conf` file:

```
DAVParam ORATraceLevel 1
```

Notes:

- Remember that Oracle HTTP Server needs restarting whenever a change is made to the `oradav.conf` file. For information about how to do this, refer to the *Oracle HTTP Server Administrator's Guide*.

  You can also refer to the section "OraDAV Configuration Parameters" in the *Oracle HTTP Server Administrator's Guide* for details of other OraDAV parameters.
- The error log is not truncated and may become quite a large file. We recommend that you periodically delete this file. The next time an error is encountered a new file will be created.
- "Not Found" messages are sometimes seen in the error log because the client computer checks for the existence of a file name. If the file does not exist, the error log correctly displays a 404 error message.

4.11 Configuring Resource Proxying

If you plan to use resource proxying with remote Web providers or WSRP producers, then you will need to configure the `resourceURLKey` parameter. This key is used by the Parallel Page Engine to calculate checksums for URLs that are requested by WSRP and JPDK resource proxying. For WSRP resource proxying to work, the key must be set to an alpha-numeric value of 10 characters or more. The WSRP samples that are shipped with the product use resource proxying. Therefore, if this is not configured correctly, then you will not be able to view images in WSRP portlets. In addition, for JPDK proxying, a JNDI environment variable, also called `resourceUrlKey`, must be set for the provider. Refer to Appendix D, "Configuring the Parallel Page Engine" for more information.

To configure WSRP resource proxying, perform the following steps:

1. Open the web.xml file associated with the OC4J_PORTAL instance on the middle tier. The file is located in the following directory:

   MID_TIER_ORACLE_HOME\j2ee\OC4J_Portal\applications\portal\portal\WEB-INF\

2. Uncomment the lines that contain the `resourceURLKey` parameter definition.

   Note: By default, the `resourceURLKey` parameter is commented out.

3. Set the value for the `resourceURLKey` parameter to an alphanumeric value of 10 characters or more.
4. Save the `web.xml` file.
5. Run the following command to synchronize the manual configuration changes:

   MID_TIER_ORACLE_HOME/dcm/bin/dcmctl updateconfig

6. Run the following command to restart OC4J_Portal:

   MID_TIER_ORACLE_HOME/opmn/bin/dcmctl restartproc process-type=OC4J_Portal

Part III

Advanced Configuration Topics

Part 3 contains the following chapters:

5

Performing Advanced Configuration

This chapter discusses the configuration that must be performed to achieve some of the more advanced configurations. You must be familiar with the available administrative tools described in Section 4.1, "Getting Started with OracleAS Portal Administration" in order to perform these configurations.

This chapter contains the following sections:

5.1 Changing the OracleAS Portal Port

For information about changing ports in Oracle Application Server, refer to the *Oracle Application Server Administrator's Guide*.

> **See Also:** Section 7.5, "Viewing Oracle Application Server Port Information"

5.2 Configuring SSL

OracleAS Portal uses a number of different components (such as the Parallel Page Engine, Oracle HTTP Server, and OracleAS Web Cache) each of which may act as a client or server in an HTTP communication. As a result, each component in OracleAS Portal's middle tier must be configured individually to use the HTTPS protocol and the Secure Sockets Layer (SSL), rather than HTTP.

Chapter 6, "Securing OracleAS Portal" describes the SSL configuration options that are available with OracleAS Portal. Refer to the following sections:

- Section 6.3.2.1.2, "SSL to OracleAS Single Sign-On"
- Section 6.3.2.1.3, "SSL to OracleAS Web Cache"
- Section 6.3.2.1.4, "SSL Throughout OracleAS Portal"
- Section 6.3.2.1.5, "External SSL with Non-SSL Within Oracle Application Server"

5.3 Configuring Multiple Middle Tiers with a Load Balancing Router

This section describes how you can set up OracleAS Portal in a multiple middle-tier environment, front-ended by a Load Balancing Router (LBR) to access the same Oracle Application Server Metadata Repository.

Note: As with all out-of-the-box portal installations, this solution is best for internal deployments because it is not configured to use SSL. For the Oracle recommended way of configuring secure enterprise deployments, refer to the *Oracle Application Server Enterprise Deployment Guide*.

The purpose of an LBR is to provide a single published address to the client tier, and front-end a farm of servers that actually service the requests, based on the distribution of the requests done by the LBR. The LBR itself is a very fast network device that can distribute Web requests to a large number of physical servers.

Let us assume that we want to configure the multiple middle-tier configuration, shown in Figure 5–1. In the example, we show OracleAS Web Cache on the same computer as the OracleAS Portal and OracleAS Wireless middle tier, although they can theoretically be on different computers.

Figure 5–1 Multiple Middle-Tier Configuration with a Load Balancing Router

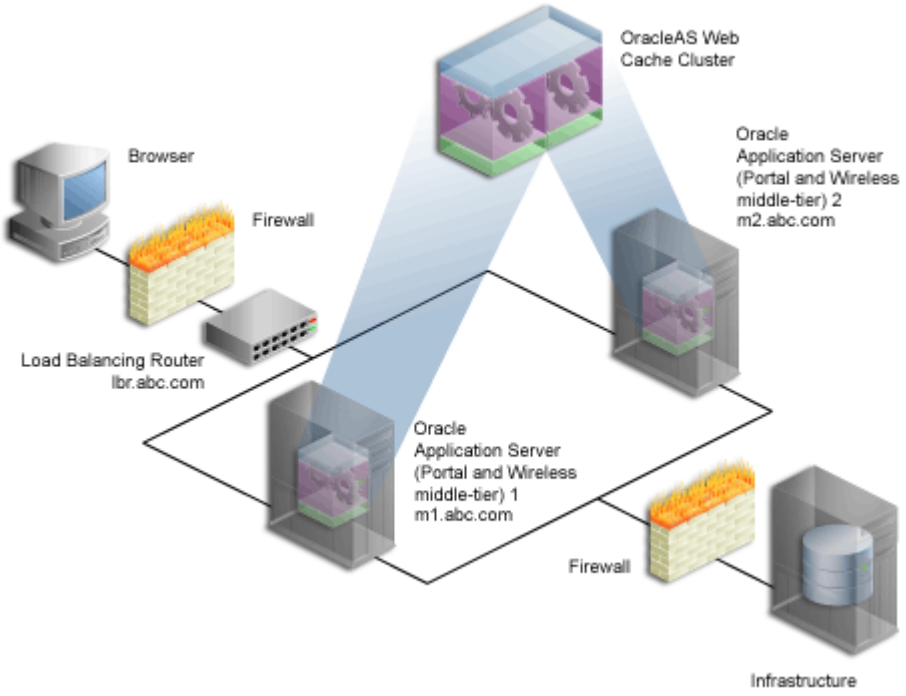


Table 5–1 Additional Information

Computer	Details
Load Balancing Router (LBR)	Computer Name: `lbr.abc.com` IP Address: `L1.L1.L1.L1` Listening Port: `80` Invalidation Port: `4001` (accessible only from inside)
Oracle Application Server (Portal and Wireless middle tier) 1 (**M1**)	Computer Name: `m1.abc.com` IP Address: `M1.M1.M1.M1` Oracle HTTP Server Listening Port: `7778` OracleAS Web Cache Listening Port: `7777` OracleAS Web Cache Invalidation Port: `4001` OracleAS Web Cache Administration Port: `4002`
Oracle Application Server (Portal and Wireless middle tier) 2 (**M2**)	Computer Name: `m2.abc.com` IP Address: `M2.M2.M2.M2` Oracle HTTP Server Listening Port: `7778` OracleAS Web Cache Listening Port: `7777` OracleAS Web Cache Invalidation Port: `4001` OracleAS Web Cache Administration Port: `4002`

Notes:

- The name and port values used in this section are for illustration purposes only, and you will need to substitute these with your own.
- Refer to the steps outlined in Section 7.5, "Viewing Oracle Application Server Port Information" to view a list of all the ports currently in use by the components of a particular Oracle Application Server instance.

Additional LBR configuration is required to successfully handle:

- Loopback Communication
- OracleAS Web Cache Invalidation

Loopback Communication

OracleAS Portal's Parallel Page Engine (PPE) retrieves page metadata information. This communication is referred to as Loopback Connections. In a default configuration, the loopback connections are local, because OracleAS Web Cache and OracleAS Portal reside on the same computer.

If an LBR is front-ending Oracle Application Server, it will need additional configuration if OracleAS Web Cache is located on the same subnet. To understand this better, let's take a look at the different parts of the loopback connections without this additional configuration.

1. The PPE sends a loopback request for page metadata when OracleAS Portal generates a page. This loopback request goes directly to the LBR.
2. The request is forwarded by the LBR to OracleAS Web Cache.
3. OracleAS Web Cache forwards the request to Portal Services, running under Oracle HTTP Server.
4. Portal Services processes the request and sends back the response to the loopback request to OracleAS Web Cache.
5. OracleAS Web Cache forwards the response to the LBR.
6. The LBR receives the response that is supposed to be routed back to the PPE.
7. The LBR detects that the source address, to which the response needs to be sent, is on the same subnet and it sends it back to OracleAS Web Cache, using the LBR's known socket connection, instead of using the PPE's socket connection.
8. OracleAS Web Cache is not listening for the request at all, and the incoming reply is dropped as there is no valid session.
9. OracleAS Portal pages time out with the error 'Timeout occurred while retrieving page metadata.'

As you can see, under normal circumstances, the behavior of the LBR is correct, as the LBR is programmed to forward all requests to OracleAS Web Cache. However, when loopback requests come from an internal network, the outcome is not desirable.

To avoid this, you must set up a Network Address Translation (NAT) bounce back rule on the LBR. This configures the LBR as a proxy for requests coming to it from inside the firewall. This setup ensures that the internal requests are forwarded correctly, and

when the response reaches the LBR, it is translated correctly and sent to the correct source address on the network (the PPE in this case).

The required steps for setting this up are discussed later. NAT bounce back is set up differently on individual LBRs. Consult your LBR's configuration guide for more information.

See Also: *Oracle Application Server Enterprise Deployment Guide*

OracleAS Web Cache Invalidation

OracleAS Portal leverages OracleAS Web Cache to cache a lot of its content. When cached content in OracleAS Web Cache changes, OracleAS Portal sends invalidation messages from the database to OracleAS Web Cache. OracleAS Portal can only send invalidation messages to one Web Cache node in an OracleAS Web Cache cluster. OracleAS Portal relies on that OracleAS Web Cache member to invalidate content in the other members of the cluster. When Oracle Application Server is front-ended by an LBR, the LBR must be configured to accept invalidation requests from the database and balance the load among the cluster members. Depending on how your routing is set up you may also need to set up NAT and open the appropriate outbound ports on your data tier. The required steps for setting this up are discussed later.

Note: You will notice in Figure 5–1 that the infrastructure is behind the LBR. The infrastructure can be one host, or distributed over multiple hosts. To configure the infrastructure properly, refer to the advanced configuration information in the *Oracle Application Server Single Sign-On Administrator's Guide*.

To configure OracleAS Portal in a multiple middle-tier environment, front-ended by an LBR, you must perform the following steps:

- Step 1: Install a Single Portal and Wireless Middle Tier (M1)
- Step 2: Configure OracleAS Portal on M1 to Be Accessed Through the LBR
- Step 3: Confirm That OracleAS Portal is Up and Running
- Step 4: Install a New Portal and Wireless Middle Tier (M2)
- Step 5: Configure the New Middle Tier (M2) to Run Your Existing Portal
- Step 6: Configure Portal Tools and Web Providers (Optional)
- Step 7: Enable Session Binding on OracleAS Web Cache
- Step 8: Confirm the Completed Configuration

5.3.1 Step 1: Install a Single Portal and Wireless Middle Tier (M1)

Install a single Portal and Wireless application server middle tier, and verify the installation. To do this perform the following steps:

1. Follow the steps described in Chapter 3, "Installing OracleAS Portal", to install a Portal and Wireless Oracle Application Server 10*g* middle tier on the first computer (**M1**). It is assumed that you use the services of an existing Oracle Application Server Infrastructure.

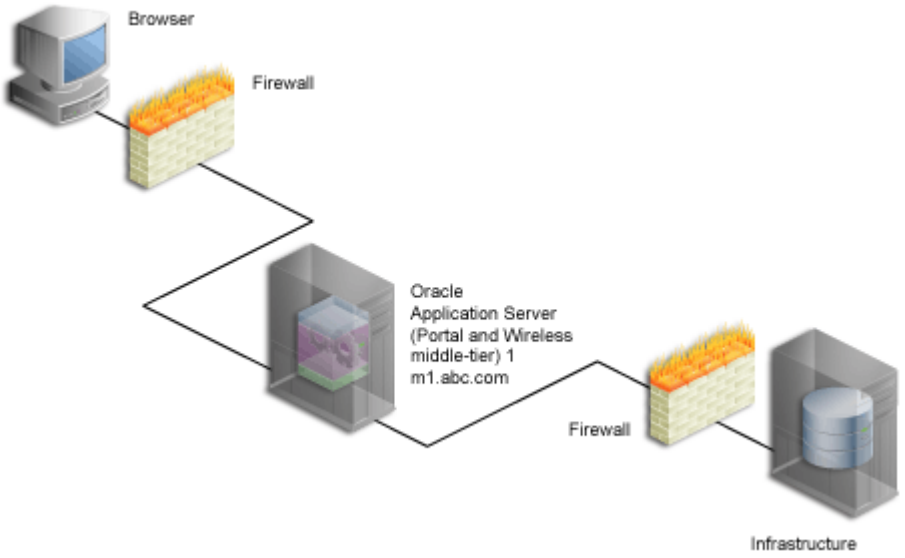
   See Also: *Oracle Application Server Installation Guide*

2. Verify that you have installed the middle tier successfully by ensuring that you can access the OracleAS Portal home page at:

   `http://m1.abc.com:7777/portal/pls/portal`

   Your configuration now looks like Figure 5–2, with the details described in Table 5–1.

Figure 5–2 Installation of OracleAS Portal Middle Tier



3. Access your `iasconfig.xml` file, located in *`MID_TIER_ORACLE_HOME`*`/portal/conf`, and verify that it looks something like Example 5–1:

Example 5–1 iasconfig.xml After the First Middle-Tier Installation

```
<IASConfig XSDVersion="1.0">
   <IASInstance Name="iAS-1.m1.abc.com" Host="m1.abc.com">
      <WebCacheComponent ListenPort="7777" InvalidationPort="4001"
InvalidationUsername="invalidator" InvalidationPassword="@Bd4D+TnIEqRc3/kleybcc70A=="
SSLEnabled="false"/>
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
   </IASInstance>
   <IASInstance Name="iAS.infra.abc.com" Host="infra.abc.com">
      <OIDComponent AdminPassword="@BVs2KPJEWC5a0l4n8lbTxUY=" PortSSLEnabled="true"
LDAPPort="3060" AdminDN="cn=orcladmin"/>
   </IASInstance>
   <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="@Beyh8p2bOWELQCsA5zRtuYc=" ConnectString="cn=iasdb,cn=oraclecontext">
      <WebCacheDependency ContainerType="IASInstance" Name="iAS-1.m1.abc.com"/>
      <OIDDependency ContainerType="IASInstance" Name="iAS.infra.abc.com"/>
      <EMDependency ContainerType="IASInstance" Name="iAS-1.m1.abc.com"/>
   </PortalInstance>
</IASConfig>
```

You now proceed with the next step where you configure OracleAS Portal to be accessed through an LBR.

5.3.2 Step 2: Configure OracleAS Portal on M1 to Be Accessed Through the LBR

To configure OracleAS Portal so it can be accessed through the Load Balancing Router (LBR), perform the following steps:

1. Configure the LBR (`lbr.abc.com`) to accept requests on port `80` and forward those to the OracleAS Web Cache port (`7777`) running on computer `m1.abc.com`. To do this, you need to:
   a. Set up a group, or *pool* on the LBR, to which individual servers can be added.
   b. Add the desired servers' IP addresses, and port numbers to the group.
   c. Create a *virtual server* that listens on port `80`, and balances load between the members of the group.
   d. Make sure the LBR translates the port that it is listening on to forward requests to the port that OracleAS Web Cache is listening on.

   Note: Consult the LBR documentation to set up the groups, and a virtual server.

2. Configure the OracleAS Portal middle tier on **M1** to allow underlying components to construct URLs based on the LBR host name (`lbr.abc.com`) and LBR port number (`80`), so that self-referential URLs rendered on OracleAS Portal pages are valid for the browser. To do this, perform the following steps:
   a. Define a virtual host, using the **Create Virtual Host** wizard, as explained in Section 5.4.1.1, "Create the Virtual Host for www.xyz.com", with the following exceptions:
      – On the **Addresses** page (Step 9), specify the host name of the LBR (`lbr.abc.com`) in the **Server Name** field for your virtual host.
      – In Step 23, specify `80` for the Port directive in the VirtualHost container.
   b. Define a second virtual host, using the **Create Virtual Host** wizard, as explained in Section 5.4.1.1, "Create the Virtual Host for www.xyz.com", with the following exceptions:
      – On the **Addresses** page (Step 9), specify the host name of **M1** (`m1.abc.com`) in the **Server Name** field for your virtual host.
      – In Step 23, specify `7777` for the Port directive in the VirtualHost container.
      – When prompted to restart the Oracle HTTP Server (Step 25), click **Yes**.
3. Define a site that matches the virtual host entry, created in the previous step (`lbr.abc.com`), using OracleAS Web Cache Manager on **M1**, as follows:
   a. Access the OracleAS Web Cache Manager on **M1**, as described in the *Oracle Application Server Web Cache Administrator's Guide*.
   b. From **Properties**, click **Sites**.
   c. Click **Create** under **Named Sites Definitions**.
   d. On the **Create Named Site** page, specify `lbr.abc.com` for the **Host** and `80` for **Port**. Keep the default values for all other fields.
   e. Click **OK**. You will now see `lbr.abc.com` in the **Named Sites Definitions** table.
4. Use OracleAS Web Cache Manager on **M1**, to map the site `lbr.abc.com` to middle tier `m1.abc.com`.
   a. In the navigation frame, select **Site-to-Server Mapping** under **Origin Servers, Sites, and Load Balancing**.

b. In the **Site-to-Server Mapping** page, select the first mapping in the table and click **Insert Above**.

c. In the **Edit/Add Site-to-Server Mapping** page, select the **Select from Site definitions** option and then select a site definition created in the previous step (`lbr.abc.com`).

d. In the **Select Application Web Servers** section, select the application server **M1** (`m1.abc.com`) specified in the **Origin Servers** page.

e. Click **Submit**.

f. Click **Apply Changes** on the top of the page.

g. In the **Cache Operations** page, click **Restart** to restart Web Cache on **M1**.

To verify that the site has been mapped correctly, navigate to the **Site-to-Server Mapping** page, and ensure that **M1** is mapped to the site `lbr.abc.com`.

5. Configure computer `m1.abc.com` so that it can resolve the LBR host name to have the correct IP address. You can either rely on DNS resolution, or make entries in the `/etc/hosts` file as follows:

```
L1.L1.L1.L1  lbr.abc.com
```

Where `L1.L1.L1.L1` is the IP address for the LBR. There is no need to restart the system after making these changes.

WARNING: Ensure that the `/etc/hosts` file does not have an entry that points the local host name to `127.0.0.1`. For example:

```
127.0.0.1 m1.abc.com
```

---

6. Configure the LBR to perform Network Address Translation (NAT) bounce back for loopback requests coming from the PPE running on `m1.abc.com`. This ensures that when the PPE makes a loopback request to OracleAS Web Cache, there are no errors.

Notes:

- NAT bounce back is set up differently on individual LBRs. Consult your LBR's configuration guide on how to set this up.
- The log files contain the NAT bounce back addresses for all loopback requests from the Parallel Page Engine (PPE), that get forwarded to OracleAS Web Cache or Oracle HTTP Server through the LBR.

---

7. Configure the LBR (`lbr.abc.com`) to accept invalidation requests from the OracleAS Metadata Repository on a separate port (`4001` in this example), so that it is forwarded to the OracleAS Web Cache running on computer `m1.abc.com` on port `4001`.

Note: The LBR does not have to listen on the OracleAS Web Cache invalidation port. On LBRs that do not have *Port Mapping* ability the port number must match the OracleAS Web Cache invalidation port.

---

a. Set up a group, or *pool* on the LBR, to which individual servers can be added.

b. Add the desired servers' IP addresses, and port numbers to the group.

c. Create a virtual server that listens on port `4001`, and balances load between the members of the group.

d. In the case where the LBR's port, that is listening for the invalidation requests, and the OracleAS Web Cache's invalidation port are different, you must ensure that the LBR translates the port that it is listening on to forward requests to the port that OracleAS Web Cache is listening on.

Notes:

- Consult the LBR documentation to set up the groups, and virtual server.
- If the Oracle Application Server Infrastructure is behind another firewall, you need to make sure that it can send invalidation messages to the LBR.

WARNING: For security reasons, the invalidation port on the LBR (port `4001`) should only be accessible from within the network.

---

8. You must manually edit the `iasconfig.xml` file, typically located in *`MID_TIER_ORACLE_HOME`*`/portal/conf`. Before editing the file, it is recommended that you make a backup copy of it. The file must be updated to have the correct *farmname*, *hostname*, and *port* information used to access OracleAS Portal, and to perform the OracleAS Web Cache invalidation, as shown in Example 5–2 (all changes are shown in bold):

Example 5–2 iasconfig.xml File Edited to Include Farm Element

```
<IASConfig XSDVersion="1.0">

   <IASFarm Name="Farm-1.lbr.abc.com" Host="lbr.abc.com">
      <WebCacheComponent ListenPort="80" InvalidationPort="4001"
InvalidationUsername="invalidator" InvalidationPassword="welcome1"
SSLEnabled="false"/>
   </IASFarm>

   <IASInstance Name="iAS.infra.abc.com" Host="infra.abc.com">
      <OIDComponent AdminPassword="@BVs2KPJEWC5a0l4n8lbTxUY="
PortSSLEnabled="true" LDAPPort="3060" AdminDN="cn=orcladmin"/>
   </IASInstance>

   <IASInstance Name="iAS-1.m1.abc.com" Host="m1.abc.com">
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
   </IASInstance>

   <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="@Beyh8p2bOWELQCsA5zRtuYc="
ConnectString="cn=iasdb,cn=oraclecontext">
      <WebCacheDependency ContainerType="IASFarm" Name="Farm-1.lbr.abc.com"/>
      <OIDDependency ContainerType="IASInstance" Name="iAS.infra.abc.com"/>
      <EMDependency ContainerType="IASInstance" Name="iAS-1.m1.abc.com"/>
   </PortalInstance>
```

```
</IASConfig>
```

> **Note:** If OracleAS Web Cache on `iAS-1.m1.abc.com` (shown in Example 5–1) is not referenced by any other OracleAS Portal instance, you can remove the entry from `iasconfig.xml`, as seen in Example 5–2.

9. Encrypt any plain text passwords in the `iasconfig.xml` configuration file. To do this, navigate to *`MID_TIER_ORACLE_HOME`*`/portal/conf`, and run the following command:

   ```
   ptlconfig -encrypt
   ```

   > **Note:** To use `ptlconfig`, the `ORACLE_HOME` environment variable must be set.

10. Register the URL changes with OracleAS Portal. Make sure that the new URLs used for accessing OracleAS Portal use the LBR host name and port, and that the OracleAS Web Cache invalidation URLs (OracleAS Web Cache host name and invalidation ports) are that of the LBR. To do this, navigate to *`MID_TIER_ORACLE_HOME`*`/portal/conf`, and run the following command:

    ```
    ptlconfig -dad <portal_dadname> -wc -site
    ```

    For example,

    ```
    ptlconfig -dad portal -wc -site
    ```

11. If the URL is using the HTTPS protocol, then the URL must be updated to use the HTTP protocol instead. Refer to the section "Setting the OracleAS Single Sign-On Query Path URL (HTTP)" for information about updating the OracleAS Single Sign-On Query Path URL.

    > **Note:** Running `ptlconfig` in the `-sso` and `-site` modes updates the OracleAS Single Sign-On query path URL, which is used by OracleAS Portal to access the list of external applications, with the URL prefix of OracleAS Single Sign-On. If this URL is using the HTTPS protocol, then the URL must be updated to use the HTTP protocol instead.

12. Run `ssoreg` to register the virtual host for which mod_osso facilitates single sign-on. The specific application URLs to be defined as partner applications within this site are defined in the `osso.conf` file. `ssoreg` is located on the middle tier in *`MID_TIER_ORACLE_HOME`*`/sso/bin`.

    The following example shows the usage of `ssoreg` on UNIX:

    ```
    MID_TIER_ORACLE_HOME/sso/bin/ssoreg.sh
      -site_name lbr.abc.com
      -mod_osso_url http://lbr.abc.com
      -config_mod_osso TRUE
      -oracle_home_path MID_TIER_ORACLE_HOME
    ```

```
-config_file MID_TIER_ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
-admin_info cn=orcladmin
-virtualhost
```

On Windows, you must run `ssoreg.bat` instead. Refer to the information about registering mod_osso in the *Oracle Application Server Single Sign-On Administrator's Guide*.

13. To enable monitoring of the LBR's front-end host and port settings for OracleAS Portal, you must edit `targets.xml` (located in *`MID_TIER_ORACLE_HOME`*`/sysman/emd/`) on **M1**, as follows:

    a. Open `targets.xml` on **M1**, using a text editor.

    b. Search for OracleAS Portal targets, that is, `TYPE="oracle_portal"`.

    c. Edit the `PortalListeningHostPort` property, to point to the LBR. For example:

    ```
    <Property NAME="PortalListeningHostPort" VALUE=http://lbr.abc.com:80/>
    ```

    d. Save the changes to `targets.xml`.

    e. Reload the targets in the Application Server Control Console:

    On Solaris and Linux, enter:

    ```
    MID_TIER_ORACLE_HOME/bin/emctl reload
    ```

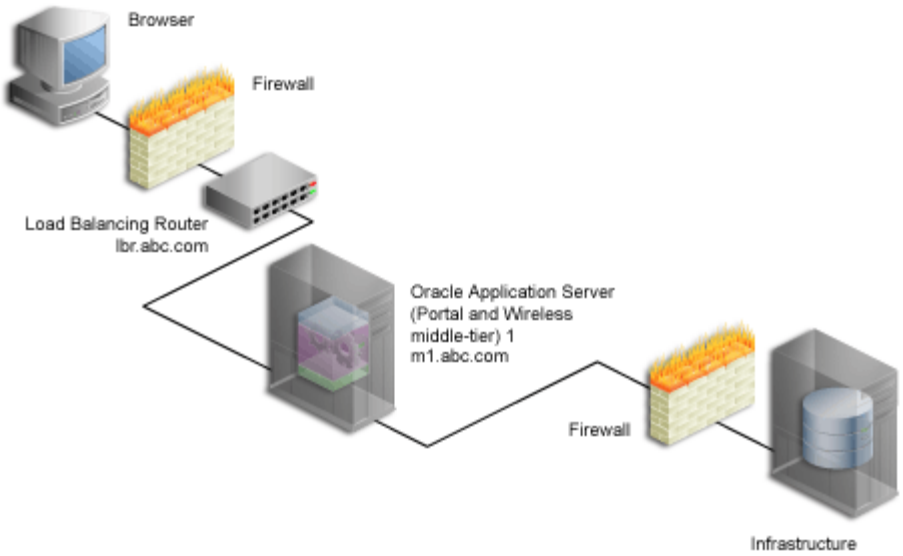
    On Windows, enter:

    ```
    MID_TIER_ORACLE_HOME\bin\emctl reload
    ```

14. Optionally, re-register the Wireless gateway URL with the load-balancer's address. See Section 5.11, "Configuring OracleAS Wireless" for more information.

After these steps, your configuration looks like Figure 5–3 with the details described in Table 5–1.

Figure 5–3 OracleAS Portal Being Accessed Through the LBR



5.3.3 Step 3: Confirm That OracleAS Portal is Up and Running

Confirm that OracleAS Portal is up and running by performing the following tests in the specified order. If a test fails, address the issues, before proceeding with the next test. To diagnose the OracleAS Web Cache, Oracle HTTP Server, and LBR configuration and logs, refer to the *Oracle Application Server Administrator's Guide*.

1. Test access to OracleAS Web Cache and Oracle HTTP Server through the LBR, by accessing a static file that is cached in OracleAS Web Cache, and make sure it works. For example, you can access the following URL:

   ```
   http://lbr.abc.com/index.html
   ```

2. Test the connection to Oracle Application Server Metadata Repository through the LBR, by accessing the following URL:

   ```
   http://lbr.abc.com/portal/pls/portal/htp.p?cbuf=test
   ```

   The response should be "**test**". If this succeeds, it means that the Oracle Application Server middle tier can connect to the OracleAS Metadata Repository. If this fails, then scan the `application.log` file for the OC4J_Portal instance for details about the request failure, and take appropriate actions.

3. Test access to OracleAS Portal, by completing the following steps:

   a. Access the OracleAS Portal home page at `http://lbr.abc.com/portal/pls/portal`. If this does not work, then scan the `application.log` file for the OC4J_Portal instance, and look for errors. The most common reason for this error is because the PPE cannot make loopback connections. For this to work:

      – Ensure that Network Address Translation (NAT) is enabled in the LBR.

      – Ensure that the middle tier on `m1.abc.com` can resolve the address of `lbr.abc.com`. To do this, run the following command from `m1.abc.com`:

        ```
        ping lbr.abc.com
        ```

   b. Click the portal login link. If this does not work, it could be due to one of the following reasons:

      – The Infrastructure middle tier is down or is not working. Check the `application.log` file for the OC4J_Portal instance in the *`INFRA_ORACLE_HOME`* for more details.

      – The partner application URL registration for OracleAS Portal is incorrect, or OracleAS Single Sign-On is down.

   c. Click some links in the portal.

   d. Confirm that content is getting cached in OracleAS Web Cache. To do this, access the OracleAS Web Cache Manager on **M1**, as described in the *Oracle Application Server Web Cache Administrator's Guide*.

      Note: The **Portal Service Monitoring** link in the **Services** portlet will not work in the multiple middle-tier configuration.

      Under **Monitoring**, click **Popular Requests**. Select **Cached** from the **Filter Objects** drop-down list, and click **Update**. If you accessed OracleAS Portal, you will see portal content (For example, URLs that contain

/portal/pls/portal). If you do not see any 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. This should provide enough content for verification.

e. Perform some basic page edits in OracleAS Portal, like adding a portlet to a page and make sure that the new content shows up. If the new content does not display properly, or if you see errors, OracleAS Web Cache invalidation is misconfigured.

5.3.4 Step 4: Install a New Portal and Wireless Middle Tier (M2)

Follow these steps to install a new Portal and Wireless middle tier on **M2** (`m2.abc.com`):

1. Set the `IASCONFIG_LOC` environment variable to point to the same location that `IASCONFIG_LOC` is pointing to on computer `m1.abc.com`. The `iasconfig.xml` file allows portal configuration to be performed from any of the hosts involved in a Web site topology. The environment variable should ideally point to a location that is accessible over a shared file system, so that installations done on different computers can reference and update the same file.

   The environment variable should be set in the second middle tier before starting the installation. To override the default location of the configuration file, you must set the environment variable `IASCONFIG_LOC` to a directory in which the file is stored, for example:

   ```
   set IASCONFIG_LOC=/usr/local/as101202
   ```

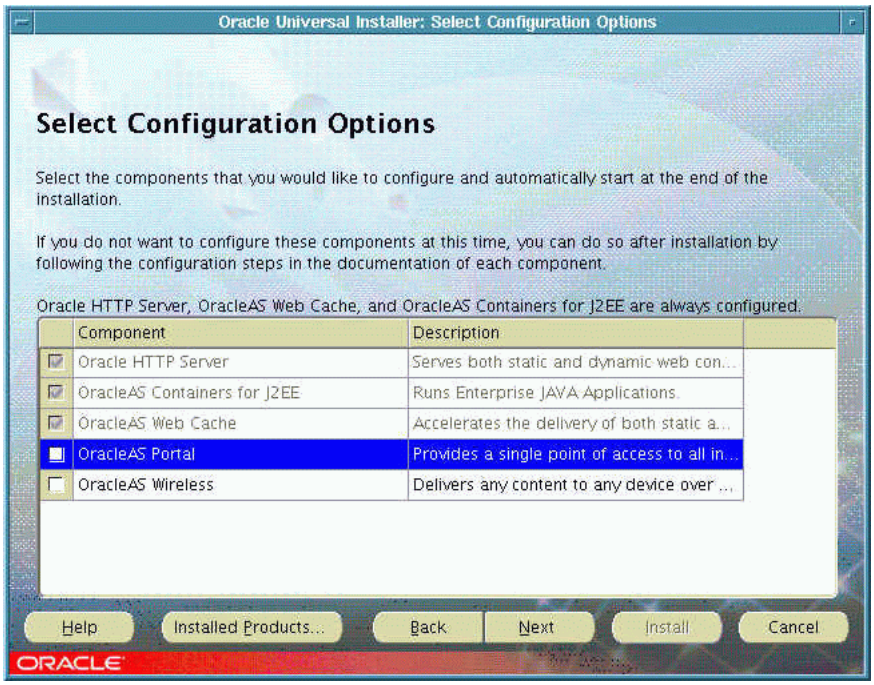
   Note: By default, `iasconfig.xml` resides in *`MID_TIER_ORACLE_HOME`*`/portal/conf`. If the Portal Dependency Settings file is accessible over a network file system, you can share the file across multiple hosts, avoiding the need to manually replicate it every time the file is modified. If the installation is running on an operating system that supports symbolic links, it is recommended that you use this mechanism to reference a shared file. If, however, the Portal Dependency Settings file is not accessible over the network, you must ensure that the file is kept up-to-date with changes to your site topology. Refer to Section A.2.4, "Updating the Portal Dependency Settings File" for more information.

2. Run Oracle Universal Installer to install a Portal and Wireless Oracle Application Server 10*g* middle tier on the second computer (**M2**).

   Note: It is recommended that you use the same physical path for installing the second middle tier. This helps when you make configuration changes on one computer and want to transfer the changes to another computer. If the physical path is different on other computers, you must ensure that the path elements are corrected after copying the files.

3. Clear the selection for **OracleAS Portal** in the **Select Configuration Options** screen during the installation of Oracle Application Server middle tier, as shown in Figure 5–4.

Figure 5–4 Select Configuration Options Screen



WARNING: Selecting OracleAS Portal in the Select Configuration Options screen overwrites your previous configuration entries in OracleAS Portal. See Section 3.3, "Configuring OracleAS Portal During and After Installation" for more information.

4. Enable OracleAS Portal to be configured using Application Server Control Console. Refer to Section 7.2.2, "Using Application Server Control Console to Configure OracleAS Portal", for more information.

Note: This will deploy the OracleAS Portal middle-tier components, but will not overwrite information in the OracleAS Metadata Repository.

5. This new installation should not have affected your previous configuration. Confirm that OracleAS Portal is up and running on **M1**, and can be accessed through the LBR. See Section 5.3.3, "Step 3: Confirm That OracleAS Portal is Up and Running" for more information about how to check this.

5.3.5 Step 5: Configure the New Middle Tier (M2) to Run Your Existing Portal

Perform the following steps, in the order specified, to configure **M2** to run your existing portal:

1. Configure the new OracleAS Portal middle tier to allow underlying components to construct URLs based on the Load Balancing Router (LBR) host name (`lbr.abc.com`) and LBR port number (`80`). To do this, perform the following steps, using the Application Server Control Console on **M2**:

a. Define a virtual host, using the **Create Virtual Host** wizard, as explained in Section 5.4.1.1, "Create the Virtual Host for www.xyz.com", with the following exceptions:
   - On the **Addresses** page (Step 9), specify the host name of the LBR (`lbr.abc.com`) in the **Server Name** field for your virtual host.
   - In Step 23, specify `80` for the Port directive in the VirtualHost container.

b. Define a second virtual host, using the **Create Virtual Host** wizard, as explained in Section 5.4.1.1, "Create the Virtual Host for www.xyz.com", with the following exceptions:
   - On the **Addresses** page (Step 9), specify the host name of **M2** (`m2.abc.com`) in the **Server Name** field for your virtual host.
   - In Step 23, specify `7777` for the Port directive in the VirtualHost container.
   - When prompted to restart the Oracle HTTP Server (Step 25), click **Yes**.

2. Copy the configuration settings for OracleAS Portal from the middle tier **M1**, to the middle tier **M2**. It is a good idea to make backup copies of the files first. To do this, perform the following steps:

   a. Copy *`MID_TIER_ORACLE_HOME`*`/Apache/modplsql/conf/dads.conf` from **M1** to **M2**.

   b. Copy *`MID_TIER_ORACLE_HOME`*`/Apache/oradav/conf/oradav.conf` from **M1** to **M2**.

   c. Copy *`MID_TIER_ORACLE_HOME`*`/Apache/modplsql/conf/cache.conf` from **M1** to **M2**.

   d. Copy *`MID_TIER_ORACLE_HOME`*`/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF/web.xml` from **M1** to **M2**.

   e. If **M1** and **M2** are installed using different physical paths, you need to make sure that the path elements are corrected after copying the files.

   f. If you have not defined `IASCONFIG_LOC` in Section 5.3.4, "Step 4: Install a New Portal and Wireless Middle Tier (M2)", you must copy the `iasconfig.xml` file from **M1** to **M2**.

3. Re-register mod_osso on **M2**. To do this, perform the following steps:

   a. Copy the *`MID_TIER_ORACLE_HOME`*`/Apache/Apache/conf/osso/osso.conf` from **M1** to **M2**. `osso.conf` is a binary file.

   b. Synchronize the Distributed Configuration Management (DCM) repository by issuing the following command:

   ```
   MID_TIER_ORACLE_HOME/Apache/Apache/bin/ssotransfer MID_TIER_ORACLE_
   HOME/Apache/Apache/conf/osso/osso.conf
   ```

   Note: This does not create any new partner applications; it enables the partner application `lbr.abc.com` for **M1** and **M2**.

4. Synchronize the manual configuration changes done on **M2** by running the following command from the directory *`MID-TIER_ORACLE_HOME`*`/dcm/bin/`:

```
dcmctl updateConfig
```

5. After copying the `dads.conf` file, you must add the necessary mod_rewrite and mod_oc4j directives to the `httpd.conf` and `mod_oc4j.conf` files respectively. To do this, perform the steps mentioned in Section E.2, "DAD Configuration File (dads.conf)" using the Application Server Control Console.
6. Run the following commands:

```
MID_TIER_ORACLE_HOME/dcm/bin/dcmctl updateConfig -ct ohs
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_Portal
```

7. Configure the computer `m2.abc.com` to resolve the LBR host name to have the correct IP address. You can either rely on DNS resolution for this, or make entries in the `/etc/hosts` file as follows:

```
L1.L1.L1.L1  lbr.abc.com
```

> **WARNING:** **Ensure that the `/etc/hosts` file does not have an entry that points the local host name to `127.0.0.1`. For example:**
>
> ```
> 127.0.0.1 m2.abc.com
> ```

8. Access the OracleAS Web Cache Manager on **M1**, as described in the *Oracle Application Server Web Cache Administrator's Guide*.
9. Use OracleAS Web Cache Manager on **M1**, to add the OracleAS Web Cache on **M2** to the OracleAS Web Cache cluster on **M1**. To do this, perform the following steps:
   a. Click **Clustering** under **Properties**.
   b. On the **Clustering** page, under the **Cluster Members** table, click **Add**.
   c. On the **Add Cache to Cluster** page, specify the following information for **M2** to be included in this Web Cache cluster:

Property	Value
Host Name	`m2.abc.com`
Admin Port	`4002`
Protocol for Admin Port	`HTTP`
Cache Name	`m2.abc.com - OracleAS Web Cache`
Capacity	`30`

> **Note:** For the value of the **Cache Name** property, you can specify any name.

   d. Click **Submit**.
   e. To verify that the OracleAS Web Cache on **M2** has been added to the cluster, locate `m2.abc.com` in the **Cluster Member** table.

For information about configuring a cache cluster, refer to the *Oracle Application Server Web Cache Administrator's Guide*.

10. Use OracleAS Web Cache Manager on **M1**, to add **M2** as an additional origin server to the OracleAS Web Cache cluster, created in the previous step. To do this, perform the following steps:
    a. Click **Origin Server**, under **Origin Servers, Sites, and Load Balancing**.
    b. In the **Origin Server** page, click **Add** under the **Application Web Servers** table.
    c. In the **Add Application Web Server** page, provide the following information:

Property	Value
Hostname	`m2.abc.com`
Port	`7778`
Routing	`ENABLED`
Capacity	`100`
Failover Threshold	`5`
Ping URL	`/`
Ping Interval	`10`
Protocol	`HTTP`

> **Note:** For the **Port** value, use the **M2**'s Oracle HTTP Server listening port.

    d. Click **Submit**.
    e. To verify that the origin server has been added properly, locate `m2.abc.com` in the **Origin Server** table.

    Refer to the section on mapping sites to origin servers in the *Oracle Application Server Web Cache Administrator's Guide* for more information.

11. Use OracleAS Web Cache Manager on **M1**, to map the site `lbr.abc.com` to the two origin servers `m1.abc.com`, and `m2.abc.com`, by performing the following steps:
    a. In the navigation frame, select **Site-to-Server Mapping** under **Origin Servers, Sites, and Load Balancing**.
    b. On the **Site-to-Server Mapping** page, select the mapping for the LBR site in the table and click **Edit Selected**.
    c. In the **Select Application Web Servers** section, select an application Web server specified in the **Origins Servers** page for **M2** (`m2.abc.com`).
    d. Click **Submit**.
    e. To verify that the site has been mapped correctly, ensure that both **M1** and **M2** are mapped to the site `lbr.abc.com` in the **Site to Server Mappings** table.

    Refer to the section on mapping sites to origin servers in the *Oracle Application Server Web Cache Administrator's Guide* for more information.

12. To save your configuration changes, click **Apply Changes** on the top of the page. Perform the following steps in the **Cache Operations** page:
    a. Click **Propagate** to propagate changes to **M2**.
    b. Click **Restart** to restart Web Caches on **M1** and **M2**.
13. Configure the LBR (`lbr.abc.com`) to forward requests on the invalidation port to OracleAS Web Cache running on the second middle tier `m2.abc.com` on port `4001`, similar to the configuration previously done for the first middle tier `m1.abc.com`.

    Note: The LBR does not have to listen on the OracleAS Web Cache invalidation port. On LBRs that do not have *Port Mapping* ability, the port number must match the OracleAS Web Cache invalidation port.

14. Configure the LBR (`lbr.abc.com`) to forward requests on port `80` to OracleAS Web Cache running on computer `m2.abc.com` on port `7777`, similar to the configuration previously done for the first middle tier `m1.abc.com`.

    Note: Consult the LBR documentation to complete this step.

15. Configure the LBR to perform Network Address Translation (NAT) bounce back for loopback requests coming from Oracle HTTP Server on `m2.abc.com`. Refer to Step 6 in Section 5.3.2, "Step 2: Configure OracleAS Portal on M1 to Be Accessed Through the LBR" section for more information.

    After these steps, your configuration looks like Figure 5–1.

    Note: For adding more middle tiers, follow the procedures outlined in Section 5.3.4, "Step 4: Install a New Portal and Wireless Middle Tier (M2)" and Section 5.3.5, "Step 5: Configure the New Middle Tier (M2) to Run Your Existing Portal", for each middle tier.

16. To enable monitoring of the LBR's front-end host and port settings for OracleAS Portal, you must edit `targets.xml` (located in *`MID_TIER_ORACLE_HOME`*`/sysman/emd/`) on **M2**, as follows:
    a. Open `targets.xml` on **M2**, using a text editor.
    b. Search for OracleAS Portal targets, that is, `TYPE="oracle_portal"`.
    c. Edit the `PortalListeningHostPort` property, to point to the LBR. For example:

    ```
    <Property NAME="PortalListeningHostPort" VALUE=http://lbr.abc.com:80/>
    ```

    d. Save the changes to `targets.xml`.
    e. Reload the targets in the Application Server Control Console:

    On Solaris and Linux, enter:

    ```
    MID_TIER_ORACLE_HOME/bin/emctl reload
    ```

    On Windows, enter:

    ```
    MID_TIER_ORACLE_HOME\bin\emctl reload
    ```

5.3.6 Step 6: Configure Portal Tools and Web Providers (Optional)

Some additional configuration is required to ensure that Portal Tools providers (OmniPortlet and Web Clipping) and locally and custom built Web providers work properly in the middle-tier environment. If OmniPortlet or any other Web providers already have personalizations in the file system, you can use the PDK-Java Preference Store Migration and Upgrade Utility to migrate the personalizations to the database and upgrade personalizations from earlier releases. Refer to Section C.11, "Using the PDK-Java Preference Store Migration and Upgrade Utility" for more information about this utility.

For the WSRP producer, the OracleAS Metadata Repository is used as the default portlet preference store. If you want to use a different preference store, then refer to the *Oracle Application Server Portal Developer's Guide* for more information.

Configuring Portal Tools Providers in the Multiple Middle-Tier Environment

Perform the following steps for Portal Tools (OmniPortlet and Web Clipping) providers to function properly in the multiple middle-tier environment:

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, like the database preference store (`DBPreferenceStore`). To configure Portal Tools providers to use `DBPreferenceStore`, perform the following steps:

   a. Navigate to the directory *`ORACLE_HOME`*`/j2ee/OC4J_Portal/applications/jpdk/jpdk/doc/dbPreferenceStore`. For example:

      ```
      cd ORACLE_HOME/j2ee/OC4J_
      Portal/applications/jpdk/jpdk/doc/dbPreferenceStore
      ```

   b. Create a user on the database containing the PORTAL schema, and grant create resource and connect privileges, using the `create user` and `grant connect` commands in SQL*Plus. Substitute the actual password in the following command. Do not use the default password of `welcome`, as this poses a security risk. For example:

      ```
      create user prefstore identified by password;
      grant connect, resource to prefstore;
      ```

   c. Connect as user `prefstore` and run the `jpdk_preference_store2.sql` script as follows in SQL*Plus:

      ```
      @jpdk_preference_store2
      ```

   d. Add the following entry to the file `data-sources.xml`, located in the directory *`ORACLE_HOME`*`/j2ee/OC4J_Portal/config`:

      ```
      <data-source
         class="com.evermind.sql.DriverManagerDataSource"
         name="omniPortletprefStore"
         location="jdbc/UnPooledConnection"
         xa-location="jdbc/xa/XAConnection"
         ejb-location="jdbc/PooledConnection"
         connection-driver="oracle.jdbc.driver.OracleDriver"
         username="prefstore"
         password="password"
      ```

```
   url="jdbc:oracle:thin:@infra.host.com:1521:orcl"
   inactivity-timeout="30"
/>
```

Note: Embedding passwords in deployment and configuration files poses a security risk. If you do not want to use a clear text password in the `data-sources.xml` file, then you can create an indirect password by performing the following steps:

1. Edit the *ORACLE_HOME*`/j2ee/OC4J_Portal/config/jazn-data.xml` file to add the `prefstore` user in the `jazn.com` realm as shown in the following example (You can create a new realm for this instead of using the `jazn.com` realm):

   ```
   <realm>
       <name>jazn.com</name>
       <users>
         <user>
            <name>prefstore</name>
            <display-name>OmniPortletprefstore</display-name>
            <description>OmniPortlet prefstore</description>
            <credentials>!welcome</credentials>
         </user>
         <user>
            ...
   ```

   Note that the password is included in the `<credentials>` element and is prefixed with an exclamation (!) mark. The next time OC4J reads the `jazn-data.xml`, it will rewrite the file with this password obfuscated.

2. Edit the *ORACLE_HOME*`/j2ee/OC4J_Portal/config/data-sources.xml` file again to use the indirect password that you created in the previous step by replacing the password attribute as follows:

   ```
   password="->jazn.com/prefstore"
   ```

   For more information about creating an indirect password, refer to the *Oracle Containers for J2EE Security Guide*.

---

e. Edit the file `provider.xml` located in the directory *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet`. Edit the `preferenceStore` tag as shown in bold:

```
<provider class="oracle.webdb.reformlet.ReformletProvider">
   <vaultId>0</vaultId>
   <session>true</session>
   <preferenceStore
class="oracle.portal.provider.v2.preference.DBPreferenceStore">
      <name>omniPortletprefStore</name>
      <connection>jdbc/PooledConnection</connection>
   </preferenceStore>
```

f. Restart OC4J_Portal.

More information about configuring the database preference store can be found at the following locations:

- The PDK article titled "Installing the DBPreferenceStore Sample (V2)", located on Portal Center at

http://portalcenter.oracle.com.

- The section on Preference Information in the *Oracle Application Server Portal Developer's Guide*.

Refer to Section C.11, "Using the PDK-Java Preference Store Migration and Upgrade Utility" for more information about the PDK Preference Store Migration Utility.

2. If you have already created an OmniPortlet instance with personalizations in the file system, then you must migrate these personalizations to the database using the Preference Store Migration Utility. To run the migration utility, perform the following steps:

   a. Navigate to the middle-tier Oracle home directory using the following command:

      ```
      cd ORACLE_HOME
      ```

   b. Run the following command to migrate OmniPortlet data from a file-based preference store (`FilePreferenceStore`) to the database preference store (`DBPreferenceStore`):

      ```
      java -classpath portal/jlib/pdkjava.jar
      oracle.portal.provider.v2.preference.MigrationTool -mode filetodb
      -pref1UseHashing true -pref1RootDirectory j2ee/OC4J_
      Portal/applications/portalTools/omniPortlet/WEB-INF
      /providers/omniPortlet -pref2User prefstore
      -pref2Password password -pref2URL
      jdbc:oracle:thin:@infra.host.com:1521:orcl
      ```

3. Typically, you perform the HTTP Proxy configuration for OmniPortlet and Web Clipping before you configure the LBR. To do it after the LBR is configured, perform the following steps:

   a. The Portal Tools configuration information is stored in the `provider.xml` file on the middle-tier server. You need to update the configuration directly on one middle tier (for example, **M1**) and then propagate it across all middle tiers front-ended by the LBR. Before you do this, you must shut down all middle tiers except **M1**.

   b. You can change the HTTP Proxy settings on the Portal Tools **Edit Provider** pages (for OmniPortlet and Web Clipping). To display these pages, access the Portal Tools providers' test pages and then click the **Edit** link next to the **HTTP Proxy** setting. The test pages are located at the following URLs:

      – OmniPortlet provider test page:

        ```
        http://lbr.abc.com/portalTools/omniPortlet/providers/omniPortlet
        ```

      – Web Clipping provider test page:

        ```
        http://lbr.abc.com/portalTools/webClipping/providers/webClipping
        ```

      Refer to Section I.1.3, "Configuring HTTP or HTTPS Proxy Settings" for more information.

   c. Propagate the changes made to the `provider.xml` file to middle tier **M2**:

      – Copy *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet/provider.xml` from **M1** to **M2**.

- Copy *ORACLE_HOME*/j2ee/OC4J_Portal/applications/portalTools/webClipping/WEB-INF/providers/webClipping/provider.xml from **M1** to **M2**.

4. Copy the *ORACLE_HOME*/j2ee/OC4J_Portal/config/data-sources.xml file from **M1** to **M2**.
5. Copy the *ORACLE_HOME*/j2ee/OC4J_Portal/config/jazn-data.xml file file from **M1** to **M2**.
6. Restart middle tier **M2**.
7. In OracleAS Portal, click **Edit Registration** for the OmniPortlet Provider on the **Providers** tab of the Navigator, under **Locally Built Providers**. Then click the **Connection** tab, and change the first part of the provider registration URL from `http://m1.abc.com:7777/` to `http://lbr.abc.com/`.
8. In OracleAS Portal, click **Edit Registration** for the Web Clipping provider on the **Providers** tab of the Navigator, under **Locally Built Providers**. Then click the **Connection** tab and change the first part of the provider registration URL from `http://m1.abc.com:7777/` to `http://lbr.abc.com/`.
9. Verify that OmniPortlet and the Web Clipping providers work properly through the LBR, by going to the test pages at the following URLs.
   - OmniPortlet Provider: `http://lbr.abc.com/portalTools/omniPortlet/providers/omniPortlet`

     If you see the "No Portlets Available" message under the Portlet Information section in the OmniPortlet Provider test page, then you may not have configured OmniPortlet correctly in Step 1. If OmniPortlet is configured correctly, the OmniPortlet and Simple Parameter Form portlets are available on the test page.
   - Web Clipping Provider: `http://lbr.abc.com/portalTools/webClipping/providers/webClipping`

     Note: If you want to use the Web Clipping provider, or the Web Page Data Source for OmniPortlet, you must also enable session binding in OracleAS Web Cache. Refer to "Step 7: Enable Session Binding on OracleAS Web Cache" for more information.

     ---
10. 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 the **Builder** link.
    b. Click the **Administrator** tab.
    c. Click the **Portlets** sub-tab.
    d. Click the **Refresh Portlet Repository** link in the **Portlet Repository** portlet.
    e. The refresh operation continues in the background.

Creating and Editing Locally Built Web Providers in the Multiple Middle-Tier Environment

Locally built providers are providers that are defined within an instance of OracleAS Portal. You typically create or edit these providers before configuring for LBR. If you are doing it after the LBR is configured, perform the following steps:

1. The Web provider information is kept in the `provider.xml` file on the middle-tier server. You need to update the configuration directly on one middle tier (for example, **M1**) and then propagate it across all middle tiers front-ended by the LBR. Before you do this, you must shut down all middle tiers except **M1**.
2. Create or edit a Web provider, and its portlets. A `provider.xml` file is created for each new provider.
3. Propagate the changes made to the files in **M1** to middle tier **M2**:
   a. Copy the *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portalTools/providerBuilder/WEB-INF/providers/<providerName>` directory from **M1** to **M2**.
   b. Copy the *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portalTools/providerBuilder/WEB-INF/deployment/<providerName>.properties` file from **M1** to **M2**.
   c. Copy the *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portalTools/providerBuilder/WEB-INF/deployment_providerui/provideruiacls.xml` file from **M1** to **M2**.
   d. Copy the entry for `<providerMap>` in *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portalTools/providerBuilder/WEB-INF/deployment_providerui/providerstore.xml` from **M1** to **M2** and change the `<warDir>` element with the appropriate value for the *ORACLE_HOME* for **M2** (shown in bold):

```
<providerMap name="MyProvider" baseLanguage="en">
    <displayName language="en" translation="myprovider"></displayName>
    <timeout>20</timeout>
    <timeoutMessage language="en" translation="Timed Out"></timeoutMessage>
    <loginFrequency>Never</loginFrequency>

<httpURL>http://lbr.abc.com:80/portalTools/builder/providers/MYPROVIDER</ht
tpURL>
    <cookieDomain>abc.com</cookieDomain>
    <serviceId>MYPROVIDER</serviceId>
    <requireSessionData>false</requireSessionData>
    <httpAppType>Portal</httpAppType>
    <warDir>ORACLE_HOME/j2ee/OC4J_
Portal/applications/portalTools/providerBuilder/WEB-INF</warDir>
    <warFile>providerBuilder</warFile>
</providerMap>
```

   e. Verify that the Web provider works properly through the LBR, by going to the test page at the URL `http://lbr.abc.com:80/portalTools/builder/providers/<providerName>`.
4. Restart the middle tier **M2**.

Configuring Custom Built Providers in a Multiple Middle-Tier Environment

A custom built provider is any Web provider that is not seeded by the OracleAS Portal installation, or created using OracleAS Portal. To configure the custom built provider, you must deploy it on the first middle tier, and register it to OracleAS Portal, using the **M1** URL (`http://m1.abc.com:7777/<webApp>/providers/<providerName>`). To configure it to work in the multiple middle-tier environment, you must perform the following steps:

1. Configure the custom built provider to use a shared preference store. Refer to the steps in the section, Configuring Portal Tools Providers in the Multiple Middle-Tier Environment, in this document.

   More information about configuring the database preference store can be found in the PDK article titled "Installing the DBPreferenceStore Sample (V2)", located on Portal Center at `http://portalcenter.oracle.com`.

2. Copy the *`ORACLE_HOME`*`/j2ee/OC4J_Portal/applications/<`*`webApp`*`>/WEB-INF/providers/<`*`providerName`*`>/provider.xml` from **M1** to **M2**.

3. In OracleAS Portal, click **Edit Registration** for the provider on the **Providers** tab of the Navigator, under **Registered Providers**. Click the **Connection** tab, and change the first part of the provider registration URL from `http://m1.abc.com:7777/` to `http://lbr.abc.com`.

4. Verify that the custom built provider works properly through the LBR, by going to the test pages at the following URL:

   `http://lbr.abc.com:80/<`*`webApp`*`>/providers/<`*`providerName`*`>`

Configuring Custom Built WSRP Producers in a Multiple Middle-Tier Environment

By default, WSRP producers store their portlet preference data in the OracleAS Metadata Repository, and will work correctly in a multiple middle-tier environment. If you want to use a custom database to store this information, then refer to the *Oracle Application Server Portal Developer's Guide* for the steps to be performed.

> **Note:** When using a custom database for portlet preference data in a multiple middle-tier environment, all WSRP producers must reference the same database schema in `data-sources.xml`.

Configuring Load Balanced Session-Based Web Providers

To configure session-based Web providers, front-ended by a Load Balancing Router (LBR), the login frequency should be set to "Once Per User Session", on the provider information page, and the LBR must be configured to do cookie-based routing. To set the login frequency, take the following steps:

1. Log in to OracleAS Portal. On the **Portal Builder** page.
2. In Portal Builder, click the **Administer** tab and then the **Portlets** tab.
3. Under **Remote Providers**, enter the name of the Web provider you want to configure, and then click **Edit**.
4. Click the **Connection** tab.

5. Under **User/Session Information**, set the **Login Frequency** to **Once Per User Session**.
6. Click **OK**.

Refer to the specific documentation of your LBR for information about how to configure an LBR to do cookie-based routing.

Editing an External Application Login URL

If your external application is hosted on the middle tier, then you need to update the external application login URL in the OracleAS Single Sign-On Server. You can do this by following the procedure in the "Editing an External Application" section in the *Oracle Application Server Single Sign-On Administrator's Guide*. Change the first part of the login URL from `http://m1.abc.com:7777/` to `http://lbr.abc.com/`.

5.3.7 Step 7: Enable Session Binding on OracleAS Web Cache

The *Session Binding* feature in OracleAS Web Cache is used to bind user sessions to a given origin server to maintain state for a period of time. Although almost all components running in a default OracleAS Portal middle tier are stateless, session binding is required for two reasons:

- The Web Clipping Studio, used by both the Web Clipping portlet and the Web Page Data Source of OmniPortlet, uses HTTP session to maintain state, for which session binding must be enabled. Refer to Appendix I, "Configuring the Portal Tools Providers" for more information about Web Clipping.
- 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. Refer to Section 1.3.2, "Understanding Portal Cache" for details on the portal cache.

> **Note:** Regardless of whether you have configured an LBR in your topology, you must enable session binding in OracleAS Web Cache, if you have more than one middle tier. In this configuration OracleAS Portal does not require session binding to be set up on the LBR.

To make OracleAS Web Cache bind the portal user session to the OracleAS Portal middle tier, perform the following steps:

1. In OracleAS Web Cache Manager on either **M1**, or **M2**, click **Session Binding** under **Origin Servers, Sites, and Load Balancing**.
2. In the **Session Binding** page, select the LBR site name (`lbr.abc.com:80`) in the table, and then click **Edit Selected**.
3. From the **Please select a session** drop-down list, change the session value to **Any Set-Cookie**.
4. From the **Please select a session binding mechanism** drop-down list, select **Cookie-based**.
5. Click **Submit** to apply the new settings to the site `lbr.abc.com:80`.
6. To save your configuration changes, click **Apply Changes** at the top of the page.
7. On the **Cache Operations** page, click **Propagate** to propagate the changes.
8. Click **Restart** to restart OracleAS Web Cache on **M1** and **M2**.

5.3.8 Step 8: Confirm the Completed Configuration

To verify that your complete configuration is working as expected, perform the following steps:

1. To clear the contents stored in OracleAS Web Cache, restart **M1** and **M2**, as follows:
   a. Access the Oracle Enterprise Manager 10*g* Application Server Control Console. For example:

      ```
      http://m1.abc.com:1810
      ```

      For details, see Section 7.2.1, "Accessing the Application Server Control Console".
   b. Click the **M1** instance.
   c. Click **Restart All**.
   d. Repeat the steps for **M2**.
2. Test access to OracleAS Portal through the LBR, by completing the following steps:
   a. Access the OracleAS Portal home page at `http://lbr.abc.com/portal/pls/portal`.
   b. Click the portal login link.
   c. Click some links in the portal.
   d. Confirm that content is getting cached in OracleAS Web Cache. To do this, access the OracleAS Web Cache Manager on **M1** as described in the *Oracle Application Server Web Cache Administrator's Guide*.

      Under **Monitoring**, click **Popular Requests**. Select **Cached** from the **Filter Objects** drop-down list, and click **Update**. If you accessed OracleAS Portal, you will see portal content (For example, URLs that contain `/portal/pls/portal`).

      Perform some basic page edits in OracleAS Portal, like adding a portlet to a page and make sure that the new content shows up. If the new content does not display properly, or if you see errors, OracleAS Web Cache invalidation is misconfigured.

5.4 Configuring Virtual Hosts

The Oracle HTTP Server supports the configuration of virtual hosts. Virtual hosts allow a single computer to appear as any number of different sites. You can, for example, configure a computer to represent both `www.abc.com` and `www.xyz.com`. Another example would be configuring a computer to represent both `my.oracle.com` and `oraclepartnernetwork.oracle.com`. To configure virtual hosts with OracleAS Portal, you must set this up on the Oracle HTTP Server. Additional Oracle Application Server Web Cache and Oracle Application Server Single Sign-On configuration is also required.

Portal pages are cached in OracleAS Web Cache with the host name of the host that you access first. Subsequent requests to the same page will always contain links with that host name irrespective of which host you are accessing.

For example, if you access Page A using virtual host `www.abc.com`, then all links in Page A will show up relative to `www.abc.com`. If another user accesses the same page, Page A, using the virtual host `www.xyz.com`, then due to the aliasing in OracleAS

Web Cache, all links created for this page will still reference `www.abc.com` and clicking on these links will result in portal pages being served from `www.abc.com`.

Unless the pages served from both virtual hosts are mutually exclusive, that is, portal pages served from site `www.abc.com` are not being served from `www.xyz.com`, users will be bouncing back and forth between the two virtual hosts. If this is not desired, then you can set up a dedicated intranet and Internet for OracleAS Portal as described in the *Oracle Application Server Enterprise Deployment Guide*.

Note: If your intent is only to change the host name of your middle tier, refer to the *Oracle Application Server Administrator's Guide*.

Let's assume that your server name is `www.abc.com`, and you connect to OracleAS Portal at `http://www.abc.com:7779/portal/pls/portal`. The IP address of the computer that the middle tier is installed on is `196.12.67.8`.

You want to access OracleAS Portal at `http://www.abc.com:7779/portal/pls/portal`, using the real server name, and `http://www.xyz.com:7779/portal/pls/portal`, using a virtual host name, where both URLs resolve to the same IP address.

In this example, port `7779` is the OracleAS Web Cache listening port, and port `7778` is the Oracle HTTP Server listening port.

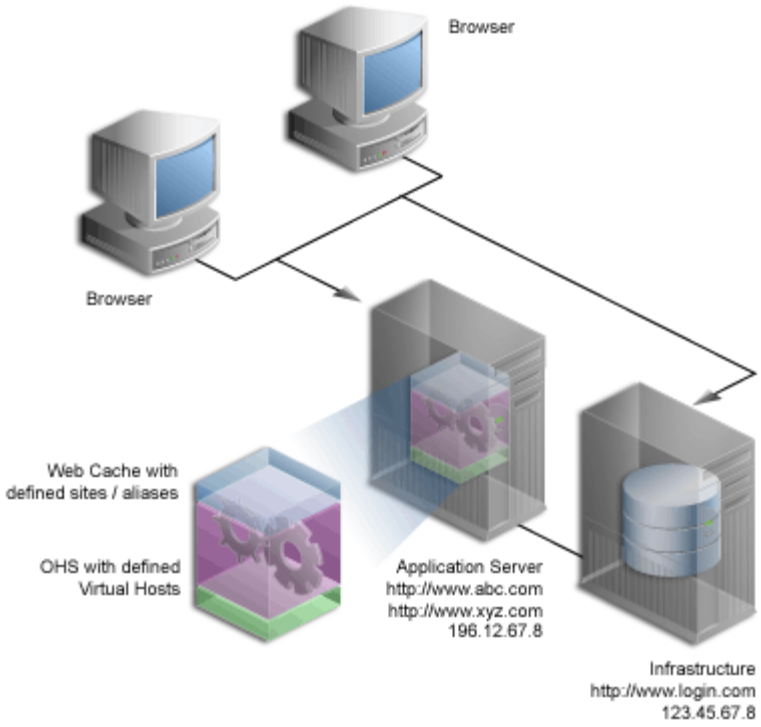
Let's also assume that the OracleAS Single Sign-On is installed on a different computer with the IP address `123.45.67.8`, and accessed at the URL `http://www.login.com:7777/pls/orasso`.

Notes:

- The IP addresses used in this example are for illustration purposes only and may not be valid IP addresses.
- The name and port values used in this section are for illustration purposes only and you will need to substitute these with your own.
- In this section we only describe how to configure virtual hosts for the OracleAS Portal middle tier, and this does not modify the host name for OracleAS Single Sign-On. For more information about how to customize the OracleAS Single Sign-On host name, refer to the information about deploying OracleAS Single Sign-On with a proxy server, in the *Oracle Application Server Single Sign-On Administrator's Guide*, and the *Oracle Application Server Administrator's Guide*.

Figure 5–5 shows the previously described configuration. OracleAS Web Cache and the Oracle Application Server are shown as residing on the same middle-tier computer, although they can exist on different computers.

Figure 5–5 Virtual Host Overview



Note: The domain names `www.abc.com`, `www.xyz.com`, and `www.login.com` must be valid domain names, and you must be able to ping them.

To configure the virtual host, perform the following steps in the specified order:

1. Create Virtual Hosts
2. Configure OracleAS Web Cache
3. Register OracleAS Portal with OracleAS Single Sign-On
4. Verify the Configuration

5.4.1 Create Virtual Hosts

You must create **virtual hosts** entries in the `httpd.conf` file for the virtual host name `www.xyz.com`, and for the real server name `www.abc.com`. To define the virtual hosts, use Oracle Enterprise Manager 10*g* Application Server Control Console to perform the following steps:

- Create the Virtual Host for www.xyz.com
- Create the Virtual Host for www.abc.com

Once you have finished this step, do the following:

1. Verify the httpd.conf File
2. Verify That the Virtual Hosts Are Configured Correctly

5.4.1.1 Create the Virtual Host for www.xyz.com

To create the virtual host for `www.xyz.com`:

1. Access the Oracle Enterprise Manager 10*g* Application Server Control Console.

   For details, see Section 7.2.1, "Accessing the Application Server Control Console".

2. Click the link for the middle tier where OracleAS Portal is installed.
3. Click the **HTTP Server** link.
4. Click the **Virtual Hosts** link.
5. Click the **Create** button in the **Virtual Hosts** page.
6. On the **Introduction** page, click **Next** to create a new virtual host, using the Virtual Host Creation wizard.
7. On the **General** page, provide the information listed in Table 5–2.

Table 5–2 Virtual Host Information

Virtual Host Information	Value
Document Root Directory	*ORACLE_HOME*`/Apache/Apache/htdocs`
Directory Index	Can be left blank
Server Administration E-Mail	Valid e-mail address
Virtual Host Type	name-based

8. Click **Next**.
9. On the **Addresses** Page, provide the following information in the **Server Name** field for your virtual host:

   ```
   www.xyz.com
   ```

10. Select the option **Listen on all the main server IP addresses**.
11. Click **Next**.
12. On the **Ports** page, select **Listen on a specific port**, and select the Oracle HTTP Server listening port, `7778` in our example, from the port drop-down list.
13. Click **Next**.
14. On the **Error Log** page, select all default values.
15. Click **Next**.
16. Review the summary on the **Summary** page.
17. Click **Finish**.
18. When prompted to restart Oracle HTTP Server, click **No**.
19. Ensure that your server name, `www.xyz.com`, is listed in the table.
20. Click the **Administration** link.
21. Click **Advanced Server Properties**.
22. Select **httpd.conf**.
23. Add the `Port` and the `Rewrite` directives in the VirtualHost container, as follows (shown in bold text):

    ```
    NameVirtualHost *:7778
    ```

```
<VirtualHost *:7778>
     ServerName www.xyz.com
     Port 7779
     ServerAdmin you@your.address
     RewriteEngine On
     RewriteOptions inherit
</VirtualHost>
```

24. Click **Apply**.
25. When asked to restart Oracle HTTP Server, click **No**.

5.4.1.2 Create the Virtual Host for www.abc.com

To create the virtual host for `www.abc.com`:

1. In Section 5.4.1.1, "Create the Virtual Host for www.xyz.com" follow steps 1 through 8.
2. On the **Addresses** Page (Step 9), provide the following information in the **Server Name** field for your virtual host:

   ```
   www.abc.com
   ```

3. In Section 5.4.1.1, "Create the Virtual Host for www.xyz.com" follow steps 10 through 24.
4. When prompted to restart the Oracle HTTP Server, click **Yes**.

5.4.1.3 Verify the httpd.conf File

After configuring virtual hosts for `www.abc.com` and `www.xyz.com`, take a look at your `httpd.conf` file, using Application Server Control Console, as follows:

1. Access the Oracle Enterprise Manager 10*g* Application Server Control Console.
2. Click the link for the application server where OracleAS Portal is installed.
3. Click the **HTTP Server** link.
4. Click the **Administration** link.
5. Click **Advanced Server Properties**.
6. Select **httpd.conf**.

Your `httpd.conf` file should have the following new section:

```
NameVirtualHost *:7778

<VirtualHost *:7778>
   ServerName www.xyz.com
   Port 7779
   ServerAdmin you@your.address
   RewriteEngine On
   RewriteOptions inherit
</VirtualHost>

<VirtualHost *:7778>
   ServerName www.abc.com
   Port 7779
   ServerAdmin you@your.address
   RewriteEngine On
   RewriteOptions inherit
```

```
</VirtualHost>
```

Entries for the virtual hosts can vary depending on the existing content of the `httpd.conf` file, but you must have virtual host entries for both `www.abc.com` and `www.xyz.com`.

Note:

- The `httpd.conf` file can also be updated manually. The file can be edited manually, to contain the right VirtualHost directives, as shown previously.

  To synchronize the manual configuration changes done on the middle tier, run ORACLE_HOME/dcm/bin/dcmctl as follows:

  ```
  dcmctl updateConfig -ct ohs
  ```

  Finally, restart Oracle HTTP Server, by running the following command from *ORACLE_HOME*/opmn/bin:

  ```
  opmnctl restartproc type=ohs
  ```

- If your site name is not registered with the DNS, you need to update the hosts file on your client computer as follows:

  On Windows, this file is typically located in the directory `C:\WINNT\system32\drivers\etc`. Here is an example of the hosts file on Windows:

  ```
  # Copyright (c) 1993-1995 Microsoft Corp.
  #
  # This is a sample HOSTS file used by Microsoft TCP/IP
  # for Windows.
  #
  127.0.0.1 localhost
  196.12.67.8 www.abc.com
  196.12.67.8 www.xyz.com
  ```

  On UNIX, the file is typically located in the directory `/etc/hosts`. You do not have to restart the system after making these changes.

5.4.1.4 Verify That the Virtual Hosts Are Configured Correctly

Verify that both the server name, and the virtual host are working, by accessing these URLs:

- `http://www.xyz.com:7779/portal/pls/portal`
- `http://www.abc.com:7779/portal/pls/portal`

5.4.2 Configure OracleAS Web Cache

The site `www.abc.com` is already defined in OracleAS Web Cache. Additionally, you must create a *site alias* in OracleAS Web Cache, to make the multiple virtual hosts transparent to the OracleAS Metadata Repository. Note that `www.abc.com` must be set up as a site, while `www.xyz.com` must be defined as a site alias. This way, invalidation messages sent to OracleAS Web Cache invalidate content that is cached for both sites.

See Also: *Oracle Application Server Web Cache Administrator's Guide* for information about setting up a site alias.

5.4.3 Register OracleAS Portal with OracleAS Single Sign-On

For Oracle Application Server Single Sign-On to work properly, it must always be referenced by a partner application with the same host name in the URL. This is because cookies are sent back only to the host that generated them. So, in the preceding example, OracleAS Single Sign-On must always be referenced as `http://www.login.com`. Therefore, you must register `www.abc.com`, and `www.xyz.com` as partner applications. To do this:

1. Add a partner application entry for `www.abc.com`, by running the script `ptlconfig`, as follows:

   ```
   ptlconfig -dad portal -sso -host www.abc.com -port 7779
   ```

2. Add a partner application entry for `www.xyz.com`, by running the script `ptlconfig`, as follows:

   ```
   ptlconfig -dad portal -sso -host www.xyz.com -port 7779
   ```

3. Run `ssoreg` to register the virtual host, `www.abc.com`, for which mod_osso facilitates single sign-on. The specific application URLs to be defined as partner applications within this site are defined in the file `osso.conf`. `ssoreg` is located on the middle tier in *`MID_TIER_ORACLE_HOME`*`/sso/bin`.

   The following example shows the usage of `ssoreg` on UNIX:

   ```
   MID_TIER_ORACLE_HOME/sso/bin/ssoreg.sh
     -site_name www.abc.com
     -mod_osso_url http://www.abc.com:7779
     -config_mod_osso TRUE
     -oracle_home_path MID_TIER_ORACLE_HOME
     -config_file MID_TIER_ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
     -admin_info cn=orcladmin
   ```

   On Windows, you must run `ssoreg.bat` instead. Refer to the *Oracle Application Server Single Sign-On Administrator's Guide* for more information.

4. Run `ssoreg` to register the virtual host, `www.xyz.com`, for which mod_osso facilitates single sign-on. The specific application URLs to be defined as partner applications within this site are defined in the file `osso.conf`.

   The following example shows the usage of `ssoreg` on UNIX:

   ```
   MID_TIER_ORACLE_HOME/sso/bin/ssoreg.sh
     -site_name www.xyz.com
     -mod_osso_url http://www.xyz.com:7779
     -config_mod_osso TRUE
     -oracle_home_path MID_TIER_ORACLE_HOME
     -config_file MID_TIER_ORACLE_HOME/Apache/Apache/conf/osso/osso_xyz.conf
     -admin_info cn=orcladmin
     -virtualhost
   ```

   Note that the `-config_file` parameter refers to the file `osso_xyz.conf`.

5. You must edit the Virtual Host container for `www.xyz.com` as follows (changes shown in bold).

   ```
   <VirtualHost *:7778>
      ServerName www.xyz.com
      Port 7779
      ServerAdmin you@your.address
      RewriteEngine On
      RewriteOptions inherit
   ```

```
    OssoConfigFile MID_TIER_ORACLE_HOME/Apache/Apache/conf/osso/osso_xyz.conf
    OssoIpCheck off
</VirtualHost>
```

Refer to the information about registering mod_osso in the *Oracle Application Server Single Sign-On Administrator's Guide*.

5.4.4 Verify the Configuration

To verify that the virtual hosts are set up correctly, connect to OracleAS Portal using either of the following URLs:

- `http://www.abc.com:7779/portal/pls/portal`
- `http://www.xyz.com:7779/portal/pls/portal`

You should get a login screen at `http://www.login.com` on the first login and must be able to log in successfully. A subsequent login from the other virtual host should result in a successful single sign-on without a prompt for login credentials.

5.5 Configuring OracleAS Portal to Use a Proxy Server

You can configure OracleAS Portal to use proxy servers to connect to providers and Web sites outside of your firewall.

Notes:

- Oracle Text uses these proxy server settings when indexing URL content. See Section 8.3.6.4, "URL Index Proxy Settings" for more information.
- To configure OracleAS Portal to use a proxy server, you must be a portal administrator.
- For the Oracle recommended way of configuring secure enterprise deployments, refer to the *Oracle Application Server Enterprise Deployment Guide*.

To specify a proxy server:

1. In the **Services** portlet, click **Global Settings**.

   The **Services** portlet is on the **Administer** tab of the **Builder** page.

2. Click the **Proxy** tab.
3. In the **HTTP Proxy Host** field, enter the address for the HTTP proxy server that you want to use to access applications outside your firewall, for example, `myproxy.mycompany.com`. Do not prefix `http://` to the proxy server name.
4. In the **Port** field, enter the port number for the proxy server. The port number defaults to `80` if no value is specified.

   Note: Contact your server administrator for the names of servers running proxy software and their port numbers.

   ---

5. Click **Add**.

You can now use this proxy server for connections between the portal and Web providers or WSRP producers. You can also use this proxy for other connections, for example, to connect to the URLs specified for URL items.

6. In the **Select Proxy** section, choose the proxy server you want to use for such connections. Choose **None** if you do not want to use a proxy server for non-provider connections.
7. In the **No Proxy Servers for Domains beginning with** field, enter the domains for which the proxy server should not be used.

> **Note:** The domains must begin with a period (.), for example, `mycompany.com`. Separate multiple domains with a comma (,).

8. Click **OK**.

You will find additional information about how to set up proxy servers in the paper "*A Primer on Proxy Servers*" on the Oracle Technology Network (OTN), http://www.oracle.com/technology/.

5.6 Configuring Reverse Proxy Servers

For information about configuring reverse proxy servers for OracleAS Portal and OracleAS Single Sign-On, refer to the *Oracle Application Server Enterprise Deployment Guide*.

5.7 Configuring a Dedicated Intranet and Internet for OracleAS Portal

You can configure OracleAS Portal to be accessible from within a company network as well as from external clients. The following links describes some important characteristics of this configuration, and provides instructions on how to configure OracleAS Portal for this purpose:

- *Oracle Application Server Enterprise Deployment Guide*
- The paper "*Expose your Intranet Portal to the Outside World in a Secured Manner*" on OTN, http://www.oracle.com/technology/products/ias/portal/pdf/admin_security_1014_secured_inside_outside.pdf.

5.8 Managing OracleAS Portal Content Cached in OracleAS Web Cache

Oracle Application Server Web Cache offers caching, page assembly, and compression features. OracleAS Web Cache accelerates the delivery of both static and dynamic Web content, and provides load balancing and failover features for Oracle Application Server. Refer to Section 1.3, "Understanding Caching in OracleAS Portal" for an overview of how caching works in OracleAS Portal.

This section discusses how to configure OracleAS Portal to work with OracleAS Web Cache.

This section contains the following topics:

- Managing Oracle Application Server Web Cache
- Configuring Portal Web Cache Settings Using Application Server Control Console
- Managing Portal Content Cached in OracleAS Web Cache

- Clearing the Cache Invalidation Queue Through SQL*Plus
- Managing the Invalidation Message Processing Job

5.8.1 Managing Oracle Application Server Web Cache

In previous releases, you had to use OracleAS Web Cache Manager to configure OracleAS Web Cache. In this release, you have two choices:

- You can use Oracle Enterprise Manager 10*g* Application Server Control Console to configure OracleAS Web Cache along with other Oracle Application Server components. Refer to *Oracle Application Server Web Cache Administrator's Guide* for more information.
- You can continue to use standalone tool OracleAS Web Cache Manager. Refer to *Oracle Application Server Web Cache Administrator's Guide* for more information.

These interfaces enable you to update to the OracleAS Web Cache configuration file, `webcache.xml`.

5.8.2 Configuring Portal Web Cache Settings Using Application Server Control Console

Use the Application Server Control Console to change OracleAS Web Cache settings that OracleAS Portal uses, such as the host name, and the invalidation port number. You configure these settings on the **Portal Web Cache Settings** page.

When you change OracleAS Web Cache properties in the **Portal Web Cache Settings** page, the properties are saved to `iasconfig.xml`, but not to the `webcache.xml` file. You must navigate back to the **Web Cache Administration** page in Application Server Control Console to make the appropriate changes.

> **See Also:** Section 7.3.3, "Portal Web Cache Settings Link" for a detailed description of how to use the **Portal Web Cache Settings** page.

5.8.3 Managing Portal Content Cached in OracleAS Web Cache

From the OracleAS Portal user interface, you can perform various other tasks to manage portal content cached in OracleAS Web Cache. You can either clear the entire portal content cached in OracleAS Web Cache, or clear content for each portal user.

> **Caution:** Clearing the cache results in cache misses on subsequent requests and may degrade the portal's performance until the cache is repopulated.

You may want to clear the cache if a user's group membership has changed, to remove the cache entries for that user, so that he or she has new privileges. Similarly, if you change a user or group's privileges on an object, you can clear the cache entries for that object.

To clear the entire cache, or to clear the cache for a particular user, you must be the portal administrator. To clear the cache for a particular portal object, you must have at least **Manage** privileges on the object.

The following sections describe the actions that can be performed using OracleAS Portal in more detail:

- Clearing the Entire Web Cache

5.8.3.1 Clearing the Entire Web Cache

You can clear the entire Web Cache by performing the following steps:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.
2. Click the **Cache** tab.
3. Select **Clear The Entire Web Cache**.
4. Click **Apply** or **OK** to clear the cache.

> **Note:** This clears all the page URLs and style sheets but not the portal images.

5.8.3.2 Clearing the Cache for a Particular User

You can clear the cache for a particular user by performing the following steps:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.
2. Click the **Cache** tab.
3. In the **Clear Cache For User** field, enter the name of the user for whom you want to clear the cache.

> **Note:** If you are not sure of the user name, click the **Browse Users** icon and select from the list provided.

4. Click **Apply** or **OK** to clear the cache for the specified user.

> **Note:** Alternatively, you can clear the cache for a particular user by editing the user's portal profile.

5.8.3.3 Setting the Expiry Time for Invalidation-based Caching

With invalidation-based caching, a cache entry is purged when the portal or a provider sends a message informing OracleAS Web Cache that the object has changed (for example, when an item is edited). However you can also set an expiry time for cache entries. A cache entry that reaches this time limit is purged, even if OracleAS Web Cache has not received an invalidation message for it.

> **Note:** To set the expiry time for invalidation-based cache entries, you must be the portal administrator.

To set the expiry time for invalidation-based cache entries:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Cache** tab.

3. In the **Maximum Expiry Time** field, enter the maximum amount of time (in minutes) a cache entry should remain in the cache before being purged.

   Note: Choosing a small value for this leads to cache misses more frequently because the cache expires more often. However, choosing a large value might lead to stale content. Avoid a value of 0 because it makes all the portal content non-cacheable.

4. Click **OK**.

5.8.3.4 Clearing the Cache for a Particular Portal Object

You can clear cache entries for page groups, pages, Portal Templates for pages, portlets in the Portlet Repository, Portal DB Providers, and Portal DB Provider components, by performing the following steps:

1. In the Navigator, drill down to the object with which you want to work.
   - For page groups, pages, and Portal Templates for pages, click **Properties**, then click the **Access** tab.
   - For Portal DB Providers, and Portal DB Provider components, click **Grant Access**.
   - For portlets, click **Edit Root Page** next to the **Portlet Repository** page group, drill down to the page that contains the portlet, click the **Actions** icon next to the portlet, and then click **Access**.
2. Click **Clear Cache**.
3. Click **OK**.

5.8.4 Clearing the Cache Invalidation Queue Through SQL*Plus

Sometimes, the cache invalidation queue can grow excessively large as a result of user actions. For example, repeated granting of security privileges on a page to a group with a large number of members will place one soft invalidation in the queue for every user for every grant.

Some soft invalidations may not be necessary, but OracleAS Portal may not be able to determine this. For example, if a group's privileges on a page are upgraded from **View** to **Fully Personalize**, and no member of the group has viewed the page yet, then no invalidation is necessary. However, Portal does not have a record of who has viewed the page. Therefore, it proceeds with the soft invalidation configured to use the security change.

The portal administrator can check the number of soft invalidations in the queue by executing the following query in SQL*Plus as the portal schema owner:

```
select count(1) from wwutl_cache_inval_msg$ where process_type=2;
```

The portal administrator can check the total number of invalidations, hard or soft, in the queue by executing the following query in SQL*Plus as the portal schema owner:

```
select count(1) from wwutl_cache_inval_msg$;
```

The number of rows in the table **wwutl_cache_inval_msg$** that can be considered excessive depends, to some extent, on the speed of the infrastructure running the database. Typically, 50000 messages will slow down the soft invalidation job, and sending 50000 invalidation messages to OracleAS Web Cache will introduce a network load, as OracleAS Portal communicates with the OracleAS Web Cache invalidation port.

If the soft invalidations are found to be unnecessary, the portal administrator can perform the following query in SQL*Plus as the portal schema owner:

```
delete from wwutl_cache_inval_msg$ where process_type=2;
```

This removes soft invalidations from the queue.

If the soft invalidations are necessary but there is an excessively large number, the portal administrator can clear the cache invalidation queue using the following command:

```
truncate table  wwutl_cache_inval_msg$;
```

The portal administrator should then clear the entire cache through the OracleAS Portal user interface. Refer to Section 5.8.3.1, "Clearing the Entire Web Cache" for information about performing this task.

5.8.5 Managing the Invalidation Message Processing Job

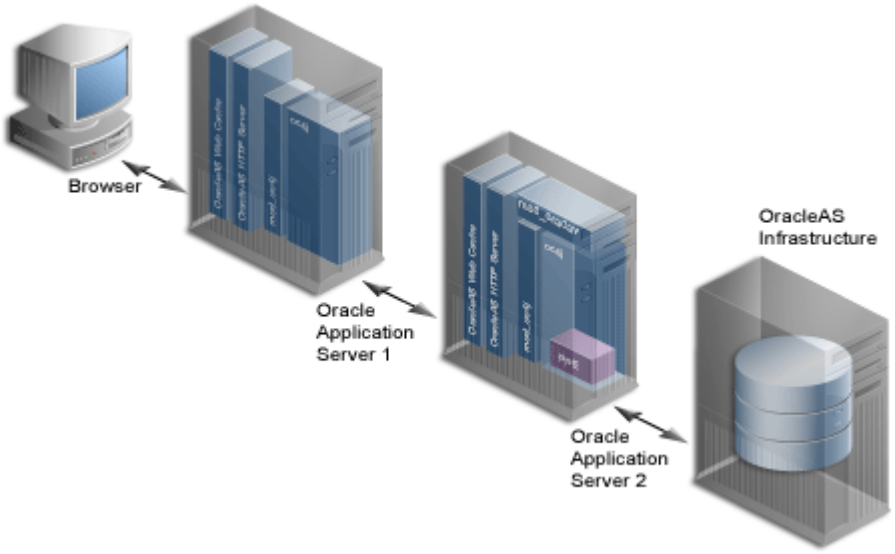
OracleAS Portal uses invalidation messages to expire objects in the cache. You can use the `cachjsub.sql` script to configure the frequency at which the invalidation job executes. Refer to Section C.1, "Managing the Invalidation Message Processing Job Using cachjsub.sql" for more information and instructions on how to run `cachjsub.sql`.

5.9 Configuring OracleAS Portal to Use a Dedicated OracleAS Web Cache Instance

You can deploy OracleAS Web Cache on a dedicated server that front-ends one or more OracleAS Portal middle-tier servers. OracleAS Web Cache performs well even on commodity hardware, so a dedicated deployment does not have to be costly in terms of hardware expenditure. In general, it is recommended to use a computer with 1 GB of memory. Both the cache server and middle-tier server need to use a high speed network card to ensure site performance. Refer to Section 1.3, "Understanding Caching in OracleAS Portal" for an overview of how caching works in OracleAS Portal.

To set up this topology, the administrator of the Web site needs to disable the OracleAS Web Cache that was installed on the same computer as OracleAS Portal middle tier, and set up a new OracleAS Web Cache instance on a dedicated server. Figure 5–6 shows the topology where OracleAS Portal uses a dedicated OracleAS Web Cache instance.

Figure 5–6 OracleAS Portal Using a Dedicated OracleAS Web Cache Instance



5.9.1 Understanding Installation Prerequisites and Requirements

- For the OracleAS Portal middle tier, you must install OracleAS Infrastructure first, and then the Portal and Wireless middle tier.
- After installing the OracleAS Infrastructure and middle tier on their respective servers, install J2EE and Web Cache middle tier on the dedicated server.

  You can also install a standalone version of OracleAS Web Cache from: `http://www.oracle.com/technology/software/products/ias/web_cache/index.html`

5.9.2 Configuring a Dedicated OracleAS Web Cache

Oracle Universal Installer automatically configures and starts OracleAS Portal middle tier and OracleAS Web Cache on the same computer during the OracleAS Portal and Wireless middle-tier installation. You need to disable the OracleAS Web Cache installed on the OracleAS Portal middle-tier computer that is not used, and configure the dedicated OracleAS Web Cache installed on a different computer to communicate with OracleAS Portal middle tier.

Configuring a dedicated OracleAS Web Cache for Web site: `www.company.com`, port number: `7777` involves the following six tasks:

- Task 1: Verify That the OracleAS Web Cache on the Dedicated Server Is Running
- Task 2: Configure OracleAS Web Cache on the Dedicated Server
- Task 3: Stop the Unused OracleAS Web Cache on the Middle-Tier Server
- Task 4: Configure OracleAS Portal Middle Tier with OracleAS Web Cache Settings
- Task 5: Configure Virtual Host Settings for Oracle HTTP Server

5.9.2.1 Task 1: Verify That the OracleAS Web Cache on the Dedicated Server Is Running

To properly configure OracleAS Web Cache on the dedicated server, OracleAS Web Cache needs to be up and running. Refer to the *Oracle Application Server Administrator's*

Guide for information about how to start, stop, restart, and view the status of OracleAS Web Cache on the Application Server Control Console home page.

5.9.2.2 Task 2: Configure OracleAS Web Cache on the Dedicated Server

You must manually configure OracleAS Web Cache on the dedicated server to properly deliver content to OracleAS Portal middle tier on a different computer. You must make appropriate changes in **Origin Servers** page from OracleAS Web Cache Manager, (**Origin Servers**, **Sites**, and **Load Balancing** > **Origin Servers**) and the **Listen Ports** page (**Ports** > **Listen Ports**).

To properly configure OracleAS Web Cache, installed on the dedicated server, you will need the origin server information from the OracleAS Web Cache installed on the same computer as OracleAS Portal middle tier.

To modify the origin server properties setting from the dedicated OracleAS Web Cache instance:

1. Make a backup copy of the `webcache.xml` file, located in the *`ORACLE_HOME`*`/webcache` directory.
2. In OracleAS Web Cache Manager on the dedicated computer, click **Origin Server** under **Origin Servers, Sites, and Load Balancing**.
3. In the **Origin Servers** page, select the Host and click **Edit**.
4. Modify the **Application Web Servers** properties for the dedicated OracleAS Web Cache using the value copied from the same page on the OracleAS Web Cache instance that was installed on the middle-tier computer. Use the online Help for guidance on changing the default Application Web Servers properties.
5. Click **OK**.
6. Click **Restart Web Cache**.

You must manually configure OracleAS Web Cache on the dedicated server to create a site definition that includes a host name `www.company.com`, and a listening port number `7777`. You must make appropriate changes in the **Site Definitions** page from OracleAS Web Cache Manager (**Origin Servers**, **Sites**, and **Load Balancing** > **Site Definitions**).

5.9.2.3 Task 3: Stop the Unused OracleAS Web Cache on the Middle-Tier Server

This task is optional. To save resources on the OracleAS Portal middle-tier server, follow the instructions in the *Oracle Application Server Administrator's Guide* to stop the unused cache on the middle-tier server. This cache instance will not be used for this deployment option.

5.9.2.4 Task 4: Configure OracleAS Portal Middle Tier with OracleAS Web Cache Settings

OracleAS Portal middle tier needs to know the OracleAS Web Cache Listen ports, the invalidator user name, invalidator password settings, and so on. You need to apply the new host name and port number of the dedicated OracleAS Web Cache to OracleAS Portal middle tier by modifying these settings in the **Portal Web Cache Settings** page:

1. From the Application Server home page on the Application Server Control Console in Oracle Enterprise Manager 10*g*, click **Portal** in the **System Components** section. The OracleAS Portal Home page appears.
2. Click the **Portal Web Cache Settings** link from the **Administration** section. The **Portal Web Cache Settings** page appears.

3. On the **Portal Web Cache Settings** page, modify the **Published Host** field with proper host name: `www.company.com`, modify the **Listening Port** field with proper port number `7777`.
4. Review the other Web Cache Settings, like Invalidation Host, to match the cache information on the dedicated server and click **Apply**. A confirmation page appears. See the online Help for guidance on changing the default ports and password settings.

See Also: Section 7.3.3, "Portal Web Cache Settings Link"

5.9.2.5 Task 5: Configure Virtual Host Settings for Oracle HTTP Server

You must create virtual host entries in the `httpd.conf` file of the Oracle HTTP Server that is part of the OracleAS Portal middle tier, with the dedicated OracleAS Web Cache settings. In this example, you will set up virtual host name `www.company.com` and port number `7777` (same as the dedicated OracleAS Web Cache Listen port). The virtual host name and port number must be consistent with the site definition value defined in OracleAS Web Cache. To do this, perform the following tasks:

1. Configure Virtual Hosts Settings, as follows:
   a. From the Application Server home page on the OracleAS Portal middle-tier server, on the Application Server Control Console, click **HTTP_Server** in the **System Components** section.

      The HTTP Server Home page is displayed.
   b. Click the **Virtual Hosts** tab.
   c. Click **Create**.
   d. On the **Introduction** page, click **Next** to create a new virtual host, using the Virtual Host Creation wizard.
   e. On the **Create Virtual Host: General** page, choose **name-based** for **Virtual Host Type**, click **Next**.
   f. On the **Create Virtual Host: Addresses** page, enter `www.company.com` in the **Server Name** field for your virtual host.
   g. Select the **Listen on all the main server IP addresses** option, and click **Next**.
   h. On the **Create Virtual Host: Ports** page, select **Listen on a specific port**, and select the Oracle HTTP Server Listen port, for example, `7778`, from the port list.
   i. Click **Next**, and click **Next** again on the **Create Virtual Host: Error Log** page.
   j. Review the summary on the **Summary** page, and then click **Finish**. A **Confirmation** page appears.
   k. When prompted to restart the Oracle HTTP Server, click **No**.
   l. Ensure that your server name, **www.company.com**, is listed in the table.
2. Configure the newly created Virtual Host, as follows:
   a. Click the **Administration** tab from the HTTP Server Home page.
   b. Click **Advanced Server Properties**.
   c. Click **httpd.conf**.
   d. Add the `Port` and the `Rewrite` directives in the VirtualHost container, as follows (shown in bold text):

```
NameVirtualHost *:7778
<VirtualHost *:7778>
      ServerName www.company.com
      Port 7777
      RewriteEngine On
      RewriteOptions inherit
</VirtualHost>
```

e. Click **Apply**.

f. When prompted to restart Oracle HTTP Server, click **Yes**.

You can verify the configuration by performing basic tests, such as navigating the Web site, or removing a portlet.

5.10 Changing the Infrastructure Services Used By a Middle Tier

Oracle Application Server 10*g* enables you to change the OracleAS Infrastructure services (either Oracle Identity Management or OracleAS Metadata Repository) that a middle tier uses. You can use this feature, for example, to move middle tiers (and their applications) from stage to production. If you are changing the OracleAS Metadata Repository that your OracleAS Portal uses, then you will also need to move application-specific data stored in the stage OracleAS Metadata Repository to an OracleAS Metadata Repository in the production environment. Changing the Infrastructure services is convenient, if you need additional computers for the production environment. In a single step, you add a computer that already has a middle tier and deployed applications. For instructions on how to change the Infrastructure Services used by a middle-tier instance, refer to the *Oracle Application Server Administrator's Guide*.

Note: By default, an OracleAS Portal middle tier is made up of one portal instance. Both the DAD name and the OracleAS Metadata Repository schema name for this instance are **portal**. You can only change the Infrastructure services of this default OracleAS Portal instance in the previously described way.

5.11 Configuring OracleAS Wireless

If Oracle Application Server Wireless is configured with OracleAS Portal during the middle-tier installation, the middle-tier installation registers the portal on the OracleAS Wireless service.

Note: If you did not configure OracleAS Wireless during installation, you can use Application Server Control Console to deploy OracleAS Wireless on the middle tier. Refer to Section 7.2.2, "Using Application Server Control Console to Configure OracleAS Portal" for similar steps used to configure OracleAS Portal.

In case of multiple middle-tier installations, the first configured OracleAS Wireless service URL is stored in the OracleAS Portal instance. You can change this to your choice of OracleAS Wireless service by running the `cfgiasw.pl` script. Refer to Section C.8, "Using the cfgiasw Script to Configure Mobile Settings" for more information.

Note: You can also change the URL to your choice of OracleAS Wireless service by running the `portalRegistrar` script in the Oracle Application Server middle tier selected for the OracleAS Wireless service. Refer to the *Oracle Application Server Wireless Administrator's Guide* for more information about configuring OracleAS Wireless.

5.12 Changing the OracleAS Portal Schema Password

This section provides information about changing schema passwords for both default and nondefault OracleAS Portals.

Changing the Schema Password for a Default OracleAS Portal Instance

For information about changing the OracleAS Portal schema password for the default OracleAS Portal instance, refer to the section on changing OracleAS Metadata Repository schema passwords, in the *Oracle Application Server Administrator's Guide*.

Note: By default, an OracleAS Portal middle tier is made up of one portal instance. Both the DAD name and the OracleAS Metadata Repository schema name for this instance are **portal**. The section on changing OracleAS Metadata Repository schema passwords in the *Oracle Application Server Administrator's Guide* describes how to change the schema password for this default OracleAS Portal instance.

Changing the Schema Password for a Nondefault OracleAS Portal Instance

Refer to Section B.1.1, "Changing the OracleAS Portal Schema Password" to change the OracleAS Portal schema password for a nondefault OracleAS Portal instance.

6

Securing OracleAS Portal

One of the most important aspects of any portal solution is security. The ability to control user access to Web content and to protect your site against people breaking into your system is critical. This chapter describes the architecture of OracleAS Portal security in the following topics:

- About OracleAS Portal Security
- Configuring OracleAS Security Framework for OracleAS Portal
- Configuring OracleAS Portal Security

See Also:

- *Oracle Application Server Security Guide*
- *Oracle Identity Management Concepts and Deployment Planning Guide*
- *Oracle Application Server Single Sign-On Administrator's Guide*

6.1 About OracleAS Portal Security

The sections that follow provide an overview of OracleAS Portal security and how it works with the OracleAS Security Framework.

- OracleAS Portal Security Model
- Classes of Users and Their Privileges
- Resources Protected
- Authorization and Access Enforcement
- Enforcing Role-Based Access Control
- Leveraging Oracle Application Server Security Services
- Leveraging Oracle Identity Management Infrastructure
- Security for Portlets
- Securing the OmniPortlet and Simple Parameter Form
- Securing the Web Clipping Provider
- Securing the Federated Portal Adapter
- Securing OraDAV

6.1.1 OracleAS Portal Security Model

When you make content available on the Web, it is very likely that you need to restrict access to at least some parts of it. For example, it is unlikely that you want every user to be able to see every document on your site. It is even less likely that you want every user to be able to change every document on your site. OracleAS Portal provides a comprehensive security model that enables you to completely control what users can see and change on your Web site.

Before a user logs on to OracleAS Portal, they can only view the content that the content contributors designate as public. Public content can be viewed by any user who knows the URL of a portal object (for example, a page) and can connect to the computer where it is stored. The user sees only those aspects of the object that are designated as public, such as the public portlets. If the object has no public contents, then the user is denied access to it.

Once the user logs in to the portal, they may or may not be able to see and change content depending upon their access privileges. Typically, an authenticated user can see and do more in the portal than a public user. For example, an authenticated user might see items or portlets on the page that the public user cannot view. An authenticated user might also be able to add and edit content, and change properties, privileges that would typically be denied to a public user. In the portal, you can control access to objects (pages, items, or portlets) by user and group. That is, you might grant access privileges for a page to specific named users, user groups, or a combination of both.

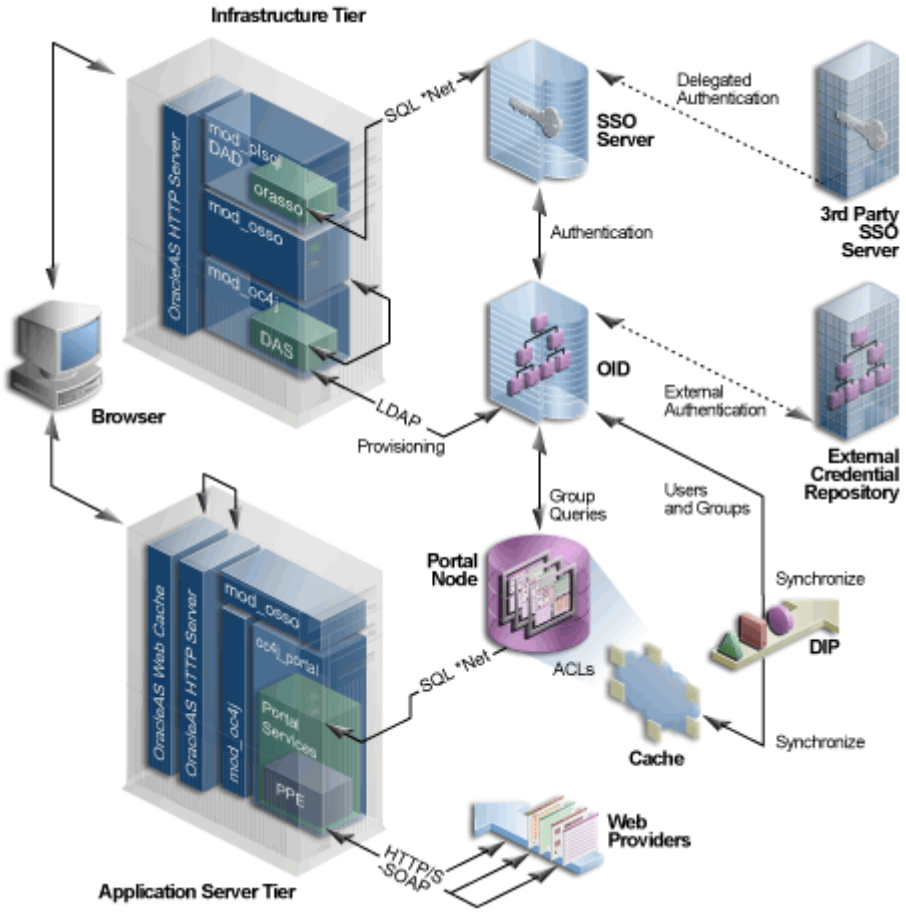
To support this flexible approach to controlling access to Web content, OracleAS Portal leverages the other components of Oracle Application Server and Oracle Database 10*g* to provide strong protection for your portal. OracleAS Portal interacts with all of the following components to implement its security model:

- Oracle Application Server Single Sign-On authenticates users, who are attempting to gain access to non-public areas of your portal.
- mod_osso is an Oracle HTTP Server module that redirects authentication requests to OracleAS Single Sign-On. It also keeps track of user activity in partner applications.
- OracleAS Web Cache is the cache used to serve up pages generated by OracleAS Portal (or proxied to the Oracle HTTP Server if not able to service the request). Based on invalidation caching, OracleAS Portal invalidates the cache when the underlying page or metadata changes.
- Oracle Internet Directory, Oracle's native LDAP version 3 service, acts as the repository for user credentials and group memberships.
- The Oracle Internet Directory's Oracle Delegated Administration Services adds or updates the information stored inside the directory (users and groups).
- Oracle Directory Integration Platform notifies OracleAS Portal upon the occurrence of any directory events (for example, user deletions) to which OracleAS Portal subscribes. In essence, the directory integration server informs OracleAS Portal when a change occurs in the directory that requires a change in OracleAS Portal.

OracleAS Portal Architecture

Figure 6–1 shows the components and relationships of the OracleAS Portal security architecture.

Figure 6–1 OracleAS Portal Security Architecture



The OracleAS Portal architecture consists of three basic tiers, including the client browser, the middle-tier server, and the infrastructure servers and repositories. By default, Oracle Internet Directory and OracleAS Single Sign-On are installed on the same host as part of the infrastructure installation. This tier is subsequently used for the OracleAS Portal installation.

While the default installation has all three servers and repositories installed on the same host, we recommend that you install these functions on separate servers.

In OracleAS Portal, the middle and infrastructure tier components have a number of components in common. These include a repository access component made up of a Database Access Descriptor (DAD) and Oracle Containers for J2EE (OC4J). The latter is used on the infrastructure tier to run Oracle Delegated Administration Services and to execute portal runtime engine on the middle tier.

To optimize the throughput and performance of OracleAS Portal, generated pages are cached in OracleAS Web Cache. If a request for a portal page can be served from OracleAS Web Cache, it will be returned without accessing the OracleAS Portal middle tier. If not, the request will be forwarded to the origin HTTP server and the Parallel Page Engine.

If the current user is not authenticated with the Single Sign-On environment, and if the requested page is not a public page, the user is prompted for a user name and password. This function is carried out through a redirection to OracleAS Single Sign-On for authentication, which in turn verifies the credentials against Oracle

Internet Directory through an LDAP request. The credentials are compared to those found in the directory.

Upon successful authentication, OracleAS Single Sign-On creates a single sign-on session cookie. Once the user is authenticated and an appropriate OracleAS Portal session created, it is necessary to determine which pages and objects the user has the necessary access privileges to. For performance reasons the access control lists (ACLs) for all portal objects are stored in the OracleAS Portal schema in the OracleAS Metadata Repository along with the definition of the objects being secured.

User and Group provisioning is a function of Oracle Internet Directory. That is, all user and group membership information is stored in the Oracle Internet Directory. When a user first logs in to OracleAS Portal, their current group membership is read from the directory and cached in the same repository as the ACLs. This process allows for fast lookup of object privileges. Once the object and page privileges of the user are known, the appropriate page metadata can be generated to allow the Parallel Page Engine to assemble the secured page.

To simplify the provisioning of users and groups in Oracle Internet Directory for use in the portal, OracleAS Portal uses Oracle Delegated Administration Services to generate a user interface to allow direct access to Oracle Internet Directory. Calls to Oracle Delegated Administration Services are protected by the mod_osso plug-in, which verifies that the user has been properly authenticated before providing access to the Oracle Internet Directory.

One important feature of the security architecture is the ability to keep the local cached group membership list synchronized with Oracle Internet Directory. The Oracle Directory Integration Platform automatically keeps the locally cached information up-to-date with changes in Oracle Internet Directory.

If you need to authenticate against an external repository, Oracle Internet Directory supports both delegated and external authentication. Likewise, just as the Oracle Directory Integration Platform keeps the local cache synchronized with the Oracle Internet Directory, it also keeps the Oracle Internet Directory synchronized with any external repository.

6.1.2 Classes of Users and Their Privileges

OracleAS Portal provides a number of user accounts and groups by default.

- OracleAS Portal Default, Seeded User Accounts
- OracleAS Portal Default, Seeded Groups
- OracleAS Portal Default Schemas

6.1.2.1 OracleAS Portal Default, Seeded User Accounts

Table 6–1 describes the user accounts created by default when OracleAS Portal is installed.

Table 6–1 Default OracleAS Portal Users

User	Description
PUBLIC	Is the user account that identifies unauthenticated access to the OracleAS Portal. Once a user logs in, the user name changes from PUBLIC to the user name by which the user is authenticated. When granting portal privileges on individual objects that do not have an explicit check box for granting the object to Public, this user can be identified as the grantee of the privilege to grant access to it for unauthenticated users.

Table 6–1 (Cont.) Default OracleAS Portal Users

User	Description
PORTAL	Is the super-user for the portal. In a standard installation, the user name is PORTAL. This user account has the highest privileges because it is granted all the global privileges available in the portal. The initial password for PORTAL is the password that is supplied during the Oracle Application Server installation. This user is also an Oracle Instant Portal administrator for every Oracle Instant Portal, regardless of who created them.
ORCLADMIN	Similar to PORTAL, this account is granted the highest privileges in OracleAS Portal. This account is created for the Oracle Application Server administrators, and uses the password that is supplied during the Oracle Application Server installation. This user is also an Oracle Instant Portal administrator for every Oracle Instant Portal, regardless of who created them.
PORTAL_ADMIN	Is a privileged OracleAS Portal user account with administrative privileges excluding those that would give the user the ability to obtain higher privileges or perform any database operations. This user cannot edit any group or manage privileges on any schema or shared object. This account is typically intended for an administrator who manages pages and provisions user accounts. The initial password for PORTAL_ADMIN is the password that is supplied during the Oracle Application Server installation.

6.1.2.2 OracleAS Portal Default, Seeded Groups

Table 6–2 describes the groups created by default when OracleAS Portal is installed.

Table 6–2 Default OracleAS Portal Groups

Group[1]	Description
AUTHENTICATED_USERS	Is the group that includes any authenticated, or logged in, user. The purpose of this group is to provide a means to assign the default privileges you want every logged in user to have in the portal. By default, this group is given the Create Group and Create All Styles privileges. This group is a member of OracleDASCreateGroup.

Table 6–2 (Cont.) Default OracleAS Portal Groups

Group[1]	Description
DBA	Is a highly privileged group established for Oracle Application Server administrators. Components that are part of Oracle Application Server grant full component-specific privileges to members of this group. The DBA group is a member of the PORTAL_ ADMINISTRATORS group. This group is also a member of the following Oracle Application Server privilege groups: ■ OracleDASCreateUser ■ OracleDASEditUser ■ OracleDASDeleteUser ■ OracleDASUserPriv ■ OracleDASCreateGroup ■ OracleDASEditGroup ■ OracleDASDeleteGroup ■ OracleDASGroupPriv ■ OracleDASConfiguration Members of DBA do not have the necessary privileges to administer OracleAS Single Sign-On. If you want members of this group to administer OracleAS Single Sign-On, then you must grant them those privileges as described in the *Oracle Application Server Single Sign-On Administrator's Guide*.

Table 6–2 (Cont.) Default OracleAS Portal Groups

Group[1]	Description
PORTAL_ADMINISTRATORS	Is a highly privileged group established for OracleAS Portal. By default, this group is given the following OracleAS Portal global privileges: ■ Manage All Page Groups ■ Manage All Pages ■ Manage All Styles ■ Manage All Providers ■ Manage All Portlets ■ Manage All Portal DB Providers ■ Manage All Portal User Profiles ■ Edit All Group Profiles ■ Manage All Logs ■ Execute All Transport Sets This group is a member of the following Oracle Application Server privilege groups: ■ OracleDASCreateUser ■ OracleDASEditUser ■ OracleDASDeleteUser ■ OracleDASCreateGroup ■ OracleDASConfiguration Members of PORTAL_ADMINISTRATORS do not have the necessary privileges to administer OracleAS Single Sign-On. If you want members of this group to administer OracleAS Single Sign-On, then you must grant them those privileges as described in the *Oracle Application Server Single Sign-On Administrator's Guide*.
PORTLET_PUBLISHERS	Is a privileged group established for users who need to publish portlets to other users of the portal. By default, this group is given the Publish All Portlets global privilege of OracleAS Portal.
PORTAL_DEVELOPERS	Is a privileged group established for users who are building portlets. By default, this group is given the following OracleAS Portal global privileges: ■ Create All Portal DB Providers ■ Manage All Shared Components If you want PORTAL_DEVELOPERS to create database providers and portlets, you need to give this group privileges that enable them to modify schema, for example, Modify Data on all schemas. For more information, refer to Table 6–6.
RW_ADMINISTRATOR	Is the group of users who administer OracleAS Reports Services reports, printers, calendars, and servers. You must assign this group any desired object privileges (for example, Manage).

Table 6–2 (Cont.) Default OracleAS Portal Groups

Group[1]	Description
RW_DEVELOPER	Is the group of users who develop OracleAS Reports Services reports. You must assign this group any desired object privileges (for example, Execute or Manage).
RW_POWER_USER	Is the group of users who can modify OracleAS Reports Services reports. You must assign this group any desired object privileges (for example, Execute or Manage).
RW_BASIC_USER	Is the group of users who use OracleAS Reports Services reports. You must assign this group any desired object privileges (for example, Execute).
OIP_USER_ADMINS	Is the group of users who can both create Oracle Instant Portals and perform user administration on them. Both the PORTAL and ORCLADMIN users are members of this group.
OIP_AVAILABLE_USERS	Is the group of users who can access Oracle Instant Portals. This list appears in the Manage User Rights dialog box in the Oracle Instant Portal user interface. **Note:** To designate OracleAS Portal users as Oracle Instant Portal users, use the **Groups** portlet to add them to the OIP_AVAILABLE_USERS group, which was created by default during the installation process. Then use the **Manage User Rights** dialog in Oracle Instant Portal to grant the appropriate privileges.

[1] All groups shown in this table are located in cn=<portal_group_container>,cn=Groups,dc=MyCompany,dc=com. Note that identity management realm name is determined by the domain name of the server on which the system is installed. For example, if the domain name of the server was oracle.com, the default identity management realm name would be dc=oracle,dc=com. If the domain name of the server could not be determined, Oracle Internet Directory defaults to the domain specified during installation by the administrator. The OracleDASxxxxx groups are Oracle Internet Directory privilege groups that reside under cn=groups,cn=OracleContext,dc=MyCompany,dc=com. These groups provide the privileges to perform operations in Oracle Internet Directory, such as creating or editing of users and groups, and their privileges.

Notes:

- When viewing portal-related roles in Oracle Internet Directory Self-Service Console, the descriptions for these roles are prefixed with numbers, for example, `portal.040823.142021.462000000`. The numbers are actually the name of the OracleAS Portal application, and are displayed to enable selection of roles in a multiple-portal environment where multiple portals associated with the same Identity Management system exist.
- When a user is granted Manage privileges on any Oracle Instant Portal's home page, he or she is granted full privileges over that particular portal. The user cannot edit, delete, or even view any other Oracle Instant Portal, unless he or she has been granted explicit permission to do so. However, the user can do the following:

  — Create new users.

  — Delete any user in the **Manage User Rights** dialog box, even those he or she did not create. (For this reason, it's wise to curtail the number of users who have Manage privileges on a home page.)

  — Create new Oracle Instant Portals.

  Refer to the *Oracle Instant Portal Getting Started* guide for more information.

6.1.2.3 OracleAS Portal Default Schemas

When you install OracleAS Portal, the installation process installs some default schemas of which you need to be aware.

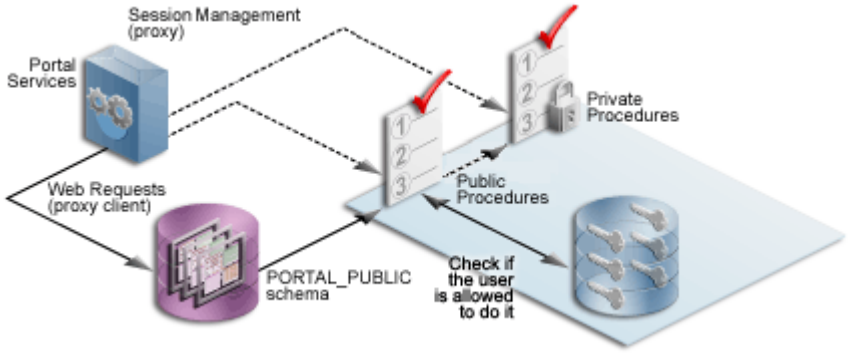
Table 6–3 describes the schemas created by default when OracleAS Portal is installed.

Table 6–3 Default OracleAS Portal Schemas

Schema	Description
PORTAL	Contains the OracleAS Portal database objects and code. To execute Web requested procedures, Portal Services uses N-Tier authentication to connect to the schema to which the lightweight user accounts are assigned (by default, PORTAL_PUBLIC). As shown in Figure 6–2, access to the database of the portal user is proxied through the single schema user. The default name for this schema in a standard OracleAS Portal installation is PORTAL. If you want to give it another name, you must perform a custom installation.
PORTAL_PUBLIC	Is the schema that all lightweight users are mapped to by default. All procedures publicly accessible through the Web are granted execute to PUBLIC, which makes them accessible through this schema. In a standard OracleAS Portal installation, this schema is named PORTAL_PUBLIC. If you want to give it another name, you must perform a custom installation.
PORTAL_DEMO	Is created to hold some demonstration code. The installation of this schema is optional.
PORTAL_APP	Is used for external JSP application authentication.

Figure 6–2 shows the N-Tier authentication by user proxy.

Figure 6–2 N-Tier Authentication By User Proxy



6.1.3 Resources Protected

Within OracleAS Portal, you decide at what level of granularity you want to control access. You can assign privileges to any object for each user or for each group. For example, you can assign access privileges for each user for each and every item in your portal, but this approach creates considerable overhead for your content contributors.

If you want to lessen the burden on contributors, then you can assign privileges for each group at the page level and simply ensure that all of the items that you place on any given page have similar security requirements. With this approach, the security that items receive through the page that contains them is usually sufficient and content contributors only need to assign privileges for items that require higher security than the page.

> **See Also:** Section 6.1.6.9, "Oracle Delegated Administration Services Public Roles" for information about how you might model privileges.

6.1.3.1 Global Privileges

Use global privileges to give a user or group a certain level of privileges on all objects of a particular type.

> **Note:** Global privileges confer a great deal of power on the user to whom they are granted. As a result, they should be granted very cautiously and only to users or groups who truly require them. You should only have a small number of users with global privileges.

There are three types of privilege groups:

- Table 6–4, " Page Group Privileges"
- Table 6–5, " Portal DB Provider Privileges"
- Table 6–6, " Administration Privileges"

Table 6–4 Page Group Privileges

Object Type	Privileges
All Page Groups	**None:** No global page group privileges are granted. **Manage All:** Perform any task on any page group. This privilege supersedes any other privilege in the other global page group privileges. For example, this also allows managing of any page. **Manage Classifications:** Create, edit, and delete any category, perspective, custom attribute, custom page type, or custom item type in any page group. **Manage Templates:** Create, edit, and delete any Portal Template or HTML template in any page group. Grant access to any template. **Manage Styles:** Create, edit, and delete any style in any page group. **View:** View any page in any page group. **Create:** Create page groups, and create any page group object in those page groups. Users or groups with these privileges can also edit and delete the page groups and page group objects they create. **Note:** These users cannot create any objects in the existing page groups.

Table 6–4 (Cont.) Page Group Privileges

Object Type	Privileges
All Pages	**None:** No global page privileges are granted.
	Manage: Create, edit, personalize, or delete any page in any page group. Grant access to any page in any page group.
	Manage Content: Add, edit, hide, show, share, or delete any item, portlet, or tab on any page in any page group.
	Manage Items With Approval: Create new items on any page in any page group. These items are not published until approved via a specified approval process. Users and groups with this privilege can also edit the items they create. Users and groups with this privilege can personalize pages. When approvals are not enabled for the page group, this privilege becomes equivalent to the global privilege **Manage Content** on the object type **All Pages** with regard to items.
	Manage Styles: Apply an available or new style to any page in any page group. Create, edit, and delete new styles. **Note:** Only allows editing of styles created by user (cannot modify or delete other user's styles).
	Personalize Portlets (Full): Personalize any page in any page group to add, show, hide, delete, move, or rearrange portlets. Personalize any page to show, hide, delete, or rearrange tabs, or add tabs to existing tabbed regions. Personalize any page in any page group to use a different style.
	Personalize Portlets (Add-only): Personalize any page in any page group to add portlets or add tabs to existing tabbed regions. Users or groups with these privileges can also delete the portlets they add. Personalize any page in any page group to use a different style.
	Personalize Portlets (Hide-Show): Personalize any page in any page group to show or hide portlets or tabs. Personalize any page in any page group to use a different style. Arrange portlets in any page in any page group.
	Personalize (Style): Personalize any page in any page group to use a different style.
	View: View any page in any page group.
	Create: Create subpages in any page group. Users or groups with these privileges can also edit and delete the subpages they create. **Note:** You must have Manage privileges on the root page in a page group in which you want to create the pages.
All Styles	**None:** No global style privileges are granted.
	Manage: Create, edit, and delete any style in any page group.
	View: View any style in any page group.
	Publish: Make any style in any page group public for other users to use.
	Create: Create styles in any page group. Users or groups with these privileges can also edit and delete the styles they create.

Table 6–4 (Cont.) Page Group Privileges

Object Type	Privileges
All Providers	**None:** No global provider privileges are granted.
	Manage: Register, edit, deregister any provider, and display and refresh the Portlet Repository. Also allowed to grant edit abilities on any provider.
	Edit: Edit any registered provider.
	Publish: Register and deregister any provider.
	Execute: View the contents of any provider.
	Create: Register portlet providers. On the provider the user (or group) creates, the user gets a Manage privilege; therefore, the user can perform all operations (including edit and deregister) on the particular provider that the user has created.
All Portlets	**None:** No global portlet privileges are granted.
	Manage: Create, edit, or delete any portlet in any provider.
	Edit: Edit any portlet in any provider.
	Execute: Execute any portlet in any provider. Users or groups with these privileges can see all portlets even if the portlet security is enforced. The **Show** link appears in the Navigator for all portlets.
	Access: View any portlet in any provider.
	Publish: Publish any page, navigation page, or Portal DB Provider portlet to the portal, making it available for adding to pages.

Table 6–5 Portal DB Provider Privileges

Object Type	Privileges
All Portal DB Providers	**None:** No global application privileges are granted.
	Manage: Edit or delete any Portal DB Provider. Create, edit, or delete any portlet in any Portal DB Provider. Grant access to any Portal DB Provider and any portlet in any Portal DB Provider.
	Edit Contents: Edit any portlet in any Portal DB Provider.
	View Source: View the package specification and body and run any portlet in any Portal DB Provider. Intended primarily for users or groups who may want to look at a portlet's source so they know how to call it.
	Personalize: Run and personalize any portlet in any Portal DB Provider.
	Run: Run any portlet in any Portal DB Provider.
	Create: Create Portal DB Providers. Users or groups with these privileges can edit, and delete the providers they create and create, edit, and delete any portlet in them.

Table 6–5 (Cont.) Portal DB Provider Privileges

Object Type	Privileges
All Shared Components	**None:** No global shared component privileges are granted. **Manage:** Create, view, copy, edit, delete, and export any shared component in any Portal DB Provider. View and copy any system shared component. Grant access to any non-system shared component. **Create:** Create shared components in any Portal DB Provider. View and copy any system shared component. View any shared component. Users and groups with these privileges can view, copy, edit, delete, and export the shared components they create.

Table 6–6 Administration Privileges

Object Type	Privileges
All User Profiles	**None:** No global user profile privileges are granted. **Manage:** Edit any user profile. Grant this privilege to other users and groups. **Edit:** Edit any user profile.
All Group Privileges (profiles)	**None:** No global group profile privileges are granted. **Manage:** Edit any group profile. Grant this privilege to other groups. The Privileges tab of the group profile allows the user to assign those privileges to the group. The Manage privilege provides the edit privilege and the ability to grant it to others. **Edit:** Edit any portal group profile (setting the default home page and default mobile home page). **Note:** The ability to change any group's description, memberships, and owners is controlled by the Oracle Internet Directory access control policies, which are administered through membership in the OracleDASEditGroup group.

Table 6–6 (Cont.) Administration Privileges

Object Type	Privileges
All Schemas	**None:** No global schema privileges are granted.
	Manage: Create, edit, and drop any schema. Grant access to any schema. Create, edit, drop, and rename any database object in any schema. Query, update, delete, and insert data in any table or view in any schema. Compile any function, procedure, package, or view in any schema. Execute any function, procedure, or package in any schema. Grant access to any database object in any schema.
	Modify Data: Create schemas. Query, update, delete, and insert data in any table or view in any schema. Compile any function, procedure, package, or view in any schema. Execute any function, procedure, or package in any schema. Users or groups with these privileges can edit, drop, and grant access to the schemas they create.
	Insert Data: Create schemas. Query and insert data in any table or view in any schema. Users or groups with these privileges can edit, drop, and grant access to the schemas they create.
	View Data: Create schemas. Query data in any table or view in any schema. Users or groups with these privileges can edit, drop, and grant access to the schemas they create.
	Create: Create schemas. Users with these privileges can also edit, drop, and grant access to the schemas they create. **Note:** If you want a user or group to access the **Schemas** portlet on the **Administer Database** tab of the **Builder** page, either make the user or group a member of the DBA group, or explicitly grant the user or group View privileges on the **Administer Database** tab. If you do not grant these privileges, the user or group will still be able to use the Navigator to access schemas.
All Logs	**None:** No global log privileges are granted.
	Manage: Edit or purge any log. Grant this privilege to others.
	Edit: Edit or purge any log.
	View: View any log.
All Transport Sets	**None:** No global transport set privileges are granted.
	Execute: Export and Import objects that are not shared. In addition, users with these privileges can edit or purge Export and Import objects that are not shared.
	Manage: Edit or purge any import or export sets. Grant this privilege to others.

6.1.3.2 Object Privileges

You can assign access privileges to users or groups for all of the following objects within OracleAS Portal through the **Access** tab of the object's Edit Page:

Table 6–7 OracleAS Portal Objects with Privilege Control

Type of Object	Available Privileges	Inherited Privileges
Calendar	■ Manage ■ View ■ Personalize ■ Execute	From Database Provider

Table 6–7 (Cont.) OracleAS Portal Objects with Privilege Control

Type of Object	Available Privileges	Inherited Privileges
Chart (based on SQL query)	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
Chart (based on wizard)	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
Data Component	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
Data Component Cell	■ Edit ■ View	From Data Component
Database Provider	■ Manage ■ Edit ■ View Source ■ Personalize ■ Execute	Not applicable
Document	■ Own ■ Manage ■ View Only	From page or item
Dynamic Page Component	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
Form[1]	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
Frame Driver	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider

Table 6–7 (Cont.) OracleAS Portal Objects with Privilege Control

Type of Object	Available Privileges	Inherited Privileges
Hierarchy	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
Image Chart	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
Link	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
List of Values	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
Menu	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
OracleAS Reports Services printer	■ Manage ■ Edit ■ View ■ Execute	From Database Provider
OracleAS Reports Services report	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
OracleAS Reports Services Server	■ Manage ■ Edit ■ View ■ Execute	From Database Provider

Table 6–7 (Cont.) OracleAS Portal Objects with Privilege Control

Type of Object	Available Privileges	Inherited Privileges
Page	■ Manage ■ Manage Content ■ Manage Items With Approval[2] ■ Manage Style ■ Personalize Portlets (Full) ■ Personalize Portlets (Add-Only) ■ Personalize Portlets (Hide-Show) ■ Personalize (Style) ■ View	From the root page of the page group
Page group	■ Manage All ■ Manage Classifications ■ Manage Templates ■ Manage Styles ■ View	Not applicable
Page Item	■ Own ■ Manage ■ View Only	From page
Portlet	■ Manage ■ Edit ■ Execute ■ Access ■ Publish	Not applicable
Provider	■ Manage ■ Edit ■ Publish ■ Execute	Not applicable
Query by example form	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
Report[3]	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider

Table 6–7 (Cont.) OracleAS Portal Objects with Privilege Control

Type of Object	Available Privileges	Inherited Privileges
Schema	■ Manage ■ Modify ■ Insert ■ View	Not applicable
URL	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider
XML	■ Manage ■ Edit ■ View ■ Personalize ■ Execute	From Database Provider

[1] You can have many different types of forms (stored procedure or table based, version 2 or version 3 based, and master-detail), but all of these types have the same available privileges and privilege inheritance.

[2] This privilege is only available on the Access tab if approvals are enabled at the page group level. When approvals are not enabled for the page group, or when approvals are enabled, but there is no approval process defined at the page or page group level, this privilege becomes equivalent to the global privilege Manage Content on the page.

[3] You can have two different types of reports (SQL and table based), but all of these types have the same available privileges and privilege inheritance.

6.1.3.3 Granting Privileges to New Providers

When you create or register a new provider, a page is created in the Portlet Repository under Portlet Staging Area to display portlets for that provider. This page is not visible to all logged in users. It is only visible to the user who published the provider and portal administrators. The publisher or portal administrator can change the provider page properties to grant privileges to appropriate users or groups, as required.

6.1.3.4 Privileges to Create and Edit Web Providers and Provider Groups

To create and manage Web providers and provider groups through the user interface, as opposed to working with files directly, you need to grant appropriate privileges to the administrative users. The access control list is implemented differently than for the OracleAS Portal schema resident objects. Refer to Section 6.1.3.1, "Global Privileges" and Section 6.1.3.2, "Object Privileges" for information about the OracleAS Portal schema resident objects. Rather, the grants for provider privileges are maintained in an XML file.

> **Note:** The privileges described here are for users developing new Web providers and pertain to authorizations that are enforced by the provider user interface. These privileges are not required to register Web providers.

To grant privileges to create, edit, and delete Web providers or provider groups, you need to manually make changes to the following file:

MID_TIER_ORACLE_HOME/j2ee/OC4J_
Portal/applications/portalTools/providerBuilder/WEB-INF/deployment_
providerui/provideruiacls.xml

An example of this file follows:

> **Note:** In this example, the user names any_provider_manage_user, any_provider_edit_user, and so on, are just sample user names used here to illustrate the privilege codes that correspond to the privileges implied by the corresponding user names. An actual user grant would have the OracleAS Single Sign-On user name as the value of the name attribute in the <user> element, and the privilege would be populated with the appropriate privilege code.

```
<providerui xmlns="http://www.oracle.com/portal/providerui/1.0">
    <objectType name="ALL_OBJECTS">
        <object name="ANY_PROVIDER" owner="providerui">
           <user name="any_provider_manager_user" privilege="500"/>
           <user name="any_provider_edit_user" privilege="400"/>
           <user name="any_provider_execute_user" privilege="300"/>
           <user name="any_provider_create_user" privilege="100"/>
        </object>
        <object name="ANY_PORTLET" owner="providerui">
           <user name="any_portlet_manage_user" privilege="500"/>
           <user name="any_portlet_edit_user" privilege="400"/>
           <user name="any_portlet_execute_user" privilege="300"/>
        </object>
   </objectType>
   <objectType name="PROVIDER">
        <object name="TEST_PROVIDER" owner="providerui">
           <user name="provider_manage_user" privilege="500"/>
           <user name="provider_edit_user" privilege="400"/>
           <user name="provider_execute_user" privilege="300"/>
      </object>
   </objectType>
   <objectType name="PORTLET">
        <object name="PORTLET_UNDER_TEST_PROVIDER" owner="TESTPROVIDER">
           <user name="portlet_manage_user" privilege="500"/>
           <user name="portlet_edit_user" privilege="400"/>
           <user name="portlet_execute_user" privilege="300"/>
        </object>
   </objectType>
</providerui>
```

This file allows for granting of the following types of privileges, described in the following sections:

- Global Privileges
- Object Level Privileges

6.1.3.4.1 Global Privileges Table 6–8 describes the global object types and corresponding privilege codes that can be granted to users in the provideruiacls.xml file. When granting a privilege to the user, you should specify the numeric privilege code.

Table 6–8 Global Privilege Codes for provideruiacls.xml

Type of Object	Available Privileges
ANY_PROVIDER	**500 (Manage):** Can create, edit, delete, and open any provider or provider group and portlets under them.
	400 (Edit): Can create and edit any provider or provider group and execute the portlets under them.
	300 (Execute): Can open any provider or provider group and execute the portlets under them.
	100 (Create): Can create any provider or provider group.
ANY_PORTLET	**500 (Manage):** Can edit, delete, and execute any portlet under any provider.
	400 (Edit): Can edit and execute any portlet under any provider.
	300 (Execute): Can execute any portlet under any provider.

To add a privilege to a particular user, add an entry in the proper object type container, for example:

```
<objectType name="ALL_OBJECTS">
    <object name="ANY_PROVIDER" owner="providerui">
       <user name="jdoe" privilege="400"/>
        …
    </object>
</objectType>
```

For these global privileges, the `objectType` name is set to ALL_OBJECTS, the object owner is set to `providerui`, and the object name should be `ANY_PROVIDER` or `ANY_PORTLET` depending on the type of grant you are setting.

You then set the user name and privilege to the values corresponding to the OracleAS Single Sign-On user name of the grantee and the privilege code you wish to assign. This model does not support any grants to groups. It only supports grants directly to users.

6.1.3.4.2 Object Level Privileges

Table 6–9 describes the object level privileges that can be granted to users to give them privileges on specific object instances as referenced within the `provideruiacl.xml` XML file.

Table 6–9 Object Privilege Codes for provideruiacl.xml

Type of Object	Available Privileges
PROVIDER	**500 (Manage):** Can edit, delete, and open the specified provider or provider group and the portlets under it.
	400 (Edit): Can edit the specified provider or provider group and execute the portlets under it.
	300 (Execute): Can open the specified provider or provider group and execute the portlets under it.
PORTLET	**500 (Manage):** can edit, delete, and execute the specified portlet under the specified provider.
	400 (Edit): Can edit and execute the specified portlet under the specified provider.
	300 (Execute): Can execute the specified portlet under the specified provider.

To add a privilege to a particular user, add an entry into the proper object type container, for example:

```
<objectType name="PORTLET">
    <object name="PORTLET_UNDER_TEST_PROVIDER" owner="TESTPROVIDER">
       <user name="jdoe" privilege="400"/>
        …
     </object>
</objectType>
```

For the object level privileges, the `objectType` name is set to `PROVIDER` or `PORTLET`, depending upon to which object instances you are providing access. The object name is set to the provider name or the portlet name, respectively. The object owner is set to `providerui` or the name of the associated provider, again respectively for providers and portlets.

Table 6–10 summarizes these rules:

Table 6–10 Attribute Values for Providers and Portlets

Attribute	Provider Instance Grant	Portlet Instance Grant
ObjectType name	`PROVIDER`	`PORTLET`
Object name	Provider or provider group name	Portlet name
Object owner	`providerui`	Provider name
User name	OracleAS Single Sign-On user name	OracleAS Single Sign-On user name
User privilege	Privilege code	Privilege code

6.1.3.5 Privileges to Create and Edit WSRP Producers

For information on granting privileges to create and edit WSRP producers, refer to the Security section of the JSR 168 specification, at `http://jcp.org/aboutJava/communityprocess/first/jsr168/index.html`.

6.1.3.6 Privileges to Create and Edit URL and XML Portlets in the Portlet Repository

To create and edit URL and XML portlets in the Portlet Repository, privileges need to be granted to the users. The URL and XML portlets are available from the Portlet Builders page in the Portlet Repository. To grant access, or to edit sample providers in the JPDK Web application, you need to manually make changes to following file:

MID_TIER_ORACLE_HOME`/j2ee/OC4J_Portal/applications/jpdk/jpdk/WEB-INF/deployment_providerui/provideruiacls.xml`

Refer to Section 6.1.3.4, "Privileges to Create and Edit Web Providers and Provider Groups" for information about the privileges you can grant.

6.1.4 Authorization and Access Enforcement

When users attempt to log in to OracleAS Portal, OracleAS Single Sign-On must first verify their credentials against the directory. Once their identity has been verified, OracleAS Portal checks their access privileges in the directory to determine which objects they may see and use within the portal.

1. From OracleAS Portal, the user requests to log in by clicking the **Login** link.
2. The login request is forwarded to OracleAS Single Sign-On for authentication.
3. OracleAS Single Sign-On verifies the user credentials against the information stored in the directory.
4. If authentication is successful, then OracleAS Single Sign-On creates an SSO cookie for the user. If authentication is not successful, the user is denied access and returned to the login page to re-enter their user name and password.
5. Once the user's identity has been verified, control is returned to OracleAS Portal, which creates a portal session cookie. OracleAS Portal then connects to the directory and determines the user's group memberships and privileges.
6. OracleAS Portal caches the user's membership and privilege information locally for the duration of their session.
7. When the user attempts to access a page, OracleAS Portal performs the following checks:
   - Checks whether the page is public. If so, the user can view it.
   - If the page is not public, OracleAS Portal checks the local privilege table to determine whether the current user has privileges to view the page. If the user has viewing privileges, the user can view it.
   - If the current user does not have direct viewing privileges on the page, OracleAS Portal checks the cached membership information and privilege table to determine whether any of the groups to which the user belongs has privileges to view the page. If one of the groups to which the user belongs has viewing privileges on the page, the user can view it.

Note: If changes are made to Oracle Internet Directory that affect the user's privileges, a notification is raised and the cached information about the user is invalidated. Thus, OracleAS Portal starts enforcing the user's updated privileges as soon as it receives the notification. If you are using groups that are based on the `groupOfNames` objectclass, then you need to update the provisioning profile. See "Update Subscription Profiles for Groups Based on groupOfNames" for more details.

6.1.5 Leveraging Oracle Application Server Security Services

OracleAS Portal leverages Oracle Application Server Security Services in the following ways:

- SSL Encryption
- JAZN
- J2EE Security

See Also: For more information:

- Section 6.3.2.1, "Configuring SSL for OracleAS Portal"
- Section F.2, "Setting Up a JAZN File for External Communication"
- *Oracle Containers for J2EE Services Guide*

SSL Encryption

The use of HTTPS and the Secure Sockets Layer (SSL) allows for the creation of a secured connection between a client and a server. Digital certificates on each end of the communication verify the validity of the server and encryption of the communication to ensure that it is not compromised. You can implement SSL encryption for OracleAS Portal through the Oracle Application Server Security Services.

JAZN

JAZN is the internal name for a Java Authentication and Authorization Service (JAAS) provider. JAAS is a Java package that enables applications to authenticate and enforce access controls upon users. The use of JAZN in OracleAS Portal is limited to the authentication of external JSPs.

J2EE Security

JPDK Web providers can leverage OC4J J2EE security roles for implementing authorization logic when the container is configured for JAZN LDAP and the portal is configured to use enhanced authentication to ensure message integrity.

See Also: Section 6.3.1.3.2, "Enhanced Authentication"

6.1.6 Leveraging Oracle Identity Management Infrastructure

To provide a more comprehensive security solution, OracleAS Portal takes advantage of a variety of components in the Oracle Identity Management infrastructure:

- Relationship Between OracleAS Portal and OracleAS Single Sign-On
- Relationship Between OracleAS Portal and Oracle Internet Directory
- Relationship Between OracleAS Portal and Oracle Directory Integration Platform
- Relationship Between OracleAS Portal and Oracle Delegated Administration Services

OracleAS Portal also takes advantage of Oracle Identity Management when it creates users and groups. The most common way to create users and groups, and set global privileges and preferences for your portal is through the following portlets:

- User Portlet
- Portal User Profile Portlet
- Group Portlet
- Portal Group Profile Portlet

See Also: *Oracle Identity Management Concepts and Deployment Planning Guide*

6.1.6.1 Relationship Between OracleAS Portal and OracleAS Single Sign-On

OracleAS Portal uses OracleAS Single Sign-On for user authentication. Refer to Section 6.1.4, "Authorization and Access Enforcement" for more information.

6.1.6.1.1 OracleAS Portal As an OracleAS Single Sign-On Partner Application OracleAS Single Sign-On manages the Single Sign-On sessions of users. In order for single sign-on security to function properly with OracleAS Portal, the following tasks must be completed:

- Add OracleAS Portal as a partner application for OracleAS Single Sign-On.

- Add OracleAS Portal entries to the partner application enabler configuration table.

The Oracle Universal Installer performs these two configuration steps for you upon installation. If you need to make changes to your configuration after installation, you can do so by:

- Using the Application Server Control Console. Refer to Section 7.2, "Using the Application Server Control Console" for details. Alternatively, you can use the Portal Dependency Settings tool. Refer to Appendix A, "Using the Portal Dependency Settings Tool and File" for details.
- Running the `ptlconfig` tool with the `-site` option. This procedure adds OracleAS Portal as a partner application to an existing OracleAS Single Sign-On installation. To work correctly, you must already have installed OracleAS Portal and OracleAS Single Sign-On, and created their DADs. See Appendix A, "Using the Portal Dependency Settings Tool and File" for details.

6.1.6.1.2 Support for External Applications OracleAS Single Sign-On supports the concept of External Applications, which retain their own authentication mechanisms, but for which OracleAS Single Sign-On can automate logins for OracleAS Portal users. This is achieved by providing an External Applications portlet that exposes the list of external applications registered with the single sign-on server.

6.1.6.1.3 Support for Global Inactivity Timeout in OracleAS Portal A Global Inactivity Timeout can be configured for the OracleAS Single Sign-On Server. This feature is supported from OracleAS Portal 10*g* Release 2 (10.1.2) onwards. To use this feature, you must configure the OracleAS Single Sign-On Server on the infrastructure tier and `mod_osso` on the Oracle Application Server middle tier. See the *Oracle Application Server Single Sign-On Administrator's Guide* for more information on configuring this feature.

6.1.6.2 Relationship Between OracleAS Portal and Oracle Internet Directory

Oracle Internet Directory is Oracle's highly scalable, native LDAP version 3 service and hosts the Oracle common user identity. As stated in the previous section, OracleAS Portal queries the directory to determine a user's privileges and what they are entitled to see and do in the portal. In particular, OracleAS Portal retrieves the group memberships of the user from the directory to determine what they may access and change.

Given this model, OracleAS Portal requires the following interactions with Oracle Internet Directory:

- OracleAS Portal specific entries stored in the directory
- Group attributes stored in the directory
- User attributes stored in the directory
- Caching of user and group information from the directory
- Populating user and group lists of values from the directory through Oracle Delegated Administration Services

OracleAS Portal makes use of the features in Oracle Internet Directory to provide efficient user and group management. For a complete list of features provided by Oracle Internet Directory, refer to the chapter, "What's New in Oracle Internet Directory" in the *Oracle Internet Directory Administrator's Guide*. However, a few of Oracle Internet Directory features are not supported by OracleAS Portal.

Table 6–11 lists and describes the Oracle Internet Directory features not supported in OracleAS Portal.

Table 6–11 Oracle Internet Directory Features Not Supported in OracleAS Portal

Features	Description
Dynamic Groups	In dynamic groups, membership is computed dynamically based on specific attribute values and assertions that you specify. The Dynamic Group feature is not supported in OracleAS Portal because it relies on the use of cached information for greater performance and scalability. That is, user and group information is cached within the portal environment while assembled pages and content are cached within OracleAS Web Cache. The invalidation and subsequent flushing of the cache occurs when the definition of a user or the user's group membership changes. The change is propagated to OracleAS Portal using a Directory Integration Subscription Event. In the case of dynamic groups, there is no group change event because the membership is evaluated for each LDAP query and therefore there is no method of propagating this to the portal. You can use the Oracle Internet Directory Plug-in functionality to generate a static group populated by the outcome of a Dynamic Group query. This enables the use of groups based on attributes, but the provisioning and maintenance of this group may have a significant impact on the overall performance of the system (based on the size of the group and so on).
Single Authentication Security Layer (SASL)	SASL is a method for adding authentication support to connection-based protocols. Oracle Internet Directory server supports SASL-based authentication mechanisms, but currently these mechanisms are not supported in the DBMS_LDAP package, which is used by OracleAS Portal.
Client-side referral caching	For performance reasons, an API can be used to cache the cached referral entries on the client side. Currently, there is no support for client-side referral caching in the DBMS_LDAP package, which is used by OracleAS Portal.

6.1.6.2.1 Directory Entries in Oracle Internet Directory for OracleAS Portal In order for security to function properly, OracleAS Portal requires the following entries in the directory's Directory Information Tree (DIT) structure:

- **Default user accounts** (cn=PUBLIC, cn=PORTAL, cn=PORTAL_ADMIN) are created in the identity management realm's user base (cn=Users,dc=MyCompany,dc=com[1]). The PORTAL and PORTAL_ADMIN users are added to the DBA and PORTAL_ADMINISTRATORS groups, respectively. The PUBLIC user is created for unauthenticated users. Typically, the PUBLIC user entry is for granting viewing privileges on portal content that is accessible to any user, unrestricted.
- **Group container** is created within the identity management realm's group base (cn=Groups,dc=MyCompany,dc=com[1]). OracleAS Portal can leverage any group

[1] The default identity management realm name is determined by the domain name of the server on which the system is installed. For example, if the domain name server was oracle, the default identity management realm name would be dc=oracle,dc=com. If the domain name server cannot be determined, the default name assigned by the directory is dc=Default Company,dc=com

in the directory, but groups are more easily accessed for display in a list of values if they are located within the OracleAS Portal group container.

The name of the group container is derived from the following in OracleAS Portal:

- OracleAS Portal schema name
- Date and time when OracleAS Portal began to use Infrastructure Services

The format of the name is:

```
schema_name.yymmdd.hh24miss.ff
```

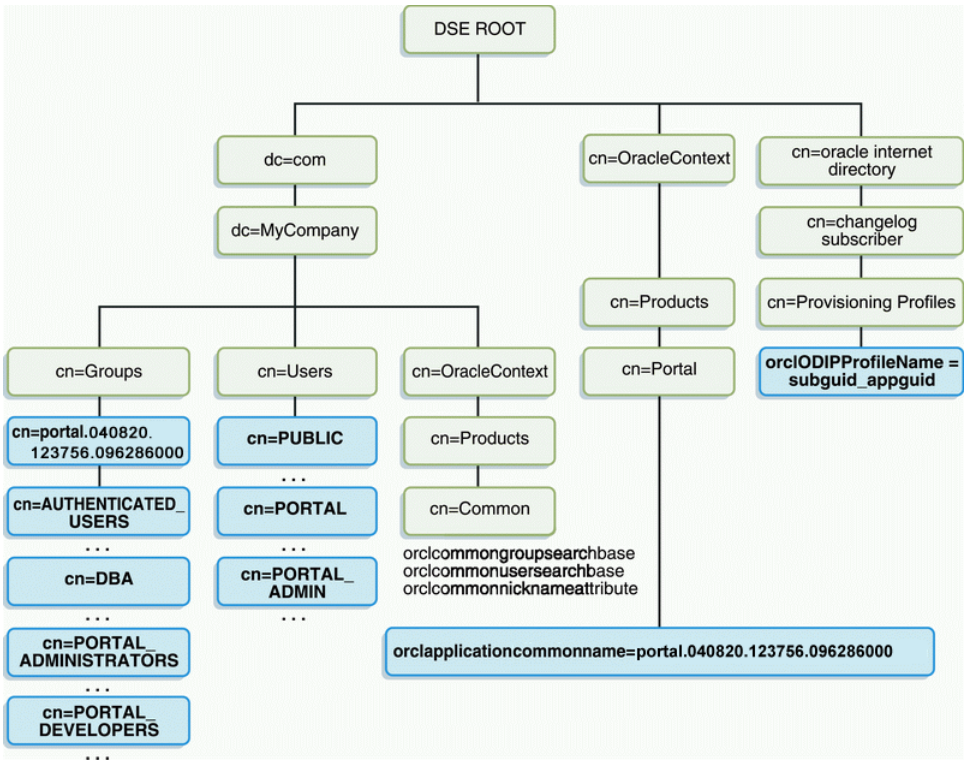
> **Note:** The name of the group container may have a different format for releases of OracleAS Portal older than 10*g* Release 1.

- **Groups** are created within the OracleAS Portal group container in the directory:
  - cn=AUTHENTICATED_USERS
  - cn=DBA
  - cn=PORTAL_ADMINISTRATORS
  - cn=PORTAL_DEVELOPERS
  - cn=PORTLET_PUBLISHERS
  - cn=RW_ADMINISTRATOR
  - cn=RW_DEVELOPER
  - cn=RW_POWER_USER
  - cn=RW_BASIC_USER
- **Application entity** (orclApplicationCommonName=*application_name*) is created in the root Oracle Context (cn=Portal,cn=Products,cn=OracleContext). The application password is randomly generated. OracleAS Portal uses this entity to bind to the directory when it needs to query it or perform actions against it (for example, adding a user) on behalf of the user. When OracleAS Portal binds to the directory for a user, it uses a proxy connection to connect as the user. This method ensures that the directory properly enforces the user's authorization restrictions. The OracleAS Portal application entity obtains the privileges to initiate proxy connections by its membership in the user proxy privileges group (cn=UserProxyPrivilege,cn=Groups,cn=OracleContext). The name of the application entity is derived from the schema and the time that OracleAS Portal began to use the Infrastructure Services. For example, the name of the application entity can be portal.040820.123756.096286000, where portal is the schema name and 040820.123756.096286000 is the timestamp in the yymmdd.hh24miss.ff format.
- **Directory synchronization subscription** A provisioning profile entry is created in the provisioning profiles node of the directory (cn=Provisioning Profiles,cn=changelog subscriber,cn=oracle internet directory). This entry indicates that the directory must notify OracleAS Portal when user or group privilege information has changed. It enables OracleAS Portal to keep its authorizations synchronized with the information stored in the directory.

Note: When the provisioning profile is deleted from Oracle Internet Directory, the **Enable directory synchronization** and **Send event notifications every n seconds**, disappear from the **Directory Synchronization** section on the **Global Settings** page's **SSO/OID** tab. To navigate to the **Global Settings** page, in the **Portal Builder**, go to the **Portal** subtab on the **Administer** tab, and click the **Global Settings** link in the **Services** Portlet.

Figure 6–3 shows where the OracleAS Portal information is located in the directory's DIT structure.

Figure 6–3 OracleAS Portal DIT Structure



Under cn=Groups,cn=OracleContext,<*subscriber_dn*>, there is a cn=Groups container, that contains the following groups:

authenticationServices
userProxyPrivilege
iasadmins
OracleDASCreateUser
OracleDASEditGroup
OracleDASCreateGroup
OracleDASEditUser
OracleDASDeleteUser
OracleDASUserPriv
OracleUserSecurityAdmins
OracleDASDeleteGroup
OracleDASGroupPriv
OracleDASConfiguration

These privilege group entries are modified during a portal installation to add portal groups and the portal application entry to their memberships, to achieve desired portal functionality. As such, portal information is contained in these groups as well.

6.1.6.2.2 User Attributes Stored in Oracle Internet Directory OracleAS Portal, like all other components of Oracle Application Server, relies upon the directory to store user information. All users in the directory are defined using the following object classes:

- The inetOrgPerson object class contains the entire user attributes defined by the Internet Engineering Task Force (IETF) Request for Comments (RFC) number 2798.
- The orclUser and orclUserV2 object classes contain a set of standard, additional attributes for Oracle products.

The subsequent tables show the various user attributes stored in Oracle Internet Directory.

Table 6–12 inetOrgPerson Attributes

inetOrgPerson (IETF) attributes	Comment
cn	Common name of the user This attribute is mandatory.
employeeNumber	Number used to identify employees
sn	Last name. This attribute is mandatory. If nothing is explicitly specified for this attribute, the user's nickname is used.
givenName	First name
middleName	
displayName	Preferred name
mail	e-mail address
telephoneNumber	
homePhone	
mobile	
pager	
facsimileTelephoneNumber	
street	
l	City of office
st	State of office
postalCode	Postal code of office
c	Country of office
homePostalAddress	Home address
jpegPhoto	Person's picture
o	Organization
title	
manager	Employee's supervisor
uid	User ID
userPassword	

Table 6–12 (Cont.) inetOrgPerson Attributes

inetOrgPerson (IETF) attributes	Comment
preferredLanguage	

Table 6–13 orclUserV2 Attributes

orclUserv2 attributes	Comments
orclIsVisible	A flag to indicate whether the user should be hidden from all but administrators.
orclDisplayPersonalInfo	A flag to indicate whether a user's personal information should be hidden from all but administrators.
orclMaidenName	
orclDateOfBirth	
orclHireDate	
orclDefaultProfileGroup	Default user group for the person
orclActiveStartDate	When account was activated
orclActiveEndDate	when account was (or will be) terminated
orclTimeZone	
orclIsEnabled	A flag to indicate whether the user account is active. If not active, the user will not be allowed to log in.

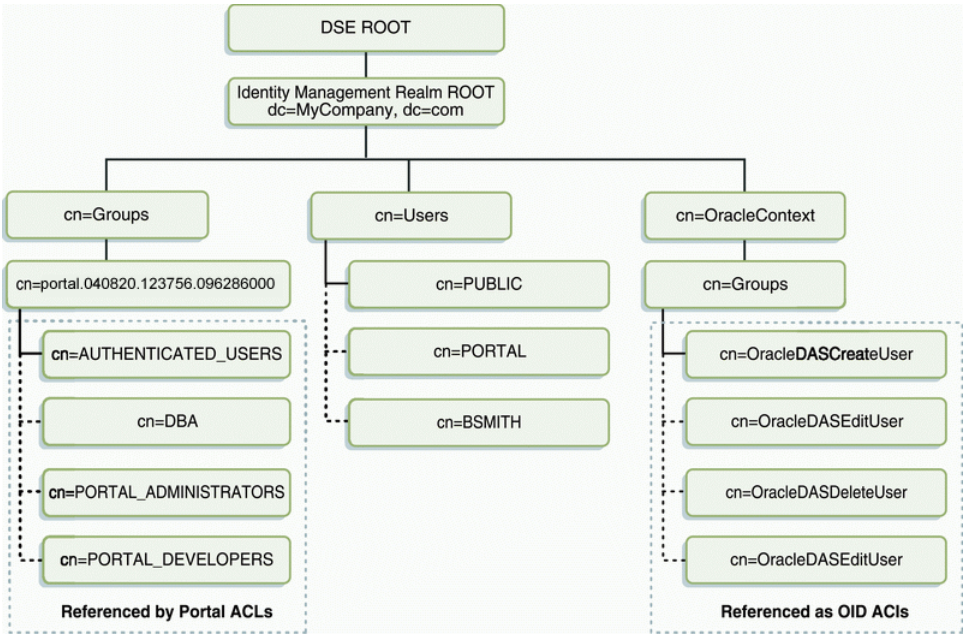
6.1.6.2.3 Group Attributes Stored in Oracle Internet Directory OracleAS Portal, like all other components of Oracle Application Server, relies upon the directory to store group information. All groups in the directory are defined using the following object classes:

- The `groupOfUniqueNames` object class contains all of the group attributes defined by IETF (RFC 2256).
- The orclGroup object class contains a set of standard, additional attributes for OracleAS Portal.

> **Note:** In OracleAS Portal 9.0.2 and subsequent releases, you cannot scope groups to a specific page group. This option was available only in OracleAS Portal 3.0.9.x and preceding releases.

Figure 6–4 shows where the OracleAS Portal information for groups is located in the directory's DIT structure.

Figure 6–4 DIT Structure for OracleAS Portal Groups



The subsequent tables show the various group attributes stored in Oracle Internet Directory.

Table 6–14 groupOfUniqueNames/groupOfNames Attributes

groupOfUniqueNames/groupOf Names (IETF) attributes	Comment
cn	The common name of the group, which can be typed into places like the Edit Group field in the Group portlet to locate the group.
description	The text description of the group, which is displayed in lists of values where the group appears.
uniqueMember/member	A list of the distinguished names (DNs) of all of the members of the group. The member DNs can represent a user or another group.
owner	A list of the DNs of all of the users and groups that have the privilege of administering this group.

Table 6–15 orclGroup Attributes

orclGroup attributes	Comment
orclGUID	The globally unique identifier (GUID) for this group.
orclIsVisible	A flag to indicate whether the group is public or private. Private groups only appear in lists of values for their owners. Other users cannot see them.

6.1.6.2.4 Oracle Internet Directory Cache in OracleAS Portal To improve performance, OracleAS Portal caches some directory information locally. In particular, OracleAS Portal caches the following:

- Directory connection information for OracleAS Portal
- URLs for Oracle Delegated Administration Services
- orclGUIDs of certain privilege groups for authorization checks on directory portlets (for example, the User and Group portlets)
- some Oracle Context information
- the locally selected group search and creation bases
- group memberships and default group for each user

The majority of the information cached by OracleAS Portal is fairly static (for example, directory connection information). For those items that are more dynamic, such as group memberships and default group, OracleAS Portal relies upon the Oracle Directory Integration and Provisioning agent for updates. OracleAS Portal maintains a directory synchronization subscription in the directory that flags the agent to notify it of any change events that affect OracleAS Portal security (for example, adding or deleting a user from a group).

6.1.6.2.5 User and Group Lists of Values in OracleAS Portal The User, Group, Portal User Profile, and Portal Group Profile portlets include lists of values for users or groups. These lists of values must be populated with information stored in the directory. By default, the list of values displays the groups contained under the OracleAS Portal group container in the OracleAS Portal DIT structure. You can browse to any group in the tree, if you have the right access privileges.

The groups that are displayed in the list of values for groups depend on the privileges of the user viewing them. For example, if a user views the list of values from the Group portlet, the list only displays those groups that can be edited or deleted by that user. From OracleAS Portal 10*g* Release 2 (10.1.2) onwards, the implementation of the LOVs supports a callback method. This callback mechanism requires corresponding support in the Oracle Delegated Administration Services environment.

If you have upgraded from a release of OracleAS Portal earlier than 10*g* Release 2 (10.1.2), and if your Infrastructure and Application Server middle tier were separated onto different hosts or protocols, you may have performed additional user and group Lists of Values (LOVs) configuration to accommodate the JavaScript Origin Server Security policy.

There were two configuration options:

- Setting up of a common-domain by running the script `secjsdom.sql`.
- Deploying Oracle Delegated Administration Services on the middle tier.

If you have installed the appropriate Oracle Delegated Administration Services version in your environment, and have not previously implemented the configuration options mentioned in the preceding text, then no subsequent configuration steps are required in OracleAS Portal to support the LOVs on a separate host. However, if you used the configuration options mentioned previously, it is required to remove these steps. This can be done as follows:

1. If a common domain was defined, reset it by executing the `secjsdom.sql` script. Refer to Example C–2, "Resetting a Previously Defined Common Domain Using secjsdom.sql" for more information.

2. If OracleAS Portal has been configured to use a locally deployed Oracle Delegated Administration Services servlet, reconfigure it to point to the Infrastructure tier by running the `secdaslc.sql` script as follows:

   a. From the operating system prompt, go to the following directory:

      DESTINATION_MID_TIER_ORACLE_HOME`/portal/admin/plsql/wwc`

   b. Using SQL*Plus, connect to the OracleAS Portal schema as the owner and run the following commands:

```
@secdaslc N
commit;
```

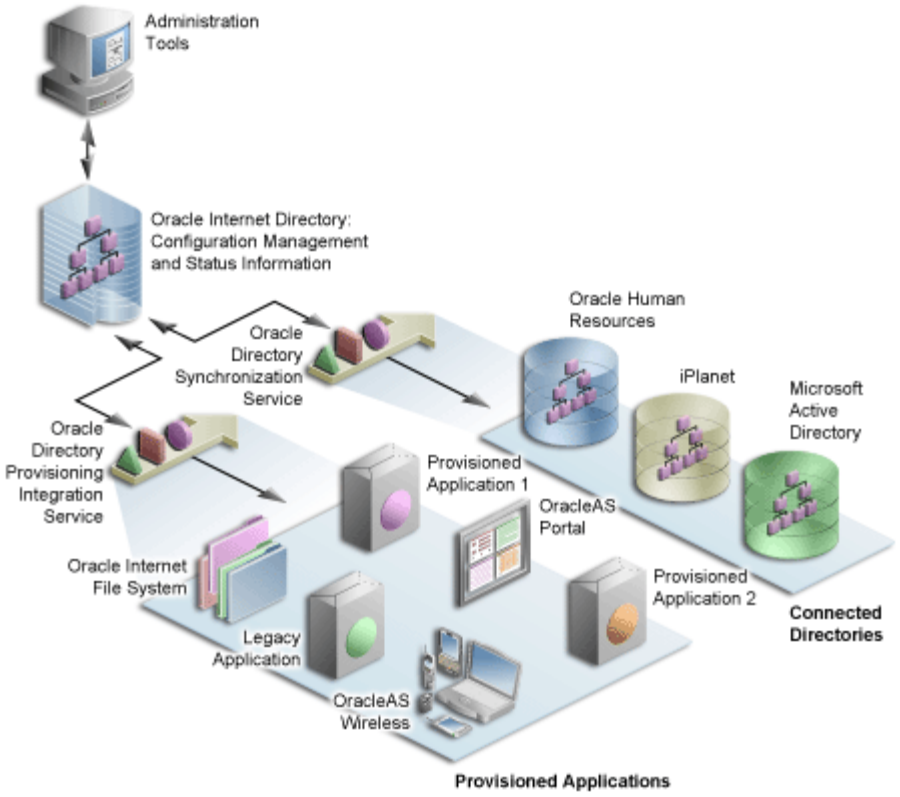
If OracleAS Portal is used with an older release of Oracle Delegated Administration Services that does not support the callback method and the directory and OracleAS Portal servers reside on different domains, you have to install the required patch to the Oracle Delegated Administration Services environment to support the use of LOVs across domains.

If it is not possible, for operational purposes, to apply the patch, you can define a Common JavaScript Domain for Oracle Delegated Administration Services Lists of Values by using the `secjsdom` script. Refer to Section C.4, "Using the secjsdom.sql Script" for information about using this script.

6.1.6.3 Relationship Between OracleAS Portal and Oracle Directory Integration Platform

As shown in Figure 6–5, the Oracle Directory Integration Platform provides important services to notify components of user and group change events and synchronize directories.

Figure 6–5 Oracle Directory Integration Platform Synchronization



In the figure, the flow to and from the Oracle Internet Directory has two paths. The first path, labeled Oracle Directory Synchronization Service, shows the concept of synchronization. In this case, the Oracle Internet Directory acts as a gateway to some external directory or repository. The synchronization service ensures that changes are coordinated between the Oracle Internet Directory and its connected directories. Whenever a change occurs in one of the directories, a notification must be raised with the Oracle Internet Directory to appropriately reflect the change across all of the affected directories.

The second path, labeled Oracle Directory Provisioning Integration Service, shows the concept of provisioning. In provisioning, an application, such as OracleAS Portal, subscribes to changes to certain user or group information. For example, suppose that an administrator removes a user from a group through the Oracle Delegated Administration Services. As a result of this change, the user should no longer be allowed to access certain pages in OracleAS Portal. The Oracle Directory Integration Platform must notify OracleAS Portal to update its local cache and immediately prevent the user from accessing the pages to which she no longer should have access.

For provisioning services, components like OracleAS Portal subscribe to provisioning events (for example, deletion of a group) to keep their local caches of user and group information synchronized with the central user and group repository in the Oracle Internet Directory. When a change event occurs, all of the components that are subscribed to that change event are notified by the Directory Synchronized Provisioning agent of the Oracle Directory Integration Platform. OracleAS Portal sets the portal directory synchronization subscription flag in the directory to indicate that it should be notified whenever a subscribed change event takes place. Table 6–16 shows the events to which OracleAS Portal subscribes and the actions it takes when those events occur.

Table 6–16 Directory Synchronized Events Handled By OracleAS Portal

Subscribed event	OracleAS Portal action
USER DELETE	The local user profile entry is deleted, resulting in the deletion of the user's privileges. Pages associated with this user are invalidated in OracleAS Web Cache.
USER MODIFY (orclDefaultProfileGroup)	The default group of the user is changed in the local user profile.
GROUP DELETE	The local group profile is deleted, resulting in the deletion of the privileges assigned to this group. The WWSEC_FLAT$ table is updated accordingly.
GROUP MODIFY (uniqueMember, member)	The WWSEC_FLAT$ table is updated to reflect membership changes that affect OracleAS Portal. If the membership changes involve a group being added or deleted from the modified group, the pages associated with the users of the added or deleted group are invalidated in OracleAS Web Cache. The reason for this action is that the security changes might affect what is visible on the page or the access privileges of the page itself.

Note: OracleAS Portal does not need to subscribe to user and group creation events. The local user profile is created automatically when a new user first logs on or is assigned some privilege that causes the user to be referenced in an access control list (ACL) of OracleAS Portal. Similarly, a local group profile is created automatically when a new group is first referenced in an ACL.

To function properly, OracleAS Portal requires the following for its integration with Oracle Directory Integration Platform:

- The Oracle Directory Integration Platform must be running. With OracleAS Portal Release 9.0.4 and later, the Oracle Directory Integration and Provisioning agent is started by default if the user had started the infrastructure tier using `opmnctl start all`. To start the Oracle Directory Integration Platform, you use the `oidctl` command, for example:

```
oidctl instance=1 server=odisrv flags="host=iasqa-ultra1.abc.com port=4032"
start
```

- The subscription profile must be created in the Oracle Internet Directory. A default subscription profile is automatically created during the installation of OracleAS Portal.

See Also: *Oracle Internet Directory Administrator's Guide*

6.1.6.3.1 Update Subscription Profiles for Groups Based on groupOfNames By default, groups created in the Oracle Internet Directory by Oracle Delegated Administration Services are based on the IETF object class `groupOfUniqueNames`. However, there is now support for handling groups created with the object class `groupOfNames` as well. If your portal has an existing Oracle Directory Integration Platform subscription profile in the Oracle Internet Directory (from 9.0.2), then it would be subscribing to group modifications and deletions based on groups using `groupOfUniqueNames`. If any existing groups in Oracle Internet Directory are based on the `groupOfNames`

object class you must update the Oracle Directory Integration Platform subscription profile to subscribe to the events for groups based on `groupOfNames` in addition to `groupOfUniqueNames`.

To create or update the subscription profile, run `ptlconfig` as follows:

```
ptlconfig -dad <dad> -dipreg
```

This will create or update the provisioning profile and subscribe to changes for the `uniqueMember` and `member` attributes.

By default, this provisioning profile is enabled.

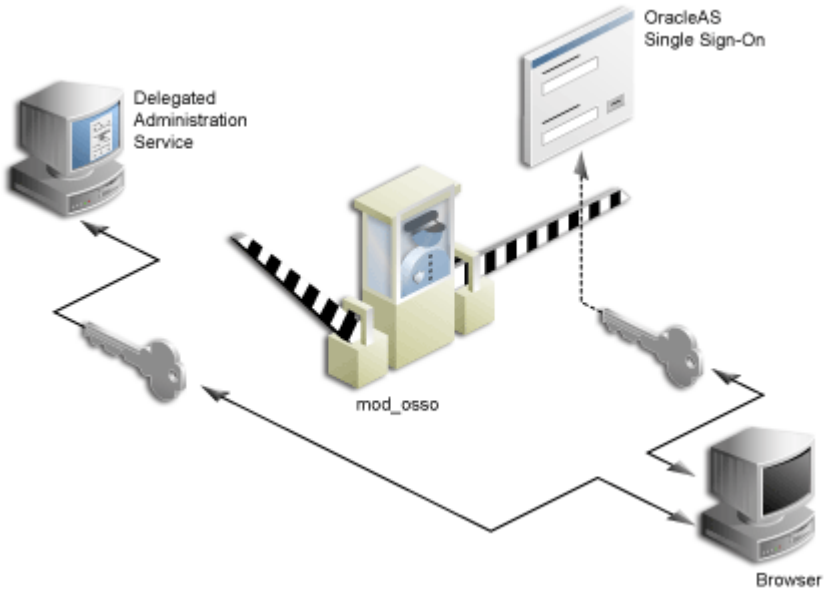
6.1.6.4 Relationship Between OracleAS Portal and Oracle Delegated Administration Services

In addition to querying the directory for user and group information, OracleAS Portal must provide users with the means to add and modify user and group information. To change information in the directory, use the Oracle Delegated Administration Services. OracleAS Portal provides links to the delegated administration server for users with the privileges to add and change users and groups.

6.1.6.4.1 Creating and updating information Stored in Oracle Internet Directory The Oracle Delegated Administration Services provides a comprehensive interface for making updates to the directory. Authenticated users who have the appropriate privileges can access the delegated administration server through the User and Group portlets on the **Administration** tab in OracleAS Portal. To access these portlets, a user must be a member of the OracleDASCreateUser and OracleDASCreateGroup groups, respectively. The PORTAL and PORTAL_ADMIN users are members of both of these groups by default. AUTHENTICATED_USERS may also create groups by default.

6.1.6.4.2 Relationship Between Oracle Delegated Administration Services, mod_osso, and the OracleAS Single Sign-On mod_osso protects URLs behind the OracleAS Single Sign-On environment by making the HTTP server effectively into a partner application. Oracle Delegated Administration Services functionality is single sign-on enabled by using mod_osso to get the user's identify from the OracleAS Single Sign-On session.

Figure 6–6 Relationship between Oracle Delegated Administration Services, mod_osso, and OracleAS Single Sign-On



mod_osso is a module of the Oracle HTTP Server that is written as a partner application. You can use mod_osso to enable applications, including OC4J applications, for single sign-on. You achieve this by configuring mod_osso with Oracle HTTP Server directives to restrict access to the OC4J application URLs.

Oracle Delegated Administration Services is implemented as an OC4J application, which relies on mod_osso to authenticate users attempting access. When a user attempts to access an Oracle Delegated Administration Services dialog (for example, a list of users or groups, or the Create User form), mod_osso checks whether the user has been authenticated. mod_osso performs no authorization checks other than checking for authentication. If the user has not been authenticated, mod_osso, which is an OracleAS Single Sign-On partner application, redirects the user's request to OracleAS Single Sign-On. OracleAS Single Sign-On either:

- Finds a cookie that indicates the user has been properly authenticated and sends back an authenticated token to mod_osso.
- Or, if no cookie has been created yet, it brings up the login page to authenticate the user.

Once the user has been properly authenticated, they are redirected by mod_osso to the requested Oracle Delegated Administration Services URL. Oracle Delegated Administration Services then becomes accessible to the user and enforces the user's privileges, typically relying on access control items in the Oracle Internet Directory.

Oracle Delegated Administration Services URLs

The first request to Oracle Delegated Administration Services from a user session in OracleAS Portal is redirected to the OracleAS Single Sign-On so that mod_osso, which acts as a partner application on behalf of Oracle Delegated Administration Services, can establish the identity of the user. OracleAS Single Sign-On constructs a URLC token that includes the requested Oracle Delegated Administration Services URL. There is about a 2K limit on the length of the URLC token imposed by Internet Explorer. As such, the length of the Oracle Delegated Administration Services URL is also limited. To provide a seamless integration with Oracle Delegated Administration Services, OracleAS Portal includes the URLs of the current portal page and the portal

home page within this Oracle Delegated Administration Services URL. A typical Oracle Delegated Administration Services URL appears as follows:

```
http://myportal.us.abc.com:7777/oiddas/ui/oracle/ldap/das/group/AppCreateGroupInfo
Admin?doneURL=https%3A%2F%2Fwebsvr.us.abc.com%3A5001%2Fportal%2Fpage%3F_
pageid%3D6%2C1%2C6_12%3A6_18%26_dad%3Dportal_9_0_2_6_7%26_schema%3DPORTAL_9_0_2_
6_7&homeURL=https%3A%2F%2Fwebserver.us.abc.com%3A5001%2Fportal%2Fpage%3F_
pageid%3D6%2C1%2C6_12%3A6_18%26_dad%3Dportal_9_0_2_6_7%26_schema%3DPORTAL_9_0_2
_6_7&parentDN=cn%3Dportal_9_0_2_6_
7.s901dev0.portalserver.us.abc.com%2Ccn%3Dgroups%2Cdc%3Dus%2Cdc%3Doracle%2Cdc%3Dco
m&enablePA=true
```

When this URL is included in the URLC token, which is then encrypted for security reasons, the length of the resulting token easily approaches the 2K threshold. If it exceeds this limit, the browser may show an error.

There is no fixed size for the URL. However, if you see browser errors when performing Oracle Delegated Administration Services operations, you should consider reducing the size of various parts that comprise the portal URL as this will help ensure that the URL does not exceed the 2k limit. For example, limit hostnames to 8 characters or less and DAD names to 6 characters or less.

In the event that you encounter this problem, the workaround is to log in to Oracle Delegated Administration Services first through a shorter URL, such as the **Directory Administration** link in the **Services** portlet. Any subsequent access to Oracle Delegated Administration Services will then not require SSO redirection, and will succeed.

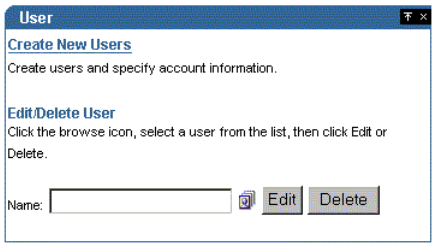
6.1.6.5 User Portlet

The **User** portlet on the **Portal** tab under **Administration** enables you to create and update users through Oracle Delegated Administration Services. To create a new user, click the **Create New Users** link in the User portlet. To update information for an existing user, enter their user name in the **Name** field or choose it from the list of values and click **Edit**. To delete a user, enter their user name in the **Name** field or choose it from the list of values and click **Delete**.

Note: Only a user who is a member of the OracleDASCreateUser, OracleDASEditUser, or OracleDASDeleteUser privilege groups can see the User portlet. The link to create new users is displayed only to users who are members of the OracleDASCreateUser group.

See Also: *Oracle Internet Directory Administrator's Guide*

Figure 6–7 User Portlet



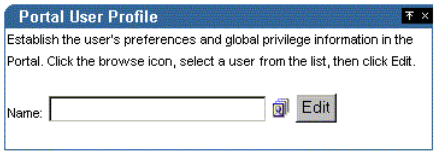
6.1.6.6 Portal User Profile Portlet

> **Note:** The Portal User Profile portlet is only visible to users with Manage or Edit privileges for All User Profiles.

To set global user privileges and preferences that pertain specifically to the portal, use the Portal User Profile portlet. To update a user's portal preferences and privileges, enter their user name in the **Name** field or choose it from the list of values. You can set all of the following for the user's profile:

- Preferences
  - whether the user can access the portal
  - database schema name for the user
  - whether the user has a personal page
  - default user group for the user
  - default home page for the user
  - default style for the user
  - whether to clear the OracleAS Web Cache for the user
- Global Privileges
  - page group privileges
  - Portal DB Provider privileges
  - administration privileges

Figure 6–8 Portal User Profile Portlet



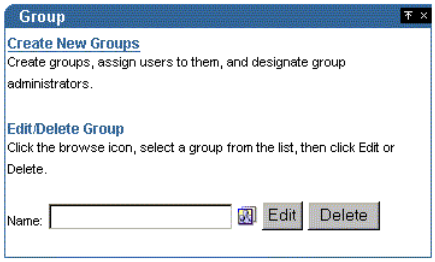
6.1.6.7 Group Portlet

> **Note:** Every user can see the Group portlet, but the link to create new groups is displayed only to users who are members of the OracleDASCreateGroup privilege group. Users can only edit or delete a group if they are the group's owner or a member of a group with appropriate access control information (ACI) to edit or delete the group. The following privilege groups are seeded in the Oracle Internet Directory:
>
> - OracleDASCreateGroup
> - OracleDASEditGroup
> - OracleDASDeleteGroup
>
> The preceding privilege groups imply global privileges, and should be allocated carefully.

The **Group** portlet on the **Portal** tab under **Administration** enables you to create and update user groups through Oracle Delegated Administration Services. To create a new group, click the **Create New Group**s link in the Group portlet. To update information for an existing group, enter its name in the **Name** field or choose it from the list of values and click **Edit**. To delete a group, enter the group name in the **Name** field or choose it from the list of values and click **Delete**.

See Also: *Oracle Internet Directory Administrator's Guide*

Figure 6–9 Group Portlet



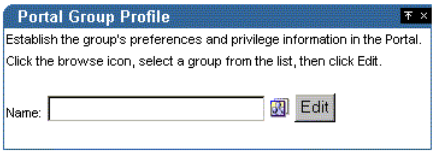
6.1.6.8 Portal Group Profile Portlet

Note: The Portal Group Profile portlet is displayed to all users, but only users with the Manage or Edit privilege for All Group Profiles, or the owner of a group can edit its profile.

To set global group preferences and privileges that pertain specifically to the portal, you need to use the Portal Group Profile portlet. To update a group's portal preferences and privileges, enter the group name in the **Name** field or choose it from the list of values. You can set all of the following for the group's profile:

- Preferences
  - default home page for the group
  - default style for the group
- Global Privileges
  - page group privileges
  - Portal DB privileges
  - administration privileges

Figure 6–10 Portal Group Profile Portlet



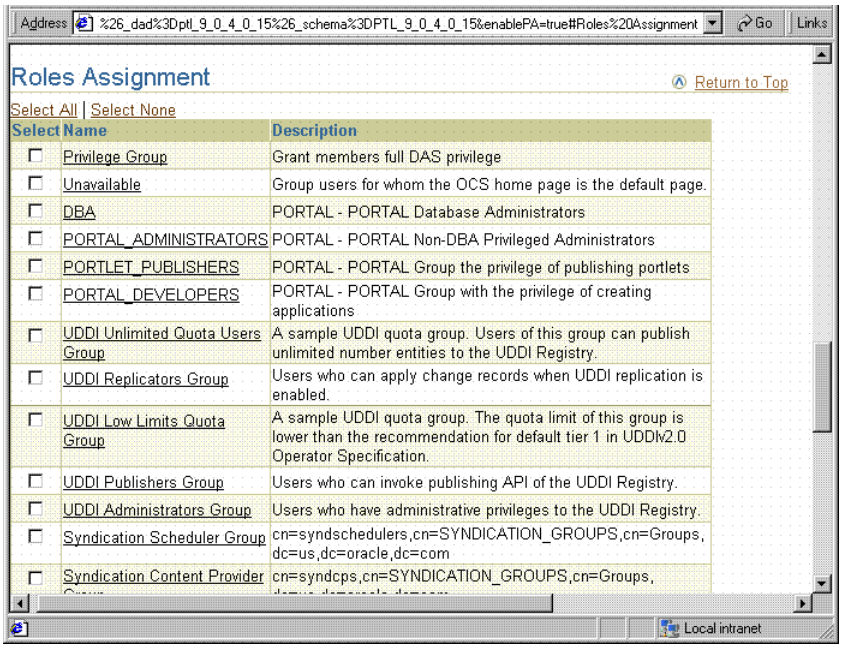
6.1.6.9 Oracle Delegated Administration Services Public Roles

In many cases, it is more efficient to use roles exposed by Oracle Delegated Administration Services to assign privileges for each individual user. When creating

users, you might notice a section called Roles Assignment on the Create User page, shown in Figure 6–11.

> **Note:** In releases before 9.0.4, roles were called public groups.

Figure 6–11 OracleAS Portal Create User Page



Roles provide a very convenient mechanism by which users can be created and granted a set of privileges simultaneously. When a check box for a role is checked for a given user, they are granted the designated role upon creation. As an administrator, you can create your own roles and pre-assign any combination of Oracle Internet Directory and OracleAS Portal privileges to them.

6.1.6.9.1 Example: Defining a User Administrator Role

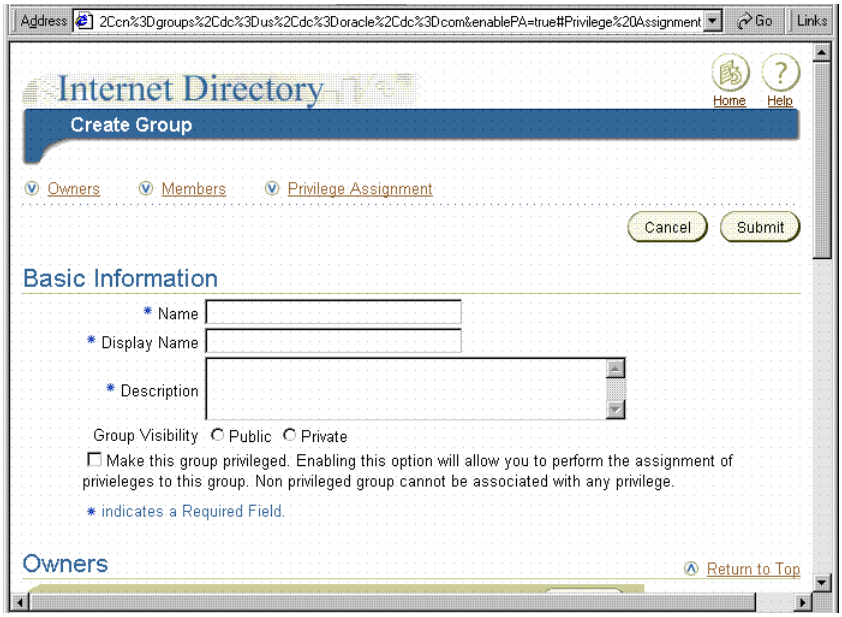
Suppose that you want to create a role with the appropriate privileges for a user administrator. You could create such a role by following these steps:

Step 1: Create a group.

You begin by creating a group in the usual manner:

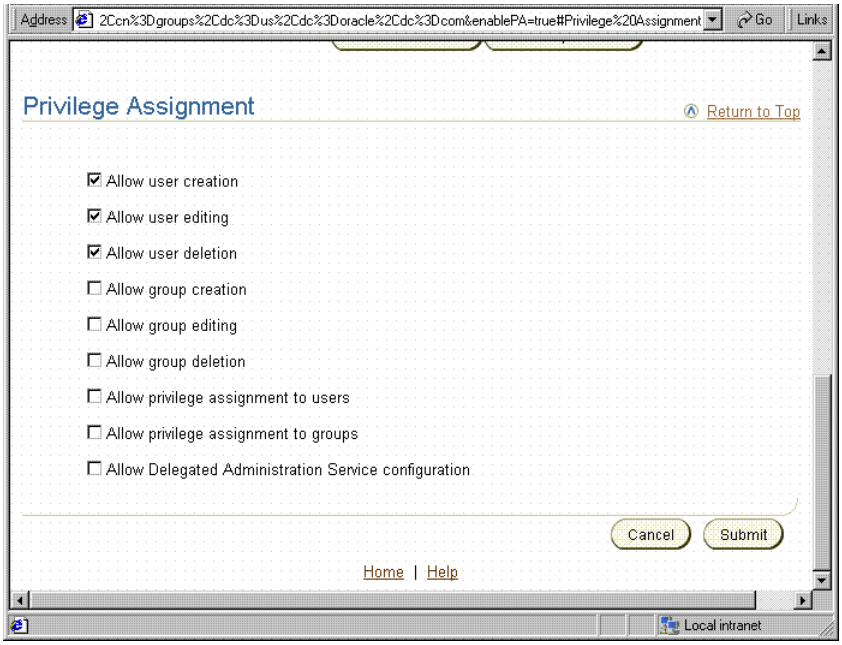
1. From **Portal Builder** (the Design-Time pages), click **Administer**, if you are not already on the **Administer** tab.
2. Click **Create New Group** in the Group portlet and the Create Group page appears, as shown in Figure 6–12.

Figure 6–12 Create Group Page



3. Enter the required fields (indicated by asterisks).
4. On the Create Group page, click **Privilege Assignment** to go to that section and choose the following privileges, as shown in Figure 6–13:
   - Allow user creation
   - Allow user editing
   - Allow user deletion

Figure 6–13 Privilege Assignment Section of the Create Group Page



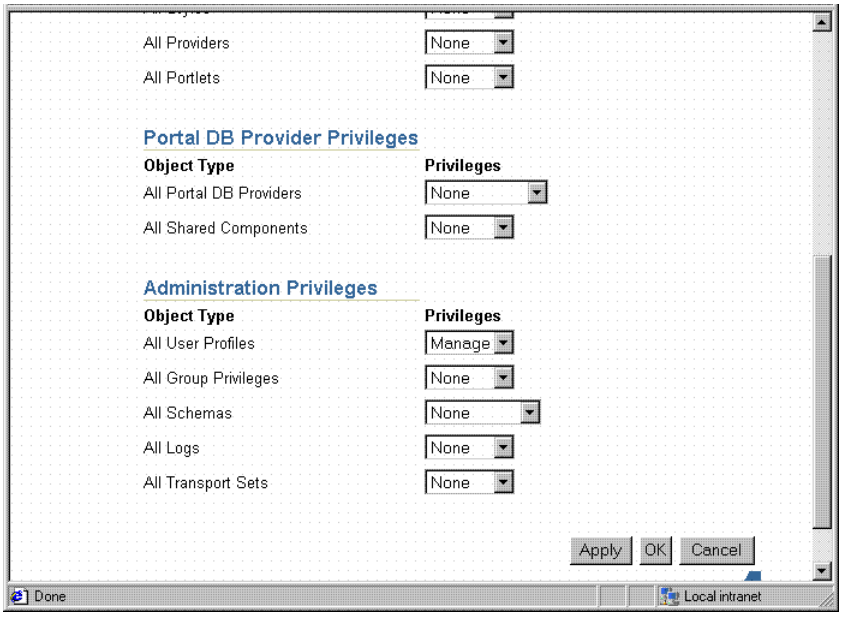
5. Click **Submit**.

Step 2: Assign the Manage privilege for all user profiles.

After you create the user administrator group, you need to assign it the Manage privilege for all user profiles. This privilege is the only global privilege that you need to assign to this group for user administration.

1. From **Portal Builder** (the Design-Time pages), click **Administer**, if you are not already on the **Administer** tab.
2. From the **Portal Group Profile** portlet, enter the name of your newly created group and click **Edit**.
3. Click **Privileges** to go to that tab.
4. Scroll down to the **Administration Privileges** section, shown in Figure 6–14. From the list next to **All User Profiles**, choose **Manage**.

Figure 6–14 Administration Privileges Section of the Edit Group Profile Page



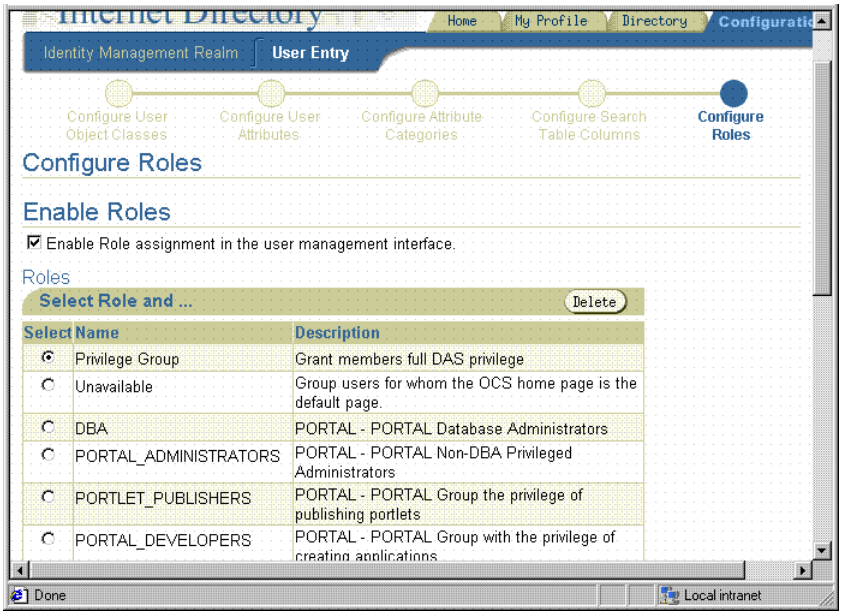
5. Click **OK**.

Step 3: Make the group a role.

Now that you have created a group representing the user administrator role, you need to enable it as a role so it appears in the list of roles on the Create User page.

1. From **Portal Builder** (the Design-Time pages), click **Administer**, if you are not already on the **Administer** tab.
2. In the **Services** portlet, click **Directory Administration**.
3. Click **Configuration** to display that tab.
4. Click **User Entry**.
5. Click **Next** until you reach Step 5 of 5, **Configure Roles**, of the wizard, as shown in Figure 6–15.
6. Click **Add Role** to choose the new group and add it to the list of roles.

Figure 6–15 Configure Roles Page



7. Click **Finish**. Your group will now appear in the list of public groups on the **Create User** page.

Step 4: Hide detailed privilege assignment section.

To encourage the usage of roles rather than direct privilege assignment, you can turn off the detailed privilege assignment section of the Create Users page. To implement this change, you need to update a configuration entry in the OracleAS Portal schema. This setting stops Oracle Delegated Administration Services from displaying the Privilege Assignment section in the **Create/Edit User** page when it is called from an OracleAS Portal administration page.

1. Log in to the PORTAL schema through SQL*Plus.

> **Note:** The PORTAL schema password is stored in the Oracle Internet Directory and the entry may be viewed by an administrator using the `oidadmin` utility with the following path under Entry Management:
>
> `OrclResourceName=PORTAL,orclReferenceName=iasdb.myhost.au.oracle.com,cn=IAS Infrastructure Databases,cn=IAS,cn=Products,cn=OracleContext`

2. Invoke the following commands to set the `das_enable_pa` variable in OracleAS Portal's Oracle Internet Directory configuration preference store:

```
$ sqlplus
...
Enter user-name: portal
Enter password:
...
SQL> set serverout on
SQL> exec wwsec_oid.set_preference_value('das_enable_pa', 'N');

PL/SQL procedure successfully completed.
```

```
SQL> commit;

Commit complete.

SQL> exit
...
```

3. Because the User Portlet is cached in OracleAS Web Cache and the OracleAS Portal middle-tier file system cache, you must invalidate the cached version of the portlet before you are done. Updating the configuration parameter changes the behavior of the portlet, but updating the parameter does not invalidate the cache. Follow these steps to invalidate the cached version of the User Portlet:
   a. Log in to OracleAS Portal as a user with administrator privileges.
   b. Go to the **Builder**.
   c. Click the **Administration** tab.
   d. From the **Portal** tab, open **Global Settings** and navigate to the **SSO/OID** tab.
   e. Scroll to the bottom of the page.
   f. Select **Refresh Cache for OID Parameters**.
   g. Click **Apply**.

      See Also: *Oracle Application Server Web Cache Administrator's Guide*

Step 5: Validate your changes.

Once you have performed steps 1 through 4, go to the **Create User** page to verify that your user administrator group appears there. Note how the other OracleAS Portal administrative roles, or groups, are already pre-seeded into the **Roles Assignment** list on this page.

6.1.7 Security for Portlets

Portlets act as windows on your application, displaying summary information and providing a way to access the full functionality of the application. Portlets expose application functionality directly in your portal or provide deep links that take you to the application itself to perform a task. Because portlets format information for display on a Web page, the underlying application need not be Web enabled to be integrated with OracleAS Portal.

In OracleAS Portal, portlets are managed by providers. A provider is an application that you register with OracleAS Portal. OracleAS Portal supports three types of providers:

- Web providers
- Database providers
- Web Services for Remote Portlets (WSRP) producers

Portlet security consists of three major areas of functionality:

- **Authentication:** When a user first accesses a secure URL, they must be challenged for information that verifies their identity, such as a user name and password, or a digital certificate.
- **Authorization:** Authorization is the process that allows certain users to access parts of an application. Some parts of an application may have public access while others may be accessible only to a limited number of authenticated users.

- **Communication security:** Communication security is the means by which OracleAS Portal establishes the authenticity of communications (for example, messages) to and from providers. In a heavily networked environment, it is critical to verify that communications are authenticate.

To make your Web providers truly secure, you need to make sure that they are secured in each of these areas. If you only implement security features for one or two out of three areas, then your providers cannot be considered secure. The effort you expend to secure a Web provider should be proportional to the confidentiality of the data the provider exposes.

To make your WSRP portlets secure, ensure that OracleAS Portal and the WSRP producers communicate through an inherently secure network, such as a VPN network or a network that is behind a firewall.

6.1.7.1 Authentication

When a user first logs in to OracleAS Portal, they must enter their password to verify their identity before being permitted access. OracleAS Single Sign-On manages the login process. Refer to Section 6.1.7.7, "Single Sign-On" for more information.

6.1.7.2 Authorization

Authorization determines whether a particular user should view or interact with a portlet. There are two types of authorization checks:

- **Portal Access Control Lists:** When you log in to OracleAS Portal, OracleAS Single Sign-On authenticates you. Once authenticated, OracleAS Portal uses access control lists (ACLs) to grant you privileges on portal objects such as pages and portlets. The actions may range from simply viewing an object to performing administrative functions. If you do not belong to a group that has been granted a specific privilege, OracleAS Portal prevents you from performing the actions associated with that privilege.
- **Programmatic Portlet Security:** The Portal Developer's Kit-Java includes APIs that are called to determine if a particular user is authorized to view a portlet. You can use these APIs to implement authorization logic that augments the Portal ACL security.
- **J2EE Security Roles:** You can use J2EE programmatic security in your provider code. To leverage this capability, you must configure your provider for enhanced authentication to protect the integrity of the asserted identity. From within your portlet code, you can use `request.isUserInRole("securityrole")`, where `request` is the `HttpServletRequest` request and `securityrole` is a declared J2EE security role. Refer to Section 6.3.1.3.2, "Enhanced Authentication" for more information on how to configure enhanced authentication.

6.1.7.3 Communication Security

Authentication and authorization are important components of securing your Web providers. They do not, however, check the authenticity of messages being received by a provider and are therefore not suitable on their own for securing access to a provider. If the communication is unsecured, someone could imitate OracleAS Portal and fool the Web provider into returning sensitive information.

Communication security focuses on securing communications between OracleAS Portal and a JPDK Web provider. These methods do not apply to database providers, which execute within the portal database. There are three types of communication security:

- Portal Server Authentication ensures that incoming messages came from a trusted host.
- Message Authentication ensures that the incoming messages were not tampered with.
- Message Encryption protects the contents of a message by encrypting them.

6.1.7.3.1 Portal Server Authentication Portal Server Authentication restricts access to a provider to a small number of recognized computers. This method compares the IP address or the hostname of an incoming HTTP message with a list of trusted hosts. If the IP address or hostname is in the list, the message is passed to the provider. If not, it is rejected before reaching the provider.

6.1.7.3.2 Message Authentication Message authentication works by appending a checksum based on a shared key to provider messages. When a message is received by the provider, the authenticity of the message is confirmed by calculating the expected value of the checksum and comparing it with the actual value received. If the values are the same, the message is accepted. If they are different the message is rejected without further processing. The checksum includes a time stamp to reduce the chance of a message being illegally recorded in transit and resent later.

See Also: Section 6.1.7.9, "Message Authentication"

6.1.7.3.3 Message encryption Message encryption relies on the use of the HTTPS protocol for communication between the provider and OracleAS Portal. Messages are strongly encrypted to protect the data therein and provide confidentiality. While encryption provides a high level of security, it also of necessity impacts performance.

Note: Use of the HTTPS protocol for communication between OracleAS Portal and WSRP producers is not supported in this release.

6.1.7.4 Access Control Lists

When you log in to OracleAS Portal, OracleAS Single Sign-On authenticates you. OracleAS Portal then uses access control lists (ACLs) to determine if you are authorized to view each piece of content, including providers and portlets. If you do not belong to a group that has been granted a specific privilege, OracleAS Portal prevents you from performing the actions associated with that privilege.

ACLs are managed by the following:

- Privileges define the actions that can be performed on the object to which they are granted. Several privileges can be granted, such as Manage, Execute, Access, and Publish. If you set any of these privileges, then the user can access the portlet.
- Users and their privileges are managed from the **Portal** tab under the **Administer** tab of the builder.
- Group membership in a group and the privileges granted to the group are managed from the **Portal** tab under the **Administer** tab of the builder. A privilege granted to a user group is inherited by all members of that group.
- Privileges can be granted to a provider. By default, those privileges apply to the provider and all the portlets in the provider. Provider ACLs are managed on the Provider tab of the navigator.

- Privileges for portlets can override the privilege set for the portlet's provider. Portlet ACLs are managed on the **Provider** tab of the navigator. Using Open for the Provider takes you to a page to manage the portlets of the provider.

6.1.7.4.1 Advantages

- ACLs offer a simple, yet very powerful, mechanism to secure portal objects.
- Because the management of users and groups is centralized, you do not have to change the ACLs as the membership of groups changes.

6.1.7.4.2 Disadvantages

ACLs are applied at the provider or portlet level. You cannot vary the security rules for a portlet depending on the portal page on which the portlet is placed.

6.1.7.5 OracleAS Portal Server Authentication

One way you can prevent unauthorized access to providers is to restrict access to the provider to known client computers at the server level. This method goes some way toward defending against denial of service attacks.

In the Oracle HTTP Server, you can permit or deny directives in the `httpd.conf` file based on hostnames or IP addresses. If hostnames are used as discriminators, the server needs to look them up on its Domain Name Server, which incurs overhead to the processing of each request. Using the IP address prevents this added overhead, but the IP address may change without warning.

6.1.7.5.1 Advantages

- This approach only allows trusted hosts to access the provider.
- You can set the restrictions up easily.

6.1.7.5.2 Disadvantages

- OracleAS Web Cache does not have IP address checking capability. If you have OracleAS Web Cache in front of a provider, a client on any host can send show requests to OracleAS Web Cache.
- You can circumvent this approach by sending messages to a provider containing fake IP addresses and hostnames. This method is tricky to carry out effectively because return messages will go to the computer with the copied IP address, but it can still cause problems.

The following sections are applicable for Web providers *only* and *not* WSRP producers.

6.1.7.6 Securing the Portal Tools Provider Configuration Pages

Out of the box, the Portal Tools (OmniPortlet and Web Clipping) provider configuration pages are protected with certain privileges. Refer to Section 6.1.3.4, "Privileges to Create and Edit Web Providers and Provider Groups" for more information about these privileges. In the event that the pages are no longer protected, check in the `web.xml` file under the *`ORACLE_HOME`*`/j2ee/OC4J_Portal/applications/portalTools/providerBuilder/WEB-INF` directory, that the following param-value is set to `true` and not `false`:

```
<init-param>
<param-name>oracle.webdb.providerui.securedAccessParam</param-name>
<param-value>true</param-value>
</init-param>
```

6.1.7.7 Single Sign-On

Portlets act as windows into an application by displaying a summary of content and a method for accessing the full functionality of the application. Portlets can expose application functionality directly in the portal or provide deep links into the application itself to perform a task.

If the application does not perform authorization, then users need not be authenticated to see and use it or its associated portlets. For more restricted applications, you need to authenticate the user who is accessing the application:

- Partner applications share the same authenticated user as the OracleAS Portal user. The application user and the OracleAS Portal user are the same in this case.
- External applications use a different authentication mechanism than OracleAS Portal and usually a different repository for user credentials. The application user name can be the same as in OracleAS Portal, but the external application verifies the user through its own mechanism.

6.1.7.7.1 Partner Application A partner application shares the same OracleAS Single Sign-On as OracleAS Portal for authentication. Sharing OracleAS Single Sign-On instances means that when a user is already logged into OracleAS Portal, their identity can be asserted to the partner application without logging in again.

Partner applications tightly integrate with OracleAS Single Sign-On. When a user attempts to access a partner application, the partner application delegates the authentication of the user to OracleAS Single Sign-On. Once authenticated with a valid user name and password, a user need not provide a user name or password when accessing other partner applications that share the same OracleAS Single Sign-On instance. OracleAS Single Sign-On determines that the user was successfully authenticated and indicates successful authentication to the other partner applications.

The partner application provider trusts OracleAS Portal to authenticate the user on the provider's behalf. This relationship is possible because OracleAS Portal is, itself, a partner application. Partner application providers must trust OracleAS Portal to authenticate the user in this way because the provider cannot perform the authentication itself. Authenticating the user directly requires the provider to redirect the browser to OracleAS Single Sign-On and provide success and failure URLs. This method is not possible due to the provider architecture. The primary reason for it is that the authentication occurs in response to an API call from OracleAS Portal to the provider. OracleAS Single Sign-On cannot imitate that call upon successful authentication to the `initSession()/dologin()` method to complete its normal processing.

Authentication of users in partner applications differ from conventional applications. Partner applications delegate user authentication to OracleAS Single Sign-On. If the user has not been authenticated, OracleAS Single Sign-On displays a login page prompting the user to enter a user name and password. The login page submits the user name and password back to OracleAS Single Sign-On.

If successfully authenticated, OracleAS Single Sign-On creates a special cookie containing information about the user. For security, OracleAS Single Sign-On encrypts the contents of the cookie. The cookie is sent back to the user's browser but is scoped such that only OracleAS Single Sign-On can access it. It is not passed to any other listeners. After creating the cookie, OracleAS Single Sign-On redirects the Web browser to the success URL specified by the partner application. At this point, the partner application creates an application session cookie which contains information the application needs to reestablish the session later. Upon making subsequent requests to

the partner application, it detects the presence of the partner application session cookie and from it knows that the user is already authenticated.

If the user later accesses another partner application, that application looks for its application specific session cookie. If the cookie is not found, the application redirects the request to OracleAS Single Sign-On as described previously. This time OracleAS Single Sign-On detects the presence of the user's OracleAS Single Sign-On cookie. This cookie indicates that the user is already authenticated and OracleAS Single Sign-On redirects the browser to the success URL of the second partner application without prompting the user for credentials again. At this point, the partner application creates its own application specific session cookie.

In order to protect the integrity of the identity assertion made by OracleAS Portal to the provider, message authentication with HMAC should be configured. See Section 6.3.1.3, "Configuring Provider Message Authentication" for information on how to set this up.

Advantages

- Provides the tightest integration with OracleAS Portal and OracleAS Single Sign-On.
- Provides the best OracleAS Single Sign-On experience to users.
- Provides the most secure form of integration because user names and passwords are not transmitted between the portal and the provider.
- The application and the portal share the same user repository, which reduces user maintenance.

Disadvantages

- The application must share the same user repository as OracleAS Portal even though the application's user community may be a subset of the portal's user community. This minor issue can be addressed because you can restrict access to the portal pages that expose the application to the application's user community.
- The application can only be tightly integrated with one or more OracleAS Single Sign-On if they share the same user repository.

Implementation Techniques

You make an application a partner application by protecting its URLs using mod_osso. Once configured, mod_osso restricts access to URLs and handles such things as the redirection to OracleAS Single Sign-On and the creation of cookies.

mod_osso

mod_osso is a general purpose Oracle HTTP Server module and a partner application of OracleAS Single Sign-On. It uses OracleAS Single Sign-On to do the authentication. The module does all the communication and handling of cookies between the Oracle HTTP Server and OracleAS Single Sign-On. If mod_osso is configured to protect the URLs of a Web application, then the application effectively becomes a partner application.

OracleAS Portal is also a partner application and uses OracleAS Single Sign-On to authenticate users. Provided OracleAS Portal and mod_osso use the same OracleAS Single Sign-On instance, the user can access either the Web application or OracleAS Portal by logging in to either one, that is, they need only login once to be able to access both the Web application and OracleAS Portal.

Advantages

- mod_osso is simple to set up.
- You need no additional code in the application.
- New features to the OracleAS Single Sign-On environment are exposed through simple dynamic directives.
- mod_osso generates a partner application cookie and does all the cookie handling.
- mod_osso secures the partner application and deep links from the partner application provider.

Disadvantages

- Although not neccessarily a drawback, mod_osso can only be used with Web applications.

6.1.7.7.2 External Application An External Application is an application that uses a different authentication mechanism than OracleAS Portal. The application may use a different instance of OracleAS Single Sign-On than that used by OracleAS Portal or some other authentication method. In either case, OracleAS Single Sign-On stores user name mappings, passwords, and any other required credentials to authenticate the user in each external application. When a user is already logged in to OracleAS Portal, they will be logged into the external application without having to enter their user name or password.

Applications that manage their own authentication of users can be loosely integrated with OracleAS Single Sign-On by registering as external applications. An external application can be exposed as a provider using the Oracle Application Server Portal Developer Kit so that it may be accessed from a portlet on a page. External application providers are only available to JPDK Web providers.

> **See Also:** For more information about the External Applications portlet, see the *Oracle Application Server Portal User's Guide*.

When a previously authenticated user accesses an external application for the first time, OracleAS Single Sign-On attempts to authenticate the user with the external application. The authentication is performed by submitting an HTTP request that combines the registration information and the user's user name and password for the application. If the user has not yet registered their user name and password for the external application, OracleAS Single Sign-On prompts the user for the required information before making the authentication request. When a user supplies a user name and password for an external application, OracleAS Single Sign-On maps the new user name and password to the user's OracleAS Portal user name and stores them. The next time the user needs to access the external application the stored credentials are used.

Note: If there is a change in the URL of an external application, then the external application must be updated in the OracleAS Single Sign-On Server. For information about updating the external application, refer to the "Editing an External Application" section in the *Oracle Application Server Single Sign-On Administrator's Guide*.

Advantages

- Allows integration with many portals. If there is a preferred portal, the application could be integrated as a partner application of that portal and an external application of other portals.
- Provides a single sign-on experience for users. However, users still need to maintain different user names and passwords. In addition, the external application user name mapping must be maintained.
- Allows integration with multiple portals independent of their user repositories and single sign-on mechanisms.

Disadvantages

- External applications do not share the same user repository as the portal, which requires additional maintenance of user information.
- The user name and password is transmitted to the provider in plain text. This approach is not as secure as a partner application. Configuring the provider URL to use SSL addresses this issue.
- The application must be written using a technology that can be easily integrated with Java or PL/SQL.

6.1.7.7.3 No Application Authentication In this case, the provider trusts the portal sending the requests. The provider can determine if the user is logged in and the portal user name, but the application has not authenticated the user.

Advantages

You can implement this form of integration most easily and very fast.

Disadvantages

Provides the weakest integration with OracleAS Portal. However, this may not be an issue if your portlet content is not sensitive, or if the provider location is secured by the network topology and only accessible by the portal.

6.1.7.8 Programmatic Portlet Security

You can implement portlet security methods within a provider to verify that given users may access portlet instances. These security methods work at the portlet level, that is, each portlet may have its own user access control. By implementing access control methods in the provider, content may only be retrieved from a portlet if the user's credentials pass the authorization logic. If you do not implement portlet security methods in the provider, any user name may be passed in, even a fictitious, unauthenticated one.

A provider can implement two portlet security methods:

- Get a list of portlets
- Determine portlet accessibility

These methods have access to the following information about authorization level:

- **Strong** indicates that OracleAS Single Sign-On has authenticated a user in the current OracleAS Portal session, that is, the user logged in to OracleAS Portal with a valid user name and password, and called the portlet in that session.

- **Public** indicates a user has not logged in within the context of the current OracleAS Portal session and does not have a persistent cookie to indicate that such a state previously existed.

Portlets can also access the OracleAS Portal user privileges and group memberships:

- User's default group
- User or group privileges
- User's highest available privilege across all groups
- Objects a user can access

6.1.7.8.1 Advantages

With portlet security methods, you can have a portlet produce different output depending on the user's level of authorization.

6.1.7.8.2 Disadvantages

Most security manager implementations use the authorization level or some other user specific element in an incoming message. A check of this type could be bypassed by an entity imitating OracleAS Portal.

6.1.7.9 Message Authentication

The Oracle Application Server Portal Developer Kit supports message authentication to limit access to a specified provider instance or group of provider instances. A provider is registered with a secret shared key known only to the portal and provider administrators.

An OracleAS Portal instance sends a digital signature, calculated using a Hashed Message Authentication Code (HMAC) algorithm, with each message to a provider. A provider may authenticate the message by checking the signature with its own copy of the shared key. This technique may be used in SSL communication with a provider instead of client certificates.

An OracleAS Portal instance calculates a signature based on user information, a shared key, and a time stamp. The signature and time stamp are sent as part of the SOAP message. The time stamp is based on UTC (coordinated universal time, the scientific name for Greenwich Mean Time) so that time stamps can be used in messages between computers in different time zones.

When the provider receives this message it will generate its own copy of the signature. If the signatures agree, it will then compare the message time stamp with the current time. If the difference between the two is within an acceptable value the message is considered authentic and processed accordingly.

A single provider instance cannot support more than one shared key. Multiple keys could cause security and administration problems if several clients sharing a provider use the same key. For instance, if one copy of the shared key is compromised in some way, the provider administrator has to create a new key, distribute it to all of the clients, and the clients must then update their provider definition. The way around this issue is to deploy different provider services, specifying a unique shared key for each service. Each provider service has its own deployment properties file so that each service is configured independently of the others. The overhead of deploying multiple provider services within the same provider adapter is relatively small.

If a provider does not have an OracleAS Web Cache in front of it, the use of the same signature cookie over the lifetime of a provider session means you must trade off between performance and the security provided by authenticating the requests. The

signature cookie value is calculated only once after the initial SOAP request establishes the session with the provider. The shorter the provider session timeout, the more often a signature will be calculated to provide greater security against an illegal show request. However, the SOAP request required to establish a session takes more time.

In a provider that uses OracleAS Web Cache to cache show request responses, you make a similar trade-off. Cached content is secured in the sense that incoming requests must include the signature cookie to retrieve the cached content, but caching content for an extended period of time leaves the provider open to illegal show requests.

The signature element provides protection against interception and the resending of messages, but it does nothing to prevent the interception and reading of message contents. Messages are still transmitted in plain text. If you are concerned about the content of messages being read by unauthorized people, you should use message authentication in conjunction with SSL.

> **See Also:** Section 6.3.1.3, "Configuring Provider Message Authentication"

6.1.7.9.1 Advantages

Message authentication ensures that the message received by a provider comes from a legitimate OracleAS Portal instance.

6.1.7.9.2 Disadvantages

- Message authentication can cause administration problems if a provider serves more than one OracleAS Portal instance.
- Message authentication has a performance implication if made very secure by having a short session timeout.

6.1.7.10 HTTPS Communication

Normal communication between OracleAS Portal and a provider uses HTTP, a network protocol that transmits data as plain text using TCP as the transport layer. An unauthorized agent can read an intercepted message. HTTPS uses an extra security layer (SSL) on top of TCP to secure communication between a client and a server.

Each entity (for example, an OracleAS Web Cache instance) that receives a communication using SSL has a freely available public key and a private key known only to the entity itself. Any messages sent to an entity are encrypted with its public key. A message encrypted by the public key may only be decrypted by the private key so that even if a message is intercepted it cannot be decrypted.

Certificates are used to sign communications, thereby ensuring that the public key does in fact belong to the correct entity. These certificates are issued by trusted third parties known as Certification Authorities (CA), for example, OracleAS Certificate Authority or Verisign. They contain an entity's name, public key, and other security credentials. They are installed on the server end of an SSL communication to verify the identity of the server. Client certificates may also be installed on the client to verify the identity of a client, but this feature is not yet supported OracleAS Portal. Message authentication may be used instead.

Oracle Wallet Manager is the application used to manage public key security credentials. It is used to generate public/private key pairs, create a certificate request to a CA, and then install the certificate on a server.

6.1.7.11 Configuration of SSL

When a provider is registered from an OracleAS Portal instance, only one URL is entered. HTTP or HTTPS may be used, but not a combination of both.

A server side certificate that is installed on a computer identifies that computer, or the domain, and may be used by any number of port definitions on that computer. A certificate trust list ensures that communication is limited to specifically identified servers. Message authentication should be used as well to fully secure communication between a trusted OracleAS Portal instance and a provider.

See Also:

- Section 6.3.2.1, "Configuring SSL for OracleAS Portal"
- *Oracle Internet Directory Administrator's Guide*
- *Oracle Application Server Web Cache Administrator's Guide*

6.1.7.11.1 Advantages

- SSL encrypts the contents of a portlet during the transmission of the data from the provider to the Parallel Page Engine. To further secure the portlet contents, the surrounding page should be invoked by a SSL based request.

6.1.7.11.2 Disadvantages

- Encryption has a performance impact on OracleAS Portal.
- If used, encryption requires all portlets from a provider to use HTTPS even if some of the content is public.

You will find additional information on security for your Web providers in the papers:

- *Overview of Provider Security*
- *Overview of Password Authenticated Applications*

on the Oracle Technology Network (OTN), http://www.oracle.com/technology/.

6.1.8 Securing the OmniPortlet and Simple Parameter Form

The OmniPortlet and simple parameter form are located under Portlet Builders in the Portlet Repository. By default, any user who has the privilege to create pages can add these portlets to a page and define them. Furthermore, a user who has at the minimum **Manage Content** privileges on the page can define these portlets by clicking the **Define OmniPortlet** or **Define Simple Parameter Form**.

To control this kind of access, you can activate a privilege check. Once you perform the procedure that follows, the display of these portlets depends upon the privileges granted to the user or user group from the **Access** tab. To perform any operations on the portlet, the user or user group needs at least the Execute privilege.

1. Log in to OracleAS Portal.
2. Click the **Navigator** link.
3. Click the **Portlet Repository** page group.
4. Click **Pages**.
5. Next to the **Portlet Builders** page, click **Edit**.
6. Click Page: **Access** in the upper left of the page.

7. Select **Enable Item Level Security**.
8. Click **OK**.
9. Click the **Edit Item** icon next to **OmniPortlet**.
10. Click the **Access** tab.
11. Check **Define Portlet Access Privileges**.
12. Click **Apply** and note the appearance of the Grant Access and Change Access sections of the page.
13. Use the **Grant Access** section to assign privileges to users and groups as desired.
14. Click **OK**.
15. Repeat steps 9 through 14 for the **Simple Parameter Form**.

6.1.9 Securing the Web Clipping Provider

Appendix I, "Configuring the Portal Tools Providers" describes the administrative tasks that must be performed before you are able to use the Web Clipping provider. The following sections describe some security configuration options that you should implement to enable the Web Clipping provider to access trusted sites and encrypt the channel between itself and the database:

- Adding Certificates for Trusted Sites
- Configuring Oracle Advanced Security for the Web Clipping Provider

6.1.9.1 Adding Certificates for Trusted Sites

When a user navigates to a secure site, the Web site typically returns a certificate, identifying itself to the user when asking for secure information. If the user accepts the certificate, the certificate is placed into the list of trusted certificates of the browser so that a secure channel can be opened between the browser and that server. Like a Web browser, the Web Clipping provider behaves as an HTTP client to external Web sites. In order for the Web Clipping provider to keep track of trusted sites, it makes use of a file that stores the certificates of those sites, namely the `ca-bundle.crt` file, located in the *`MID_TIER_ORACLE_HOME`*`/portal/conf` directory.

The shipped `ca-bundle.crt` is an exported version of the trusted server certificate file from Oracle Wallet Manager. The default trusted server certificate in Oracle Wallet Manager does not cover all possible server certificates that exist on the Web. For this reason, when a user navigates to a secure server using HTTPS, the user may get an SSL Hand-shake failed exception in the Web Clipping Studio. To solve this problem, the `ca-bundle.crt` file needs to be augmented with new trusted sites that are visited. As a portal administrator, you must do the following to extend the shipped `ca-bundle.crt` file:

1. Use a browser (preferably Internet Explorer) to download the root server certificate from each external HTTPS Web site in BASE64 format that is visited, and is missing from the trusted certificate file.
2. Use Oracle Wallet Manager to import each certificate.
3. Export the trusted server certificates into a file and replace the `ca-bundle.crt` file with that file.

For more information about Oracle Wallet Manager, see the *Oracle Database Advanced Security Administrator's Guide* in the Oracle Database documentation on OTN, `http://www.oracle.com/technology/`.

6.1.9.2 Configuring Oracle Advanced Security for the Web Clipping Provider

The Web Clipping provider can use Oracle Advanced Security Option (ASO) to secure and encrypt the channel between itself (on the middle tier) and the database that hosts the Web Clipping Repository. As ASO is a feature available only on Oracle Application Server Enterprise Edition, or as an add-on option to the Standard Edition, this feature is disabled by default. To enable it, perform the following steps:

1. Go to the Web Clipping provider test page at:

   `http://<host>:<port>/portalTools/webClipping/providers/webClipping`

2. Under the **Provider Configuration** section, in the **Setting** column, there is a **Web Clipping Repository** field. Click its corresponding **Edit** link in the **Actions** column.
3. In the **Repository Settings** section of the **Edit Provider: webClipping** page, select the **enable (secure database connections)** option in the **Advanced Security Option** field.
4. Select **OK** to save the settings and return to Web Clipping provider test page.

In addition, you must set the following ASO configuration parameters in the `sqlnet.ora` file to ensure that the database connections created between the Web Clipping provider and the database hosting the Web Clipping Repository are using ASO. See the *Oracle Advanced Security Administrator's Guide* for the list of values to use as all possible combinations of parameters are described in detail.

- `SQLNET.AUTHENTICATION_SERVICES` -- This parameter is used to select a supported authentication method in making database connections with ASO. See the *Oracle Advanced Security Administrator's Guide* for more information about setting this parameter.
- `SQLNET.CRYPTO_SEED` -- This parameter denotes the cryptographic seed value (FIPS 140-1 setting), used in making database connections with ASO.

  See the *Oracle Advanced Security Administrator's Guide* for more information about setting this parameter.

> **Note:** When setting these parameters after the initial configuration (where the database parameters are already set up), the database connections are assumed to be open already. Because enabling ASO affects all connections made to the database, it is advisable to restart the OC4J instance containing the Web Clipping provider to reset all the current connections to now use ASO. You would also need to do this when disabling ASO.

6.1.10 Securing the Federated Portal Adapter

The Federated Portal Adapter is a component of OracleAS Portal that allows portal instances to share their portlets through the Web portlet interface. For example, suppose that a user displays a page in one portal instance that contains a portlet whose source resides on another portal instance. When the Federated Portal Adapter on the remote portal receives the request for the portlet, it starts a session for the user on the remote portal. The portlet can then be run from the remote portal instance by the user. This scenario has a couple of security implications:

- Because the Federated Portal Adapter must create a session for the user on the remote portal, it would be best for the two portal instances to share the same

single sign-on server. Otherwise, name collisions could occur when the Federated Portal Adapter attempts to log the user onto the remote portal instance.

- Because the Federated Portal Adapter creates private portal sessions based on SOAP messages it receives, it is a potential security risk. A message authentication code must be used to ensure that any messages received by the Federated Portal Adapter emanate from a trusted source.

See Also: Chapter 11, "Using the Federated Portal Adapter"

You will find additional information in the article "How to Add Remote Portlets Using the Federated Portal Adapter," on OTN, http://www.oracle.com/technology/.

6.1.11 Securing OraDAV

WebDAV (World Wide Web Distributed Authoring and Versioning) is the IETF's standard for collaborative authoring on the Web. It defines a set of extensions to HTTP that facilitates collaborative editing and file management between users located remotely from each other on the Internet.

OraDAV, Oracle's implementation of WebDAV, is the mechanism used by the Oracle HTTP Server to communicate with WebDAV clients. OraDAV enables your users to connect to OracleAS Portal using their WebDAV clients. In terms of security, accessing OracleAS Portal through WebDAV presents two security issues for you to consider:

- Expiration of OracleAS Portal session cookies for OraDAV
- SSL and OraDAV

6.1.11.1 Session Cookie Expiration

The OraDAV configuration parameter, `ORACookieMaxAge`, specifies, in seconds, the length of time for which the DAV client should retain the cookie. The default value is 28800 (that is, 8 hours).

`ORACookieMaxAge` can be changed in Oracle Enterprise Manager or by directly editing it in the `oradav.conf` file located in *`MID_TIER_ORACLE_HOME`*`/Apache/oradav/conf`. If you choose to manually change the file, you must synchronize the changes with Dynamic Configuration Management. Once the change has been made in the configuration file, you need to restart the Oracle HTTP Server to have the changes take effect in the runtime system:

```
cd MID_TIER_ORACLE_HOME/dcm/bin
./dcmctl shell
- dcmctl> updateConfig -ct ohs
```

After you exit the `dcmctl` shell, execute the following command from *`MID_TIER_ORACLE_HOME`*`\opmn\bin` to restart the Oracle HTTP Server:

```
opmnctl restartproc type=ohs
```

Note: Not all WebDAV clients use cookies. Some perform their authentication on each request using HTTP basic authentication. A client may choose to record the user name and password for the duration of that WebDAV client session and thus only need to prompt the user once for their credentials. In either case, though, this behavior results in a slower response time from OracleAS Portal because every request from such clients must be authenticated, requiring added communication with the Oracle Internet Directory.

See Also: *Oracle HTTP Server Administrator's Guide*

6.1.11.2 SSL and OraDAV

The use of SSL for WebDAV communication is supported with OraDAV.

6.2 Configuring OracleAS Security Framework for OracleAS Portal

This section describes considerations for:

- Configuring OracleAS Security Framework Options for OracleAS Portal
- Configuring Oracle Identity Management Options for OracleAS Portal

6.2.1 Configuring OracleAS Security Framework Options for OracleAS Portal

For OracleAS Portal, the main consideration when configuring the OracleAS Security Framework is how to properly configure SSL. Refer to Section 6.3.2.1, "Configuring SSL for OracleAS Portal" for a full description of SSL configuration for OracleAS Portal.

6.2.2 Configuring Oracle Identity Management Options for OracleAS Portal

As you configure OracleAS Portal for security, you should consider the following topics related Oracle Identity Management:

- Setting the Appropriate Naming and Nickname Attributes
- Configuring the Portal Administrator for Single Sign-On Administration

6.2.2.1 Setting the Appropriate Naming and Nickname Attributes

When deciding on the Directory Information Tree structure and the setting of the Oracle Context parameters for your Oracle Identity Management Infrastructure, you should consider making the naming attribute different from the nickname attribute. The naming attribute is used for the first attribute in the entry's Distinguished Name. By contrast, the nickname attribute holds the OracleAS Single Sign-On user name.

For OracleAS Portal to properly support renaming users by changing the value of the nickname attribute in the Oracle Internet Directory, the nickname attribute must be different than the naming attribute. By keeping the two separate, the Distinguished Name of the user's entry in the Oracle Internet Directory remains unchanged even when the value of the nickname attribute changes.

See Also: *Oracle Identity Management Concepts and Deployment Planning Guide*

6.2.2.2 Configuring the Portal Administrator for Single Sign-On Administration

In previous releases of OracleAS Portal, the super user, PORTAL, was able to perform OracleAS Single Sign-On administration. With OracleAS Portal Release 9.0.4, the ability to perform OracleAS Single Sign-On administration out of the box is removed. The rationale for this change is that in enterprise settings it is not necessarily appropriate for an OracleAS Portal administrator to have permissions to perform Oracle Internet Directory and OracleAS Single Sign-On administration. Much like the discussion in the previous section, regarding the roles of the centralized Oracle Identity Management Infrastructure administrator and the departmental OracleAS Portal administrator, it may be inappropriate for the OracleAS Portal administrator to have the permissions to perform OracleAS Single Sign-On administration.

If you need to allow the OracleAS Portal account to perform OracleAS Single Sign-On administration, you need to explicitly give the user the privilege.

See Also:

- *Oracle Identity Management Concepts and Deployment Planning Guide*
- *Oracle Application Server Single Sign-On Administrator's Guide*

6.3 Configuring OracleAS Portal Security

This section describes configuration considerations for OracleAS Portal.

- Configuring OracleAS Portal Security Options
- Configuring Options for OracleAS Security Framework
- Configuring OracleAS Portal Options for Database Security

6.3.1 Configuring OracleAS Portal Security Options

This section contains the following subsections:

- Changing Settings on the Global Settings Page
- Enforcing Role-Based Access Control
- Configuring Provider Message Authentication

6.3.1.1 Changing Settings on the Global Settings Page

Once you have installed OracleAS Portal and performed the appropriate tasks from Section 6.3.2.3, "Post-Installation Security Checklist", you can change all of the following settings that pertain to security from the **Global Settings** page of OracleAS Portal:

- Cache for Oracle Internet Directory Parameters
- Oracle Directory Integration Platform Synchronization
- Group Search Base Distinguished Name (DN)
- Group Creation Base Distinguished Name (DN)

6.3.1.1.1 Cache for Oracle Internet Directory Parameters As pointed out in Section 6.1.6, "Leveraging Oracle Identity Management Infrastructure", OracleAS Portal maintains a cache of information from the directory. From the **Global Settings** page, you can refresh this cache with the updated information from the directory. **Refresh Cache for OID Parameters** immediately updates the cache with the latest parameters values

from the directory. The cached information is relatively static information, hence you do not need to refresh the cache frequently.

6.3.1.1.2 Oracle Directory Integration Platform Synchronization Because OracleAS Portal caches group membership information, it requires a mechanism for updating the cache when the information is changed in the directory. The directory integration server notifies OracleAS Portal whenever a change is made in the directory that must be reflected in OracleAS Portal. In **Global Settings**, you can set:

- **Enable directory synchronization** defines whether the directory integration server notifies OracleAS Portal when a relevant change is made in the directory. If this setting is not checked, then OracleAS Portal will not be notified of any directory integration server subscribed events.
- **Send event notifications every n seconds** defines the interval of time between event notifications sent by the directory integration server to notify OracleAS Portal of relevant changes. This setting is available only when Enable directory synchronization is checked.

When the Oracle Directory Integration and Provisioning server is running and configured to work with OracleAS Portal, group membership changes in Oracle Internet Directory will result in soft cache invalidations in OracleAS Portal.

See Also:

- Section 1.3.3, "Understanding Cache Invalidation in OracleAS Portal"
- Section 5.8, "Managing OracleAS Portal Content Cached in OracleAS Web Cache"
- Section 6.1.6.3, "Relationship Between OracleAS Portal and Oracle Directory Integration Platform"

Some examples of group membership cache invalidations are:

- If you add a user to a group, the Oracle Directory Integration and Provisioning server notifies OracleAS Portal of the change. OracleAS Portal will then issue a single soft invalidation message that will be processed by the soft invalidation job. This is because of the possibility that the user may have new privileges that can affect what data can be viewed.
- If you add group *_A* to group *_B*, the Oracle Directory Integration and Provisioning server notifies OracleAS Portal of the change. OracleAS Portal will then issue a soft invalidation message for each user in group *_A*. This is because of the possibility that the users in group *_A* may have new privileges that affect what data they can view.

6.3.1.1.3 Group Creation Base Distinguished Name (DN) OracleAS Portal maintains its user group information in the directory. When groups are created through the Group portlet, they are created under a node of the LDAP Directory Information Tree (DIT). A node is identified by its distinguished name (DN). Therefore, in OracleAS Portal, you need to specify in which node you wish to create groups:

Group Creation Base DN is the DN of the node in which you want OracleAS Portal to maintain its user groups. For example:

```
cn=portal.040820.123756.096286000,cn=Groups,dc=MyCompany,dc=com
```

This setting is particularly useful if you adapt OracleAS Portal to interact with an existing DIT.

6.3.1.1.4 Group Search Base Distinguished Name (DN) Just as you need to define the node in which you want to create groups, you must also define the node in which you want OracleAS Portal to search for existing groups. For example, you need to specify where OracleAS Portal searches when it displays the group's list of values in the **Group** portlet.

Local Group Search Base DN is the DN of the node in which you want OracleAS Portal to maintain its user groups. For example:

```
cn=portal.040820.123756.096286000,cn=Groups,dc=MyCompany,dc=com
```

This setting is particularly useful if you adapt OracleAS Portal to interact with an existing DIT.

6.3.1.2 Enforcing Role-Based Access Control

From OracleAS Portal 10*g* Release 2 (10.1.2) onwards, it is possible to enable or disable enforcement of Role-Based Access Control. This option is disabled in the default configuration. Role-Based Access Control can prevent the assignment of both object level privileges and global privileges directly to users from the OracleAS Portal User Interface and forces them to be granted only to groups. However, this option does not have any impact on the privileges granted directly to users:

- Before Role-based Access Control was enabled
- Automatically when an object is created
- Through the use of the OracleAS Portal APIs

To enable or disable Role-based Access Control, you must run the script `secrlacl.sql` located in the directory *`ORACLE_HOME`*`/portal/admin/plsql/wwc`. The syntax for `secrlacl.sql` is:

```
@secrlacl.sql Y|N
```

For example, if you want to enable Role-based Access Control, run the script as follows:

```
@secrlacl.sql Y
```

Once Role-based Access Control is enabled, you will see the following changes in OracleAS Portal:

- Access Tab for Objects. Here you will see only the Group LOV; the User LOV does not appear.
- The Privilege tab is not rendered on the Edit User Profile page.

6.3.1.3 Configuring Provider Message Authentication

Additional configuration is required to set up a provider service that expects HMAC-enabled communication. You can set up basic or enhanced authentication.

6.3.1.3.1 Basic Authentication If your JPDK provider code is accepting the portal user's identity through the `oracle.portal.provider.v2.ProviderUser` from the `oracle.portal.provider.v2.render.PortletRenderRequest` object, and using this identity for any sensitive operations, you should configure the portal and this provider for basic message authentication. Basic message authentication utilizes a Hashed Message Authentication Code (HMAC), which is a mechanism for message

authentication using cryptographic hash functions, to ensure the integrity of the message.

To configure basic authentication using HMAC, you need to configure both the JPDK Web provider and OracleAS Portal as follows:

JPDK Web Provider

For the JPDK Web provider, add a shared key as a Web provider JNDI environment variable to the `web.xml` file. The exact position of environment entries in `web.xml` is toward the end, as shown in bold in Table 6–17, which shows the relative order of the elements in `web.xml`.

Table 6–17 Relative Order of the Elements In web.xml

Element Name
icon?
display-name?
description?
distributable?
context-param*
filter*
filter-mapping*
listener*
servlet*
servlet-mapping*
session-config?
mime-mapping*
welcome-file-list?
error-page*
taglib*
resource-env-ref*
resource-ref*
security-constraint*
login-config?
security-role*
env-entry*
ejb-ref*
ejb-local-ref*

Add a JNDI environment variable definition to the `web.xml` file, by adding the following `env-entry` element in the appropriate location:

```
<env-entry>
   <env-entry-name>oracle/portal/provider/service_name/sharedKey</env-entry-name>
   <env-entry-value>shared_key_value</env-entry-value>
   <env-entry-type>java.lang.String</env-entry-type>
</env-entry>
```

Enter the service name and the shared key value, as shown in Example 6–1.

Example 6–1 Adding a JNDI Environment Variable Definition to web.xml

```
<env-entry>
   <env-entry-name>oracle/portal/provider/sample/sharedKey</env-entry-name>
   <env-entry-value>1234567890abcdeFGHIJ</env-entry-value>
   <env-entry-type>java.lang.String</env-entry-type>
</env-entry>
```

In this example, `sample` is the service name for the provider and `1234567890abcdeFGHIJ` is the shared key value. The value of the shared secret key used for the HMAC computation must contain between 10 and 20 alphanumeric characters.

You must define this environment variable for each provider instance that you want to secure, as shown in Example 6–2.

Example 6–2 Defining JNDI Environment Variables for Multiple Provider Instances in web.xml

```
<env-entry>
   <env-entry-name>oracle/portal/provider/provider0/sharedKey</env-entry-name>
   <env-entry-value>1234567890abcdeFGHIJ</env-entry-value>
   <env-entry-type>java.lang.String</env-entry-type>
</env-entry>

<env-entry>
   <env-entry-name>oracle/portal/provider/provider1/sharedKey</env-entry-name>
   <env-entry-value>0987654321abcdeFGHIJ</env-entry-value>
   <env-entry-type>java.lang.String</env-entry-type>
</env-entry>

<env-entry>
   <env-entry-name>oracle/portal/provider/provider2/sharedKey</env-entry-name>
   <env-entry-value>123123123123</env-entry-value>
   <env-entry-type>java.lang.String</env-entry-type>
</env-entry>
```

Alternatively, you can add the provider property `sharedKey` in the `.properties` file. To do this, perform the following steps:

1. Open the file `<app_root>/WEB-INF/deployment/`*`service_name`*`.properties`. (Substitute *`service_name`* with the name of your provider service instance.)
2. Add the provider property `sharedKey` and the value for your shared key, as shown in the following example:

   ```
   sharedKey=1234567890abcdeFGHIJ
   ```

   You must set this property in each of the `service_name.properties` files for each provider instance that you want to secure.

Note: The disadvantage of this alternate approach is that you cannot manage the environment variables for the provider using Application Server Control Console, as you would with JNDI environment entries.

OracleAS Portal

In OracleAS Portal, register the provider with the shared key and login frequency settings, as follows:

1. In the **Portal Builder**, click the **Administer** tab, then click the **Portlets** subtab.
2. In the **Remote Providers** portlet, click **Register a Provider**.
3. In the **Register a Provider** page, enter the provider details, and click **Next**.
4. In the **General Properties** section, for **Shared Key**, enter the value of the secret key.
5. In the **User/Session Information** section, for **Login Frequency**, select **Once Per User Session**.
6. Follow the instructions in the wizard and click **Finish**.

6.3.1.3.2 Enhanced Authentication If your JPDK Web provider code is running in an OC4J container configured for JAZN-LDAP, and the provider code uses J2EE security or obtains the user's identity through the `getRemoteUser()` or `getUserPrincipal()` methods of the `HttpServletRequest` object, you should configure the portal and this provider for enhanced message authentication. You should also configure enhanced message authentication if you are using the LDAP (Oracle Internet Directory) Security features in your provider, as documented in the *Oracle Application Server Portal Developer's Guide*. Enhanced message authentication secures the integrity of the additional headers that are used to propagate the user's authenticated identity to the provider.

> **Note:** Enhanced authentication is a new feature in OracleAS Portal 10*g* Release 2 (10.1.4).

To configure enhanced authentication using HMAC, you need to configure both the JPDK Web provider and OracleAS Portal.

JPDK Web Provider

To configure the JPDK Web provider, perform the following steps:

1. For the JPDK Web provider, add a shared key as a Web provider JNDI environment variable to the `web.xml` file. The exact position of environment entries in `web.xml` is toward the end, as shown in Table 6–17, which shows the relative order of the elements in `web.xml`.

   Add a JNDI environment variable definition to the `web.xml` file, by adding the following `env-entry` element in the appropriate location:

   ```
   <env-entry>
      <env-entry-name>oracle/portal/provider/service_
   name/sharedKey</env-entry-name>
      <env-entry-value>shared_key_value</env-entry-value>
      <env-entry-type>java.lang.String</env-entry-type>
   </env-entry>
   ```

   Enter the service name and the shared key value, as shown in Example 6–1.

   In Example 6–1, `sample` is the service name for the provider and `1234567890abcdeFGHIJ` is the shared key value. The value of the shared secret key used for the HMAC computation must contain between 10 and 20 alphanumeric characters.

You must define this environment variable for each provider instance that you want to secure, as shown in Example 6–2.

Alternatively, you can add the provider property `sharedKey` in the `.properties` file. To do this, perform the following steps:

a. Open the file `<app_root>/WEB-INF/deployment/service_name.properties`. (Substitute *service_name* with the name of your provider service instance.)

b. Add the provider property `sharedKey` and the value for your shared key, as shown in the following example:

```
sharedKey=1234567890abcdeFGHIJ
```

You must set this property in each of the `service_name.properties` files for each provider instance that you want to secure.

Note: The disadvantage of this alternate approach is that you cannot manage the environment variables for the provider using Application Server Control Console, as you would with JNDI environment entries.

---

2. Add the provider property `enhancedAuthentication` in the `.properties` file. To do this, perform the following steps:

a. Open the file `<app_root>/WEB-INF/deployment/service_name.properties`. (Substitute *service_name* with the name of your provider service instance.)

b. Add the provider property `enhancedAuthentication` and set it to `true`, as shown in the following example:

```
enhancedAuthentication=true
```

You must set this property in each of the `service_name.properties` files for each provider instance that you want to secure.

OracleAS Portal

In OracleAS Portal, register the provider with the shared key and login frequency settings, as follows:

1. In the **Portal Builder**, click the **Administer** tab, then click the **Portlets** subtab.
2. In the **Remote Providers** portlet, click **Register a Provider**.
3. In the **Register a Provider** page, enter the provider details, and click **Next**.
4. In the **General Properties** section, for **Shared Key**, enter the value of the secret key.
5. In the **User/Session Information** section, for **Login Frequency**, select **Once Per User Session**.
6. Follow the instructions in the wizard and click **Finish**.

6.3.2 Configuring Options for OracleAS Security Framework

When configuring OracleAS Portal, you should consider the following options that leverage the OracleAS Security Framework:

- Configuring SSL for OracleAS Portal

- Securing the Connection to Oracle Internet Directory (Optional)
- Post-Installation Security Checklist

6.3.2.1 Configuring SSL for OracleAS Portal

The sections that follow provide an overview of the most common SSL configurations for OracleAS Portal and describe the procedures necessary to implement them.

- Section 6.3.2.1.1, "Overview of SSL Configurations"
- Section 6.3.2.1.2, "SSL to OracleAS Single Sign-On"
- Section 6.3.2.1.3, "SSL to OracleAS Web Cache"
- Section 6.3.2.1.4, "SSL Throughout OracleAS Portal"
- Section 6.3.2.1.5, "External SSL with Non-SSL Within Oracle Application Server"
- Section 6.3.2.1.6, "Configuring and Registering Web Providers, Provider Groups, and WSRP Producers Exposed Over SSL"

6.3.2.1.1 Overview of SSL Configurations

OracleAS Portal uses a number of different components (such as the Parallel Page Engine, Oracle HTTP Server, and OracleAS Web Cache) each of which may act as a client or server in the HTTP communication. As a result, each component in the Oracle Application Server middle tier may be configured individually for the protocols of HTTPS rather than HTTP.

Interacting with OracleAS Portal involves a number of distinct *network hops*. These hops are as follows:

- Between the client browser and the entry point of the portal environment. That is either:
  - OracleAS Web Cache
  - Network edge hardware device such as a Reverse Proxy or SSL accelerator
- Between OracleAS Web Cache and Oracle HTTP Server of the Oracle Application Server middle tier
- Between the client browser and the Oracle HTTP Server of the OracleAS Single Sign-On or Oracle Internet Directory (or infrastructure) tier
- A loopback between the Parallel Page Engine (PPE) on the middle tier and OracleAS Web Cache or front-end Reverse proxy
- Between the Parallel Page Engine (PPE) and the Remote Web Provider producing Portlet content
- Between OracleAS Portal infrastructure and the Oracle Internet Directory

SSL Usage Restriction

External JavaServer Pages do not work with the Parallel Page Engine in a partial SSL configuration mode. They will work when SSL is used throughout OracleAS Portal. If SSL is implemented externally with a load balancing router, only internal JavaServer Pages will work. As a result, with the SSL configurations described in Section 6.3.2.1.2, "SSL to OracleAS Single Sign-On", Section 6.3.2.1.4, "SSL Throughout OracleAS Portal", and Section 6.3.2.1.5, "External SSL with Non-SSL Within Oracle Application Server", you should only use external JSPs.

Configuring Enterprise Manager Security

In all the SSL configurations discussed here, you need to perform an additional task whenever you use Enterprise Manager to monitor SSL URLs.

Follow the instructions provided in the *Oracle Application Server Administrator's Guide*.

SSL Configuration Tool and Its Limitations

This release introduces the SSL Configuration Tool (`SSLConfigTool`), which helps automate many of the manual steps currently required for configuring SSL. The `SSLConfigTool` executable is located in the *`ORACLE_HOME`*`/bin` directory and its usage syntax is as follows:

```
SSLConfigTool ( -config_w_prompt
              | -config_w_file <input_file_name>
              | -config_w_default
              | -rollback )
              [-dry_run]
              [-wc_for_infra]
              [-secure_admin]
              [-opwd <orcladmin_pwd>]
              [-ptl_dad <dad_name>]
              [-ptl_inv_pwd <ptl_inv_pwd>]
```

See *Oracle Application Server Administrator's Guide* for more details.

Notes:

- The SSL Configuration Tool is available with any Oracle Application Server installation type. OracleAS Infrastructure installations are the only installation type that support SSL configuration during the installation. This option is available on one of the installation screens.
- If you install Oracle Application Server and choose to make some configuration changes before running the SSL Configuration Tool, you should run the tool and then refer to the SSL Configuration Tool log files to verify that your changes were not overwritten. The SSL Configuration Tool creates log files in the directory from which the tool is run. A new log file is created each time the tool is run. For these reasons, it is suggested that you create a separate directory from which you can run the SSL Configuration Tool.

Table 6–18 describes the command-line options for the `SSLConfigTool` command.

Table 6–18 SSL Configuration Tool Command Line Options

Parameter	Description
`-config_w_prompt`	Run in interactive mode.
`-config_w_file <input_file_name>`	Run in silent mode using the values specified in the *`<input_file_name>`* file. This input file should be an XML file. See the *Oracle Application Server Administrator's Guide* for more information.
`-config_w_default`	Run in silent mode using the values specified in the `portlist.ini` and `ias.properties` files.

Table 6–18 (Cont.) SSL Configuration Tool Command Line Options

Parameter	Description
`-rollback`	Revert to the previous state before the command was last run. SSO registration will be done using virtual host and port.
`-dry_run`	Print the steps without implementing them.
`-wc_for_infra`	Forces an OracleAS Web Cache to be used as a load balancer for an infrastructure environment.
`-secure_admin`	Secure the OracleAS Web Cache and Enterprise Manager administration ports (the ports used to display Application Server Control Console).
`-opwd <orcladmin_pwd>`	Set the Oracle administrator password. This parameter is required.
`-ptl_dad <dad-name>`	Set the OracleAS Portal DAD name. If no name is specified, the default "portal" will be used.
`-ptl_inv_pwd <ptl_inv_pwd>`	Set the Portal invalidation password used to send invalidation to OracleAS Web Cache. If you are running `SSLConfigTool` with the `-rollback` parameter, this parameter is not required.

Rolling Back Changes Made by SSLConfigTool

If you have run `SSLConfigTool` in an OracleAS Infrastructure or middle-tier home earlier, you need to first roll back the changes that were made when you last ran the tool, before you can run `SSLConfigTool` again in that Oracle home. To roll back the changes, run `SSLConfigTool` as shown here in an example for Windows:

```
SSLConfigTool.bat -rollback -opwd <orcladmin_pwd>
```

Where:

- `rollback` is used to roll back the changes made by running `SSLConfigTool` in the same Oracle home earlier.
- `opwd` is the Oracle administrator password. If no password is specified, you will be prompted to enter the password.

See Also: *Oracle Application Server Administrator's Guide*

The sections that follow describe the usage of the SSL Configuration Tool for all the SSL configurations. You can also perform manual steps to configure SSL, which are also described in detail.

Checks to Perform Before Configuring SSL

Before using the methods recommended to configure SSL, you must confirm that OracleAS Portal is working correctly in the default non-SSL configuration. To test this, you must ensure that the following portal tasks work without errors:

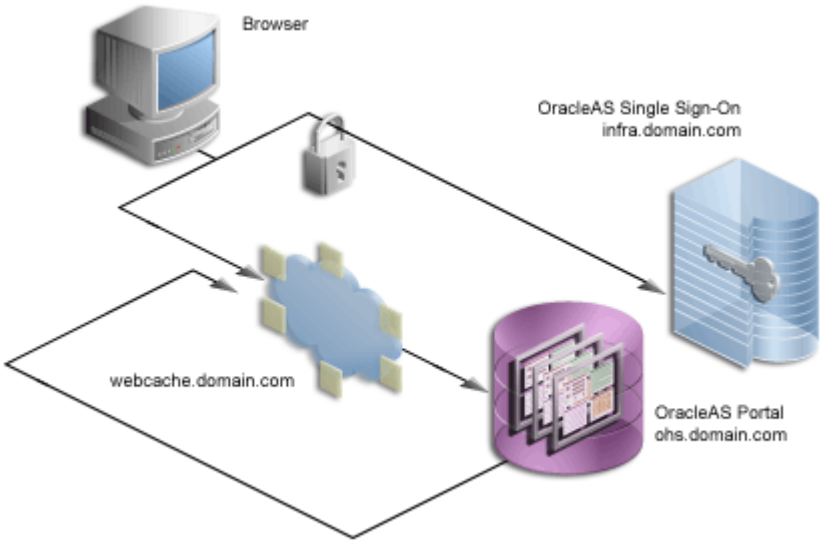
- The OracleAS Portal home page is accessible
- Users can log in to OracleAS Portal
- Edits to content are shown immediately

6.3.2.1.2 SSL to OracleAS Single Sign-On

If any connection should be secured with SSL, it is the connection between the browser and OracleAS Single Sign-On. The password should be protected by SSL in transit between the browser and OracleAS Single Sign-On. For at least a minimal level of security, you should configure your installation with this option. All of the subsequent SSL configurations assume that you have configured SSL for OracleAS Single Sign-On.

Figure 6–16 shows a configuration where OracleAS Single Sign-On is configured to use SSL.

Figure 6–16 Secured Connection to OracleAS Single Sign-On



After you have successfully performed the checks described in "Checks to Perform Before Configuring SSL", you can use either of the following two methods to configure SSL to OracleAS Single Sign-On:

- Configuring SSL to OracleAS Single Sign-On Using SSLConfigTool
- Configuring SSL to OracleAS Single Sign-On Manually

Configuring SSL to OracleAS Single Sign-On Using SSLConfigTool

Refer to the section "SSL Configuration Tool and Its Limitations" before using `SSLConfigTool`.

To configure SSL to OracleAS Single Sign-On using SSLConfigTool, perform the following steps:

1. Run the `SSLConfigTool` script on the Oracle Application Server Infrastructure.
   a. Enable SSL on the OracleAS Infrastructure that has Identity Management installed. Run `SSLConfigTool` in the infrastructure Oracle home, as shown in the following example for Windows:

      ```
      SSLConfigTool.bat -config_w_default -opwd <orcladmin_pwd>
      ```

      Where:

      - `config_w_default` is used to run the tool in silent mode using the values specified in the `portlist.ini` and `ias.properties` files.
      - `opwd` is the Oracle administrator password. If no password is specified, you will be prompted to enter the password.

See Also: *Oracle Application Server Administrator's Guide*

The information on enabling SSL in the *Oracle Application Server Single Sign-On Administrator's Guide*. If you are going to configure OracleAS Single Sign-On behind a reverse proxy server, you should refer to the information on deploying OracleAS Single Sign-On with a proxy server, in the *Oracle Application Server Enterprise Deployment Guide*.

Note: In the previously described configuration of SSL, you must re-register the OracleAS Single Sign-On middle-tier partner application. Because the OracleAS Single Sign-On middle-tier partner application is still non-SSL, you must re-register it as non-SSL. Therefore, the re-registration of mod_osso needs to specify the non-SSL URL of the OracleAS Single Sign-On middle tier for the `mod_osso_url` parameter to `ssoreg`.

Refer to the information on registering mod_osso in the *Oracle Application Server Single Sign-On Administrator's Guide*.

---

b. After enabling SSL on the OracleAS Infrastructure that has Identity Management installed, you must protect OracleAS Single Sign-On URLs. To do this, refer to the section "Protect Single Sign-On URLs" in the *Oracle Application Server Single Sign-On Administrator's Guide*.

2. Create a wallet. See "Creating an Empty Wallet (HTTPS)" for more information.
3. Check if the issuer of the OracleAS Single Sign-On certificate is listed in the **Trusted Certificates** list. If not, you must add the Trusted Root Certificate to the Wallet, as shown in "Adding the Trusted Root Certificate to the Wallet (HTTPS)".
4. Update wallet path and password in `iasconfig.xml`. See "Updating Wallet Path and Password in iasconfig.xml (HTTPS)" for more information.
5. Run the `SSLConfigTool` script on the Oracle Application Server middle tier, or multiple middle-tier Oracle homes, as shown in the following example, for Windows:

   ```
   SSLConfigTool.bat -config_w_prompt -ptl_inv_pwd <ptl_inv_pwd> -opwd <orcladmin_pwd>
   ```

   Where:

   - `config_w_prompt` is passed to run `SSLConfigTool` in interactive mode.
   - `ptl_inv_pwd` is the portal invalidation password.
   - `opwd` is the Oracle administrator password. If no password is specified, you will be prompted to enter the password.

   Enter `n` when prompted by the script to configure your site to accept browser requests using the SSL protocol.

   Choose **No** when prompted to configure HTTPS.
6. Verify the wallet path and the OracleAS Single Sign-On Query Path URL. See "Verifying the Wallet Path and the OracleAS Single Sign-On Query Path URL (HTTP and HTTPS)" for more information.
7. The `orcldasurlbase` attribute in the cn=OperationURLs, cn=DAS, cn=Products, cn=OracleContext entry may need to be updated in Oracle Internet Directory. If it

is not set to use the HTTPS port, you must refresh the cache for the Oracle Internet Directory parameters: To do this, perform the following steps:

a. Using Oracle Directory Manager (Integrated Management Tools : Oracle Directory Manager on Windows, or *INFRA_ORACLE_HOME*/bin/oidadmin on UNIX), run the Oracle Directory Manager and log in as cn=orcladmin.

b. Navigate to Entry Management, **cn=OracleContext > cn=Products > cn=DAS > cn=OperationURLs**.

c. Check if the orcldasurlbase entry reflects the HTTPS port being used on the infrastructure tier, that is, https://*<infrahost>*:*<port>*/.

If the orcldasurlbase entry does not reflect the HTTPS port, change it in Oracle Internet Directory and force a refresh of the portal cache, which holds the relevant Oracle Internet Directory information. To refresh the portal cache, perform the following steps:

a. Logon to OracleAS Portal as a user with administrator privileges.

b. Go to the **Builder**.

c. Click the **Administration** tab.

d. From the **Portal** tab, open **Global Settings** and navigate to the **SSO/OID** tab.

e. Scroll to the bottom of the page.

f. Select **Refresh Cache for OID Parameters**.

g. Click **Apply**.

h. The page should refresh with the appropriate value in the **DAS Host Name** field.

At this point, configuration is complete for SSL communication to OracleAS Single Sign-On.

Configuring SSL to OracleAS Single Sign-On Manually

To manually configure this option, refer to the information on enabling SSL in the *Oracle Application Server Single Sign-On Administrator's Guide*. If you are going to configure OracleAS Single Sign-On behind a reverse proxy server, you should refer to the information on deploying OracleAS Single Sign-On with a proxy server, in the *Oracle Application Server Single Sign-On Administrator's Guide*.

Note: In the previously described configuration of SSL, you must re-register the OracleAS Single Sign-On middle-tier partner application. As the OracleAS Single Sign-On middle-tier partner application is still non-SSL, you must re-register it as non-SSL. Therefore, the re-registration of mod_osso needs to specify the non-SSL URL of the OracleAS Single Sign-On middle tier for the mod_osso_url parameter to ssoreg.

Refer to the information on registering mod_osso in the *Oracle Application Server Single Sign-On Administrator's Guide*.

After enabling SSL on OracleAS Single Sign-On following the steps listed in the *Oracle Application Server Single Sign-On Administrator's Guide*. In this release of OracleAS Portal, you have the option of configuring portal to access the OracleAS Single Sign-On URLs over HTTP or HTTPS. The steps that follow indicate which steps are needed for each configuration:

Creating an Empty Wallet (HTTPS)

Perform the steps in this section *only* if you plan to use an HTTPS port on OracleAS Single Sign-On, after configuring OracleAS Single Sign-On to use SSL.

Create an empty wallet to establish the trust points for SSL access to the OracleAS Single Sign-On. To do this, perform the following steps:

1. Open the Oracle Wallet Manager on the *INFRA_ORACLE_HOME*. Note that you can run Oracle Wallet Manager from any location, as long as the generated wallet is accessible from the portal schema in the OracleAS Metadata Repository. You can also save the wallet (the directory containing the wallet files) anywhere and move it to another location that is accessible from the portal schema in the OracleAS Metadata Repository.
2. Choose **Wallet > New**.

   On UNIX, the wallet is stored in the following location by default:

   ```
   /etc/ORACLE/WALLETS/<Account Name creating the Wallet>
   ```

   On Windows, the wallet is stored in the following location by default:

   ```
   \Documents And Settings\<Account Name creating the Wallet>\ORACLE\WALLETS
   ```

3. Create a password for the wallet.
4. Enter the same password in the **Confirm Password** field.
5. Select **Standard** for **Wallet Type**.
6. Click **OK**.
7. Click **No** when asked to create a certificate request.
8. Check if the issuer of the OracleAS Single Sign-On certificate is listed in the **Trusted Certificates** list. If not, you must add the Trusted Root Certificate to the Wallet, as shown in "Adding the Trusted Root Certificate to the Wallet (HTTPS)".
9. Save the wallet in a convenient directory, for example:

   ```
   INFRA_ORACLE_HOME\wallets
   ```

10. Choose **Wallet > Save**.
11. Check **Wallet > AutoLogin**, if it is not already checked.

Adding the Trusted Root Certificate to the Wallet (HTTPS)

Perform the steps in this section only if you do not have the trusted root certificate of the OracleAS Single Sign-On server's issuing certificate authority listed in the **Trusted Certificates** list. In this case, you must add the Trusted Root Certificate to the Wallet as shown in the following steps, which are based on the Internet Explorer browser:

1. Using the browser, go to the OracleAS Single Sign-On home page, `https://infra.domain.com/pls/orasso`. Ensure that this is on an HTTPS URL.
   a. If the certificate on the server is not automatically trusted by your browser, the **Security Alert** dialog box is shown.
   b. Click **View Certificate**.
   c. Click the **Certification Path** tab.
   d. Select the **Certificate Authority Root**, which is the first certificate in the list.
   e. Click **View Certificate**.
   f. Click **Install Certificate**.

      This brings up the **Certificate Import Wizard**. This will import the certificate into the browser's list of trusted certificate authorities.
   g. Click **Next**.
   h. Select **Automatically select a certificate store based on the type of certificate**.
   i. Click **Next**.
   j. Click **Finish**.

      The trusted root certificate is installed in your browser.
2. Click the lock icon in the status bar to bring up the **Certificate** dialog box.
3. Click the **Certification Path** tab.
4. Select the **Certificate Authority Root**, which is the first certificate in the list.
5. Click **View Certificate**.
6. Click the **Details** tab.
7. Click **Copy to File...**.

   This brings up the **Certificate Export Wizard**.
8. Click **Next**.
9. Under **Select the format you want to use,** select **Base-64 encoded X.509 (.CER)**.
10. Click **Next** and specify a file name for the certificate. You can provide any filename for the certificate with a `.cer` extension.
11. Click **Next** and then **Finish**.

At this point, a `.cer` certificate file is created, which contains the trusted certificate authority's root certificate. This certificate must be added to the Wallet's list of Trusted Certificates. To accomplish this, assuming that the wallet manager is already running and the empty wallet has been opened, perform the following steps:

1. Right-click the **Trusted Certificates** node.
2. Select **Import Trusted Certificate...**.
3. Select **Paste the certificate**, and click **OK**.

4. Copy the contents of the certificate file you created earlier into the text area for the BASE64 format certificate, and then click **OK**.
5. Verify that the Certificate Authority Root has been added to the list of trusted certificates.
6. Save the wallet.

Updating Wallet Path and Password in iasconfig.xml (HTTPS)

Perform the steps in this section *only* if you plan to use an HTTPS port on OracleAS Single Sign-On, after configuring OracleAS Single Sign-On to use SSL.

To update the wallet path and password in the `iasconfig.xml` file, perform the following tasks:

1. Open the `iasconfig.xml` file, which is available in the following directory:

   MID_TIER_ORACLE_HOME/portal/conf

2. Update the wallet path and password as shown in the following example:

```
<IASConfig XSDVersion="1.0">
    <IASInstance Name="iAS.infra.abc.com" Host="infra.abc.com">
        <OIDComponent AdminPassword="@BVs2KPJEWC5a0l4n8lbTxUY="
        PortSSLEnabled="true" LDAPPort="3060" AdminDN="cn=orcladmin"/>
    </IASInstance>
    <IASInstance Name="iAS.w1.abc.com" Host="infra.abc.com">
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
    </IASInstance>
    <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
    SchemaPassword="@Beyh8p2bOWELQCsA5zRtuYc="
    ConnectString="cn=iasdb,cn=oraclecontext"
    WalletPath="file:/export/home/oracle/wallets"
    WalletPassword="welcome">
       <WebCacheDependency ContainerType="IASFarm" Name="Farm-1.lbr.abc.com"/>
       <OIDDependency ContainerType="IASInstance" Name="iAS.infra.abc.com"/>
       <EMDependency ContainerType="IASInstance" Name="iAS.w1.abc.com"/>
    </PortalInstance>
</IASConfig>
```

> **Note:** While performing the next task, which is, reregistering OracleAS Portal partner applications, the wallet password in the `iasconfig.xml` gets encrypted, and the portal schema in the OracleAS Metadata Repository gets updated with the changes made in the `iasconfig.xml` file.

Setting the OracleAS Single Sign-On Query Path URL (HTTP)

Perform the steps in this section *only* if you plan to use an HTTP port on OracleAS Single Sign-On to support these interfaces, after configuring OracleAS Single Sign-On to use SSL.

OracleAS Portal maintains the URL prefix of OracleAS Single Sign-On, which accesses certain information through calls from the database using the UTL_HTTP package. These calls can be made using HTTP or HTTPS. As a result, even if OracleAS Portal and OracleAS Single Sign-On are configured to use HTTPS, you can still use an HTTP port on OracleAS Single Sign-On to support these interfaces.

The calls made across this interface are required for the following reasons:

- Obtain the list of external applications to allow personalization of the **External Applications** portlet.
- Perform the mapping of OracleAS Single Sign-On user names to external application user names.

To set this URL prefix, and the OracleAS Single Sign-On Query Path URL, perform the following steps:

1. Open the `iasconfig.xml` file, which is available in the following directory:

   MID_TIER_ORACLE_HOME/portal/conf

2. Add or update the `SSOQueryPath` value as shown in the following example:

```
<IASConfig XSDVersion="1.0">
    <IASInstance Name="iAS.infra.abc.com" Host="infra.abc.com">
        <OIDComponent AdminPassword="@BVs2KPJEWC5a0l4n8lbTxUY="
        PortSSLEnabled="true" LDAPPort="3060" AdminDN="cn=orcladmin"/>
    </IASInstance>
    <IASInstance Name="iAS.w1.abc.com" Host="infra.abc.com">
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
    </IASInstance>
    <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
    SchemaPassword="@Beyh8p2bOWELQCsA5zRtuYc="
    ConnectString="cn=iasdb,cn=oraclecontext"
    SSOQueryPath="http://infra.abc.com:7777/pls/orasso/">
       <WebCacheDependency ContainerType="IASFarm" Name="Farm-1.lbr.abc.com"/>
       <OIDDependency ContainerType="IASInstance" Name="iAS.infra.abc.com"/>
       <EMDependency ContainerType="IASInstance" Name="iAS.w1.abc.com"/>
    </PortalInstance>
</IASConfig>
```

Re-Registering the OracleAS Portal Partner Application (HTTP and HTTPS)

After OracleAS Single Sign-On is SSL-enabled, all OracleAS Single Sign-On partner applications need to be re-registered so that the updated SSL login URL is obtained by each partner application for subsequent authentication requests.

To re-register the OracleAS Portal partner applications, invoke `ptlconfig` (on the OracleAS Portal middle tier) in the following modes:

1. Encrypt any plain text passwords in the `iasconfig.xml` configuration file. To do this, navigate to *MID_TIER_ORACLE_HOME*/portal/conf, and run the following command:

   ```
   ptlconfig -encrypt
   ```

   Note: To use `ptlconfig`, the `ORACLE_HOME` environment variable must be set.

2. Register the URL changes with OracleAS Portal. To do this, navigate to *MID_TIER_ORACLE_HOME*/portal/conf, and run the following command:

   ```
   ptlconfig -dad <portal_dadname> -site
   ```

   For example,

   ```
   ptlconfig -dad portal -site
   ```

Note: If you have multiple virtual hosts configured in OracleAS Portal, you will need to re-register each of the virtual hostnames individually using the command `ptlconfig -dad portal -sso -host <portal_host> -port <port> [-ssl]`, specifying the host and port, for each virtual hostname. Refer to Section A.1, "Portal Dependency Settings Tool" for more information on using `ptlconfig`.

Verifying the Wallet Path and the OracleAS Single Sign-On Query Path URL (HTTP and HTTPS)

OracleAS Portal maintains the URL prefix of OracleAS Single Sign-On, which accesses certain information through calls from the database using the UTL_HTTP package. The calls made across this interface are required for the following reasons:

- Obtain the list of external applications to allow customization of the External Applications portlet.
- Perform the mapping of OracleAS Single Sign-On user names to external application user names.

To verify the wallet path and OracleAS Single Sign-On Query Path URL, perform the following steps:

1. Log in to OracleAS Portal as the portal administrator.
2. Click the **Administer** tab.
3. Click the **Portal** tab.
4. Click **Global Settings** in the Services portlet.
5. Click the **SSO/OID** tab.
6. The SSO Server Settings section should have the appropriate values.

Conditionally Updating the Oracle Delegated Administration Services URL Base Entry in Oracle Internet Directory (HTTP and HTTPS)

After enabling the infrastructure tier's Oracle HTTP Server for SSL, you were asked to re-register all partner applications, which includes mod_osso on the infrastructure tier. You have the option of accessing Oracle Delegated Administration Services over non-SSL or SSL. The base URL for Oracle Delegated Administration Services is stored in Oracle Internet Directory, and this determines the URL that other applications render when providing links to Oracle Delegated Administration Services functionality.

If you want Oracle Delegated Administration Services accessed over SSL, then the re-registration of mod_osso needs to specify an SSL URL for the `mod_osso_url` parameter to `ssoreg`. Refer to the information on registering mod_osso in the *Oracle Application Server Single Sign-On Administrator's Guide*.

If you decide that you want to access Oracle Delegated Administration Services over SSL, then the `orcldasurlbase` attribute in the cn=OperationURLs, cn=DAS, cn=Products, cn=OracleContext entry needs to be updated in Oracle Internet Directory to reflect this fact. This attribute value is then used by OracleAS Portal for generating subsequent Oracle Delegated Administration Services URLs. This procedure assumes that the Oracle HTTP Server on the infrastructure tier is also listening on an HTTPS port.

1. For this step, you need Oracle Directory Manager (Integrated Management Tools : Oracle Directory Manager on Windows, or *INFRA_ORACLE_HOME*/bin/oidadmin on UNIX). Run the Oracle Directory Manager and log in as cn=orcladmin.
2. Navigate to Entry Management, **cn=OracleContext > cn=Products > cn=DAS > cn=OperationURLs**.
3. Update the orcldasurlbase entry to reflect the HTTPS port being used on the infrastructure tier, that is, https://*<infrahost>*:*<port>*/.

Once the entry is updated in Oracle Internet Directory, you must force a refresh of the portal cache, which holds the relevant Oracle Internet Directory information.

1. Logon to OracleAS Portal as a user with administrator privileges.
2. Go to the **Builder**.
3. Click the **Administration** tab.
4. From the **Portal** tab, open **Global Settings** and navigate to the **SSO/OID** tab.
5. Scroll to the bottom of the page.
6. Select **Refresh Cache for OID Parameters**.
7. Click **Apply**.
8. The page should refresh with the appropriate value in the **DAS Host Name** field.

Re-Registering the Oracle HTTP Server Partner Application (HTTP and HTTPS)

Run ssoreg to register the virtual host for which mod_osso facilitates single sign-on. The specific application URLs to be defined as partner applications within this site are defined in the file osso.conf. ssoreg is located in the directory *MID_TIER_ORACLE_HOME*/sso/bin.

The following example shows the usage of ssoreg on UNIX:

```
MID_TIER_ORACLE_HOME/sso/bin/ssoreg.sh
-oracle_home_path MID_TIER_ORACLE_HOME
-site_name www.abc.com
-config_mod_osso TRUE
-mod_osso_url http://www.abc.com:7777
-config_file MID_TIER_ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
-admin_info cn=orcladmin
```

On Windows, you must run the ssoreg.bat batch file instead. Refer to the information on creating partner applications in the *Oracle Application Server Single Sign-On Administrator's Guide*.

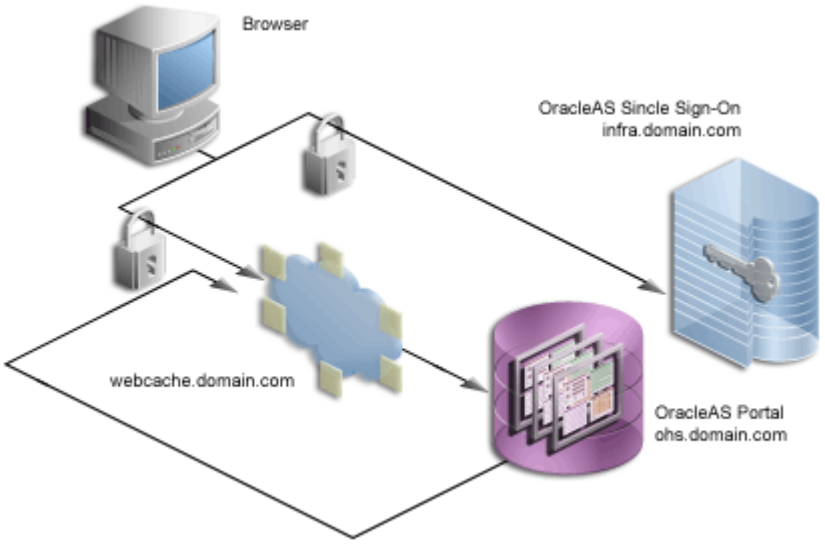
At this point, configuration is complete for SSL communication to OracleAS Single Sign-On.

6.3.2.1.3 SSL to OracleAS Web Cache

Once you secure the OracleAS Single Sign-On communication, the next option is to secure the communication to the front door of OracleAS Portal, which is OracleAS Web Cache. In this configuration, OracleAS Web Cache can forward the request to the Oracle HTTP Server, which is acting as the OracleAS Portal middle tier, using HTTP for better performance. Similarly, the Parallel Page Engine requests for portlet content that loopback to OracleAS Web Cache can request the content using HTTP.

Figure 6–17 shows a configuration where OracleAS Web Cache is configured to use SSL.

Figure 6–17 Secured Connection to OracleAS Web Cache



After you have successfully performed the checks described in "Checks to Perform Before Configuring SSL", you can use either of the following two methods to configure SSL to OracleAS Single Sign-On:

- Configuring SSL to OracleAS Web Cache Using SSLConfigTool
- Configuring SSL to OracleAS Web Cache Manually

Configuring SSL to OracleAS Web Cache Using SSLConfigTool

Before using `SSLConfigTool`, refer to the section "SSL Configuration Tool and Its Limitations".

To configure SSL for OracleAS Web Cache, perform the following tasks:

1. Perform the steps described in "Configuring SSL to OracleAS Single Sign-On Using SSLConfigTool" to enable OracleAS Single Sign-On for SSL.
2. Create a wallet. Refer to "Creating a Wallet" for details.
3. Run `SSLConfigTool` in the middle-tier Oracle home to setup SSL for OracleAS Web Cache, as shown in the following example for Windows:

   ```
   SSLConfigTool.bat -config_w_prompt -ptl_inv_pwd <ptl_inv_pwd> -opwd <orcladmin_pwd>
   ```

   Where:

   - `config_w_prompt` is passed to run `SSLConfigTool` in interactive mode.
   - `ptl_inv_pwd` is the portal invalidation password.
   - `opwd` is the Oracle administrator password. If no password is specified, you will be prompted to enter the password.

   Enter the following values when prompted by the script:

   - `y`, when prompted to configure your site to accept browser requests using the SSL protocol.
   - `n`, when asked if your Oracle HTTP Server accepts requests in SSL protocol.

4. Configure and register Web providers or Provider groups. See "Configuring and Registering Web Providers or Provider Groups Exposed over SSL" for details.
5. Configure and register WSRP producers exposed over SSL. See "Configuring and Registering WSRP Producers Exposed Over SSL" for details.
6. Add the Web provider server certificate to the trusted certificate file. See "Augmenting the Portal Tools Trusted Certificate File" for details.
7. Enable access for Oracle Ultra Search. See "Enabling Access for Oracle Ultra Search" for details.
8. Re-register OracleAS Portal as an Oracle Ultra Search content source. See "Re-Registering OracleAS Portal as an Oracle Ultra Search Content Source" for details.

See the *Oracle Application Server Administrator's Guide* for details.

Configuring SSL to OracleAS Web Cache Manually

To configure SSL to OracleAS Web Cache manually, perform the following steps:

- Enabling OracleAS Single Sign-On for SSL
- Creating a Wallet
- Securing OracleAS Web Cache
- Securing the Parallel Page Engine
- Re-Registering the Oracle HTTP Server Partner Application
- Specifying the OracleAS Portal Published Address and Protocol
- Configuring and Registering Web Providers or Provider Groups Exposed over SSL
- Configuring and Registering WSRP Producers Exposed Over SSL
- Augmenting the Portal Tools Trusted Certificate File
- Enabling Access for Oracle Ultra Search
- Re-Registering OracleAS Portal as an Oracle Ultra Search Content Source

Enabling OracleAS Single Sign-On for SSL

Perform the steps described in "Configuring SSL to OracleAS Single Sign-On Manually" to enable OracleAS Single Sign-On for SSL.

Creating a Wallet

The various components of OracleAS Portal use the Oracle Wallet Manager to store the certificates for the secure communication. The first step in this process is to obtain a certificate from a Certificate Authority (for example, OracleAS Certificate Authority, Verisign, GTE CyberTrust, and so on).

> **Note:** When you enabled OracleAS Single Sign-On for SSL, you had to create an empty wallet. For this procedure, Oracle recommends that you create a new wallet.

Obtaining a Certificate To obtain a digital certificate from the relevant signing authority, you submit a Certificate Request (CR) uniquely identifying your server to the signing authority.

1. Open the Oracle Wallet Manager in the middle-tier *MID_TIER_ORACLE_HOME*. On UNIX, enter `owm` from the command prompt. On Windows, invoke Oracle Wallet Manager from the **Start** menu.
2. Choose **Wallet > New**.

   On UNIX, the wallet is stored in the following location by default:

   ```
   /etc/ORACLE/WALLETS/<Account Name creating the Wallet>
   ```

   On Windows, the wallet is stored in the following location by default:

   ```
   \Documents And Settings\<Account Name creating the Wallet>\ORACLE\WALLETS
   ```

3. Create a password for the wallet.
4. Click **Yes** to accept the option to create a CR.
5. Fill out the **Certificate Request** dialog with details that uniquely identify your server. Table 6–19 shows some sample values for the various fields in the **Certificate Request** dialog.

Table 6–19 Sample Values for Fields in the Certificate Request Dialog

Field Name	Sample Values
Common Name	www.abc.com
Organizational Unit	DOCUMENTATION
Organization	Oracle Corporation
Locality/City	Redwood Shores
State/Province (**Note:** Do not use abbreviations; the value specified for state or province must be completely spelled out)	California

6. Click **OK**. A dialog will inform you that the certificate request was created successfully. The Certificate node in the Wallet Navigator will change to `Requested`.
7. Save the wallet in a convenient directory, for example:

   MID_TIER_ORACLE_HOME`\webcache\wallets\portalssl`

8. Send the CR to the chosen Certificate Authority (CA).

> **Note:** Certificates are issued by trusted third parties known as Certification Authorities (CA), for example, OracleAS Certificate Authority or Verisign. For details on how to obtain a certificate, check the appropriate vendor's Web site.

Cutting and Pasting

Depending on the CA, you may need to cut and paste the certificate request in their Web page or export the CR to a file for subsequent uploading to the site.

1. Select the **Certificate** node in the Wallet Navigator.
2. Highlight the Certificate text in the **Certificate Request** field. Make sure to include the `BEGIN/END NEW CERTIFICATE REQUEST` lines.

3. Copy and paste into the **Certificate Request** field on the CA's Web site.

Exporting Certificate Request

To export the certificate request:

1. Choose **Operations > Export Certificate Request**.
2. Choose the Name and Location of the CR file. A Status Line Message will confirm the successful export of the CR.
3. Once exported, the CR can be uploaded to the CA's Web site.

Managing Trusted Certificates Once the CA has processed the request, the User Certificate is forwarded to you either as text within an e-mail or as a simple file that is downloaded from a given Web page.

Note: If you are using a trial Root Certificate or have chosen a CA which is not currently installed in the Oracle Wallet Manager, you must first import the CA's Trusted Certificate before importing your server-specific User Certificate.

Importing Trusted Certificate (if necessary)

To import the trusted certificate:

1. Choose **Operations > Import Trusted Certificate**.
2. Based on the CA, choose **Paste the Certificate** or **Select a file that contains the certificate**.
3. Select the appropriate certificate file or paste in the text from the e-mail. Oracle Wallet Manager expects base-64 encoded root certificates. If you do not have a base-64 encoded root certificate, then you must perform the steps described in the "Changing Trusted Certificate Format (if necessary)" section.
4. Click **OK**.

A status line message should appear indicating that the certificate was successfully imported. When you import the server specific User Certificate, the certificate node in the tree structure should also display as **Ready**.

Changing Trusted Certificate Format (if necessary)

If the certificate import fails, then it is possible that the Certificate is in a format that the Oracle Wallet Manager does not support. In this case, you need to convert it to a supported format before importing. The easiest way to do this is through the certificate Import and Export Wizards within a browser. The following steps are for the Internet Explorer browser:

1. In Internet Explorer, select **Tools > Internet Options...**.
2. Click the **Content** tab.
3. Click **Certificates...**.
4. Click the **Trusted Root Certification Authorities** tab.
5. Select **Import...** and follow the wizard to import the certificate.
6. Highlight the newly imported certificate from the list.
7. Click **Export...** and follow the wizard to the Export File Format page.

8. Choose **Base-64 encoded X.509**.
9. Click **Next** and give the certificate a file name.
10. Click **Next**.
11. Click **Finish**.
12. In Oracle Wallet Manager, choose **Operations > Import Trusted Certificate**.

Once the Trusted Root Certificate has been successfully imported into the Oracle Wallet Manager you may then import the server specific User Certificate.

Importing Server's User Certificate

To import the server's user certificate:

1. Choose **Operations > Import User Certificate**.
2. Based on the CA, choose **Paste the Certificate** or **Select a file that contains the certificate**.
3. Select the appropriate certificate file or paste in the text from the e-mail.
4. Click **OK**.

A status line message should appear indicating that the User Certificate has been successfully imported.

Having imported the certificate, it is important to save the wallet with the Autologin functionality enabled. This step is required because OracleAS Web Cache accesses the wallet as the process starts and the wallet password is not held by OracleAS Web Cache. If this property is not set, OracleAS Web Cache immediately shuts down if running in SSL mode. To set this property, perform the following steps:

1. Choose the Trusted Certificate you just imported from the list in the Oracle Wallet Manager.
2. Choose **Wallet > Save**.
3. Check **Wallet > AutoLogin**, if it is not already checked.

Securing OracleAS Web Cache

The sections that follow describe how to configure OracleAS Web Cache to accept SSL connections.

> **Note:** Changing OracleAS Web Cache settings (for example, Listening Port) can change the OracleAS Portal URL. If you do this, mobile settings need to be updated manually. See Section C.8, "Using the cfgiasw Script to Configure Mobile Settings" for more information.

Configuring OracleAS Web Cache SSL Port To Configure the OracleAS Web Cache SSL port, perform the following steps:

1. From the OracleAS Web Cache administration page, click the **Listen Ports** link in the **Ports** section.
2. To add the SSL port, click **Add...** and enter the following information:
   - **IP Address:** `ANY`
   - **Port Number:** SSL port that Web Cache will listen on

- **Protocol:** `HTTPS`
- **Require Client-Side Certificate:** No (unchecked)
- **Wallet:** Path to the Oracle Wallet directory containing the SSL server certificate

3. Click **Submit**.

For more information on the procedure in the preceding text, refer to "Task 3: Configure HTTPS Operations Ports for the Cache" in Chapter 8, "Specialized Configurations" of the *Oracle Application Server Web Cache Administrator's Guide*.

Defining a Site for the Published SSL Hostname and Port As there is no out-of-box default SSL configuration, you need to add a Site definition for the SSL-based Site that OracleAS Web Cache will be caching. To add a site definition, perform the following steps:

1. From the OracleAS Web Cache administration page, click **Site Definitions** under **Origin Servers, Sites, and Load Balancing**.
2. Define a Site where **Host Name** is the hostname seen by the browser.

   This is the load balancing router or reverse proxy server name for configurations that use a load balancing router or other reverse proxy, or it is the OracleAS Web Cache hostname in a configuration where OracleAS Web Cache receives browser requests.
3. Set **Port** to the HTTPS port addressed by the browser requests.
4. Enter Site information as follows:

   URL Path Prefix: Leave blank

   HTTPS Only Prefix: Leave blank

   Client-Side Certificate: Not required

   URL Parameters to Ignore: Leave blank

   Default Site: Yes

   Create Alias from Site Name with/without www: No
5. Click **Submit.**
6. Remove any sites that are no longer applicable to your modified configuration, including the default, out-of-the-box non-SSL site.

For more information on the procedure in the preceding text, refer to:

- The section on creating a site for HTTPS requests for the cache, in the *Oracle Application Server Web Cache Administrator's Guide*.
- The section on creating site definitions in the *Oracle Application Server Web Cache Administrator's Guide*.

Adding Site Aliases Site aliases are necessary if content is cached across a number of different hostnames, or ports, or both, but which actually refer to the same logical content. For example, when the PPE makes a request for a portlet, and this portlet is requested on a non-SSL port, but the main Site is accessed over SSL, then an alias entry is needed. This equates the content accessed through the Site with SSL, to the content accessed over non-SSL. This way, invalidation requests that are sent to invalidate the content, will invalidate the content that is cached over either form of URL.

You need to create an alias for the non-SSL OracleAS Web Cache listening port for the OracleAS Web Cache SSL site. To create a site alias:

1. From the OracleAS Web Cache administration page, return to the **Site Definitions** page, select the newly added site, and click **Add Alias**.
2. Enter the same hostname as used by the site entry, and provide the non-SSL port that the PPE will use to request portlets from OracleAS Web Cache. Click **Apply Changes**.
3. Restart the server after making the necessary additions.

For example, if the logical site name is accessed using the URL `https://portal.mycompany.com:4443`, and the non-SSL Listen Port for OracleAS Web Cache is `7777`, then you should select the Site with SSL Port `4443`, and add an alias for it using the non-SSL port of OracleAS Web Cache (`7777`).

For more information on the procedure in the preceding text, refer to the section on creating site definitions in the *Oracle Application Server Web Cache Administrator's Guide*.

Adding Site to Server Mappings of the New Site to the Origin Server After adding a new site definition, you must add the site to server mapping for the newly defined site to the origin server. To do this:

1. From the OracleAS Web Cache Administration page, click **Site-to-Server Mapping** under **Origin Servers**, **Sites**, and **Load Balancing**.
2. Select the first option of the Site-to-Server Mapping table.
3. Click **Insert Above..** to bring up the **Edit/Add Site-to-Server Mapping** dialog window.
4. Select the **Site** you are mapping from the drop-down list, which should be the OracleAS Web Cache site with the SSL port. For example, if your logical site name is `https://portal.mycompany.com:4443`, then select the site with **Host Name** `portal.mycompany.com` and **Port** `4443`.
5. In the **Select Application Web Servers** field, select the non-SSL Origin Server to which requests should be routed for content. For example, if the origin server is running on port `7778` on host `portal.mycompany.com`, then select **portal.mycompany.com:7778 HTTP**.
6. Click **Submit** to close the dialog box and see the new mapping appear in Table 6–21, " Site-to-Server Mapping" of the original page.
7. Click **Apply Changes**.
8. Restart the server.

For more information on the procedure in the preceding text, refer to the section on mapping sites to origin servers in the *Oracle Application Server Web Cache Administrator's Guide*.

Securing the Parallel Page Engine

In this configuration, you need to configure the PPE to make portlet requests using HTTP requests. The sections that follow describe how to implement a partial SSL PPE configuration for this purpose. To configure the PPE, perform the following steps:

1. Open the `web.xml` file associated with the OC4J_PORTAL instance on the middle tier. The file is located in the following directory:

   MID_TIER_ORACLE_HOME`\j2ee\OC4J_Portal\applications\portal\portal\WEB-INF\`

2. Add `useScheme`, `usePort`, and `httpsports` in additional `<init-param>` blocks in `web.xml`. The `useScheme http` indicates that the HTTP protocol should be used for PPE loopbacks and `usePort` indicates which port these

non-SSL loopbacks should use. The HTTP port you specify for `usePort` should be the OracleAS Web Cache non-SSL HTTP port. The `httpsports` parameter must point to the OracleAS Web Cache HTTPS listening port. For example:

```
<servlet>
<servlet-name>page</servlet-name>
   <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
   <init-param>
     <param-name>useScheme</param-name>
     <param-value>http</param-value>
   </init-param>
   <init-param>
     <param-name>usePort</param-name>
     <param-value>7777</param-value>
   </init-param>
   <init-param>
     <param-name>httpsports</param-name>
     <param-value>4443</param-value>
   </init-param>
 </servlet>
```

(Optional) If you want the PPE to trust only specific certificates, add `x509certfile` in an additional `<init-param>` block in `web.xml`. The value of `x509certfile` is the absolute path to the text file containing the list of certificates which defines the group of "trusted" servers. For example:

```
<servlet>
<servlet-name>page</servlet-name>
   <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
     <init-param>
        <param-name>x509certfile</param-name>
        <param-value>C:\mySSLconfig\trustedCerts.txt</param-value>
     </init-param>
 </servlet>
```

Note: If you choose not to implement x509certfile, the PPE trusts any SSL certificate.

---

3. Save the `web.xml` file.

Re-Registering the Oracle HTTP Server Partner Application

Run `ssoreg` to register the virtual host for which mod_osso facilitates single sign-on. The specific application URLs to be defined as partner applications within this site are defined in the file `osso.conf`. `ssoreg` is located on the middle tier in *`MID_TIER_ORACLE_HOME`*`/sso/bin`.

The following example shows the usage of `ssoreg` on UNIX:

```
MID_TIER_ORACLE_HOME/sso/bin/ssoreg.sh
  -oracle_home_path MID_TIER_ORACLE_HOME
  -site_name www.abc.com
  -config_mod_osso TRUE
  -mod_osso_url https://www.abc.com:4443
  -config_file MID_TIER_ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
  -admin_info cn=orcladmin
```

On Windows you must run the `ssoreg.bat` batch file instead. Refer to the information on creating partner applications in the *Oracle Application Server Single Sign-On Administrator's Guide*.

At this point, configuration is complete for SSL communication to OracleAS Web Cache.

Specifying the OracleAS Portal Published Address and Protocol

To specify the published address for OracleAS Portal with the modified port for SSL, you need to use Oracle Enterprise Manager as follows:

1. Navigate to the Oracle Enterprise Manager 10*g* Application Server Control Console.
2. Click the Standalone Instance with the Oracle Application Server that is running the OracleAS Portal middle tier.
3. Click the OracleAS Portal system component.
4. Under **Administration**, click **Portal Web Cache Settings**.
5. For **Listen Port**, enter the OracleAS Web Cache SSL port number.
6. For **Listening Port SSL Enabled**, choose **Yes**.
7. Click **OK**. The OracleAS Portal schema is updated with the setting and the Oracle Enterprise Manager 10*g* target instance is updated to use HTTPS for its URL tests.

   If at a later time you choose to switch to HTTP, you would perform this same procedure and return **Listening Port SSL Enabled** to **No**.

   Notes:

   - This procedure updates the settings in the `iasconfig.xml` file.
   - See Appendix A, "Using the Portal Dependency Settings Tool and File" for more information about `iasconfig.xml`.

8. Edit `httpd.conf` as follows:
   a. Access the Application Server Control Console.
   b. Click the link for the application server where OracleAS Portal is installed.
   c. Click the **HTTP Server** link.
   d. Click the **Administration** link.
   e. Click **Advanced Server Properties**.
   f. Select **httpd.conf**.
   g. Scroll to the bottom of the file and enter the following `LoadModule` directive:

      On UNIX:

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

      On Windows:

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

For more information on mod_certheaders, refer to the *Oracle HTTP Server Administrator's Guide*.

h. Add a Virtual Host definition. This must be inserted after the `LoadModule` directive, as follows:

```
NameVirtualHost <OHS_computer_IP_address>:<OHS_listening_port>

<VirtualHost <OHS_computer_IP_address>:<OHS_listening_port>>
  ServerName <portal_site_server_name>
  Port <ssl_listening_port_for_portal_site>
  SimulateHttps On
  RewriteEngine On
  RewriteOptions inherit
</VirtualHost>
```

For example:

```
NameVirtualHost 123.45.67.8:7778

<VirtualHost 123.45.67.8:7778>
  ServerName w1.abc.com
  Port 4443
  SimulateHttps On
  RewriteEngine On
  RewriteOptions inherit
</VirtualHost>
```

i. Click **Apply**.

j. When asked to restart Oracle HTTP Server, click **Yes**.

9. Run the following command to synchronize the manual configuration changes:

```
MID_TIER_ORACLE_HOME/dcm/bin/dcmctl updateconfig
```

10. Restart your Oracle Application Server instance:

```
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl stopall
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl startall
```

Configuring and Registering Web Providers or Provider Groups Exposed over SSL

See "Configuring and Registering Web Providers or Provider Groups Exposed Over SSL" for details.

Configuring and Registering WSRP Producers Exposed Over SSL

See "Configuring and Registering WSRP Producers Exposed Over SSL" for details.

Augmenting the Portal Tools Trusted Certificate File

If you use the Web Page data source of OmniPortlet provider, you are doing a loopback to the Web Clipping provider, and so need to add the web provider server certificate to the trusted certificate file pointed to by the `<trustedCertificateLocation>` tag in OmniPortlet `provider.xml` file. The default certificate file is the `ca-bundle.crt` file, located in the *`MID_TIER_ORACLE_HOME`*`/portal/conf` directory.

To do this, perform the steps shown subsequently, which are based on the Internet Explorer browser. The steps may differ slightly if you are using another browser for capturing and exporting the necessary certificate.

1. Capture the necessary certificate: Point your browser to the Web Clipping provider test page:
   `http://<host>:<port>/portalTools/webClipping/providers/webClipping`.

   You should see a **Security Alert** dialog box that shows "*The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority*." or something similar. Click **View Certificate**, and then go through the following steps to export the certificate into a temporary file:

   a. Select the **Details** tab.

   b. Select **Show: <All>** in the drop-down list, and click **Copy to File...**.

   c. Run the Export wizard to export the certificate in Base-64 encoded X.509 format into a temporary file *`MID_TIER_ORACLE_HOME`*`/portal/conf/providertemp.cer`.

2. Append the certificate in the temporary file to the certificate file used by OmniPortlet provider (default is *`MID_TIER_ORACLE_HOME`*`/portal/conf/ca-bundle.crt`).

Enabling Access for Oracle Ultra Search

For Oracle Ultra Search to access secure Web sites, you need to import certificates into the crawler's trust store and the Oracle Containers for J2EE (OC4J) JVM's trust store on the middle-tier and infrastructure instances.

By default, the OC4J JVM recognizes certificates of well-known certificate authorities. However, if the secure portal instance uses a self-signed certificate or a certificate signed by an unknown certificate authority, then you must import the portal's certificate into the OC4J JVM's truststore. The OC4J JVM default truststore is located at *`ORACLE_HOME`*`/jdk/jre/lib/security/cacerts`.

To add the required certificate to the trust store, perform the following steps on the middle-tier and infrastructure instances:

1. Change directory to *`ORACLE_HOME`*`/jdk/jre/lib/security/` on the middle tier.

2. Create a backup of the truststore file `cacerts`, for example, `cacerts.bak`.

3. Execute the following command to add the required certificate to the trust store:

```
$ORACLE_HOME/jdk/bin/keytool -import -alias <aliasName> -file <root_
certificate_file_name> -trustcacerts -v -keystore $ORACLE_
HOME/jdk/jre/lib/security/cacerts
```

> **Note:** Use any arbitrary value for `-alias`, but ensure that it is unique for each certificate.

4. Provide the trust store password, and type "Yes", when prompted for confirmation.

Note: The preceding steps also need to be performed on the OracleAS Infrastructure containing the Oracle Ultra Search crawler.

Re-Registering OracleAS Portal as an Oracle Ultra Search Content Source

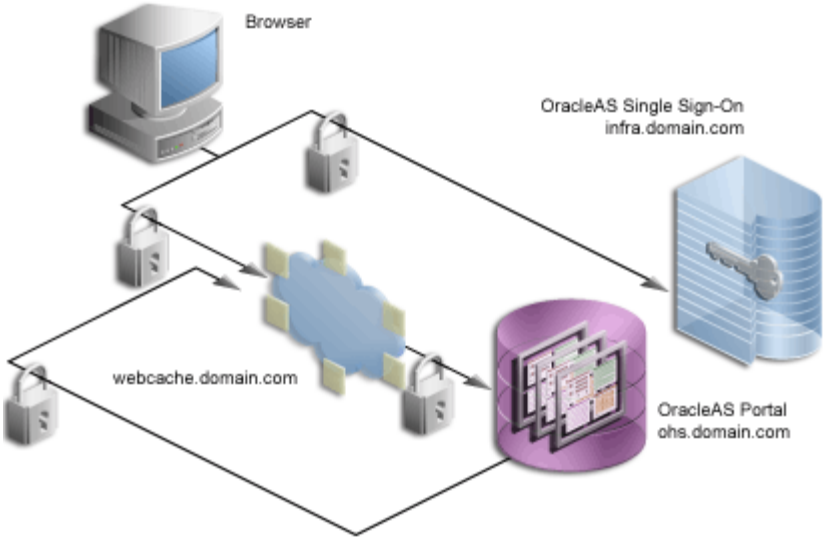
If you use Oracle Ultra Search to crawl your portal and you reconfigure OracleAS Web Cache to use SSL, you must re-register OracleAS Portal as a content source with Oracle Ultra Search. See Section 8.2.4.2, "Registering OracleAS Portal as a Content Source" for details.

6.3.2.1.4 SSL Throughout OracleAS Portal

For installations that require the utmost security, it is possible to configure SSL throughout the system. In this configuration, the loopback from the PPE to OracleAS Web Cache uses a wallet and the hop between the PPE and the Web providers uses a certificate directly rather than through a wallet.

Figure 6–18 shows a configuration where SSL is configured throughout OracleAS Portal.

Figure 6–18 Secured Connections Throughout the System



Note: If you have already followed the steps described in Section 6.3.2.1.3, "SSL to OracleAS Web Cache", you must revert all the changes you have made before configuring SSL throughout OracleAS Portal.

After you have successfully performed the checks described in "Checks to Perform Before Configuring SSL", you can use either of the following two methods to configure SSL to OracleAS Single Sign-On:

- Configuring SSL Throughout OracleAS Portal Using SSLConfigTool
- Configuring SSL Throughout OracleAS Portal Manually

Configuring SSL Throughout OracleAS Portal Using SSLConfigTool

Refer to the section "SSL Configuration Tool and Its Limitations" before using `SSLConfigTool`.

To configure SSL throughout OracleAS Portal using the SSL configuration tool, perform the following tasks:

1. Perform the steps described in "Configuring SSL to OracleAS Single Sign-On Using SSLConfigTool" to enable OracleAS Single Sign-On for SSL.
2. Create a wallet. See "Creating a Wallet" for details.
3. Run `SSLConfigTool` in the middle-tier Oracle home, or multiple middle-tier Oracle homes, as shown in the following example for Windows:

   ```
   SSLConfigTool.bat -config_w_prompt -ptl_inv_pwd <ptl_inv_pwd> -opwd <orcladmin_pwd>
   ```

   Where:

   - `config_w_prompt` is passed to run `SSLConfigTool` in interactive mode.
   - `ptl_inv_pwd` is the portal invalidation password.
   - `opwd` is the Oracle administrator password. If no password is specified, you will be prompted to enter the password.

   Enter the following values when prompted by the script:

   - `y`, when prompted to configure your site to accept browser requests using the SSL protocol.
   - `y`, when asked if your Oracle HTTP Server accepts requests in SSL protocol.
4. Configure and register Web providers or Provider groups. See "Configuring and Registering Web Providers or Provider Groups Exposed Over SSL" for details.
5. Configure and register WSRP producers exposed over SSL. See "Configuring and Registering WSRP Producers Exposed Over SSL" for details.
6. Add the Web provider server certificate to the trusted certificate file. See "Augmenting the Portal Tools Trusted Certificate File" for details.
7. Enable access for Oracle Ultra Search. See "Enabling Access for Oracle Ultra Search" for details.
8. Re-register OracleAS Portal as an Oracle Ultra Search content source, and specify the SSL URL of OracleAS Portal. See "Re-Registering OracleAS Portal as an Oracle Ultra Search Content Source" for details.
9. Register the Ultra Search provider with OracleAS Portal. See "Registering the Ultra Search Provider with OracleAS Portal" for details.

Configuring SSL Throughout OracleAS Portal Manually

To manually configure SSL throughout OracleAS Portal, perform the following tasks:

- Enabling OracleAS Single Sign-On for SSL
- Creating a Wallet
- Securing the Oracle HTTP Server
- Securing OracleAS Web Cache
- Securing the Parallel Page Engine

- Specifying OracleAS Portal Published Address and Protocol
- Re-Registering the Oracle HTTP Server Partner Application
- Associating the Federated Portal Adapter with SSL
- Configuring and Registering Web Providers or Provider Groups Exposed over SSL
- Configuring and Registering WSRP Producers Exposed Over SSL
- Augmenting the Portal Tools Trusted Certificate File
- Enabling Access for Oracle Ultra Search
- Re-Registering OracleAS Portal as an Oracle Ultra Search Content Source
- Registering the Ultra Search Provider with OracleAS Portal

Enabling OracleAS Single Sign-On for SSL

Perform the steps described in "Configuring SSL to OracleAS Single Sign-On Manually" to enable OracleAS Single Sign-On for SSL.

Creating a Wallet

It is possible to share a single wallet across both the listener and origin server (and all other available ports in OracleAS Web Cache) if OracleAS Web Cache and the Oracle HTTP Server are on the same computer. Conversely, specific wallets may be created for each port. In this case the two servers and ports will be sharing the same wallet specified earlier.

In some cases, such as when the Oracle HTTP Server is on a different computer than OracleAS Web Cache, you need to create a separate wallet for the Oracle HTTP Server. For this situation, refer to the steps in "Creating a Wallet" to create the wallet for the Oracle HTTP Server.

In the default case, where the Oracle HTTP Server is on the same computer as OracleAS Web Cache, you can share the wallet between the two.

Securing the Oracle HTTP Server

You need to configure the Oracle HTTP Server as the OracleAS Web Cache's origin server to accept HTTPS-based communication. The Oracle HTTP Server implements SSL by the use of mod_ssl. As such, the configuration to use HTTPS is fairly straightforward.

The SSL configuration of the Oracle HTTP Server is defined within `ssl.conf`. This file may be edited directly or by using the Advanced Server Properties page under the Oracle HTTP Server node of the appropriate Oracle Application Server instance within the Oracle Enterprise Manager Administration pages. If you edit the files manually, it is recommended that you run the `dcmctl` utility with the following options to make sure that the file is synchronized with the Distributed Configuration Management (DCM) repository:

```
MID_TIER_ORACLE_HOME/dcm/bin/dcmctl updateConfig -ct ohs
```

1. Open `ssl.conf` in *`MID_TIER_ORACLE_HOME`*`/Apache/Apache/conf`.
2. Search for the following directives and change the values accordingly:

Table 6–20 Wallet Entries in ssl.conf

Default Entry	Updated Entry
SSLWallet file:	SSLWallet file:
MID_TIER_ORACLE_HOME/Apache/Apache/conf/ssl.wlt/default	/Directory where the wallet has been saved

3. In `ssl.conf` in *MID_TIER_ORACLE_HOME*`/Apache/Apache/conf`, verify that the default virtual host for SSL communication specifies the correct, preallocated port number for SSL. When creating the corresponding Web Cache site to server mappings, you should take note of the following properties in the `ssl.conf` file, for example:

```
Listen 4445

Port 4444
```

> **Note:** When setting up OracleAS Portal to communicate through HTTPS, it is common to configure both the middle and infrastructure tiers to operate in this mode. You must leave an HTTP port open on the infrastructure tier for OracleAS Portal to communicate with OracleAS Single Sign-On for External Application information. This call is made directly from the repository using the UTL_HTTP package.

4. Ensure that the start mode of the Oracle HTTP Server is set to `ssl-enabled` in *MID_TIER_ORACLE_HOME*`/opmn/conf/opmn.xml`. For example:

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

Securing OracleAS Web Cache

The sections that follow describe how to configure OracleAS Web Cache to accept SSL connections and forward SSL requests to the SSL-enabled origin server.

> **Note:** Changing OracleAS Web Cache settings (for example, Listening Port) can change the OracleAS Portal URL. If you do this, mobile settings need to be updated manually. See Section C.8, "Using the cfgiasw Script to Configure Mobile Settings" for more information.

Configuring the OracleAS Web Cache SSL Port To configure the OracleAS Web Cache SSL port, perform the following steps:

1. From the OracleAS Web Cache Administration page, click the **Listen Ports** link in the **Ports** section.
2. To add the SSL port, click **Add...** and enter the following information:

   IP Address: `ANY`

   Port Number: SSL port that Web Cache is listening on. This property maps to the port setting (Port number `4444`, as in the earlier example) in the HTTP server's `ssl.conf` file.

   Protocol: `HTTPS`

   Require Client-Side Certificate: No (unchecked)

   Wallet: Path to the directory where the SSL server certificate is stored
3. Click **Submit**.

For more information on the procedure in the preceding text, refer to the section on configuring HTTPS operations ports for the cache, in the *Oracle Application Server Web Cache Administrator's Guide*.

Adding the SSL Origin Server To add the SSL origin server:

1. From the OracleAS Web Cache administration page, click **Origin Servers** under **Origin Servers, Sites, and Load Balancing**.
2. Click **Add...** to add the SSL Origin Server.
3. Enter the information as follows:

   Host Name: Physical hostname of the computer in which Oracle HTTP Server is running

   Port: Oracle HTTP Server's SSL listen port. This maps to the Listen Parameter (Port number `4445`, as in the earlier example) in the `ssl.conf` file.

   Routing: Enable

   Capacity: 100

   Failover Threshold: 5

   Ping URL: /

   Ping Interval: 10

   Protocol: HTTPS
4. Click **Submit**.

For more information on the procedure in the preceding text, refer to the section on configuring Origin Server, Load Balancing, and Failover Settings, in the *Oracle Application Server Web Cache Administrator's Guide*.

Defining a Site for the Published SSL Host Name and Port As there is no out-of-box default SSL configuration, you need to add a site definition for the SSL-based Site that OracleAS Web Cache will be caching. To define a site, perform the following steps:

1. From the OracleAS Web Cache Administration page, click **Site Definitions** under **Origin Servers**, **Sites**, and **Load Balancing**.
2. Define a site where **Host Name** is the hostname seen by the browser, which would be the OracleAS Web Cache hostname.

3. Set **Port** to the HTTPS port addressed by the browser requests, which would be the OracleAS Web Cache SSL listen port (Port number `4444`, as in the earlier example).
4. Enter site information as follows:

   HTTPS Only Prefix: Leave blank

   Client-Side Certificate: Not required

   Default Site: Yes

   Create Alias from Site Name with/without www: No

   URL Path Prefix: Leave blank

   URL Parameters to Ignore: Leave blank
5. Click **Submit.**
6. Remove any sites that are no longer applicable to your modified configuration.

For more information on the procedure in the preceding text, refer to:

- The section on creating a site for HTTPS requests for the cache, in the *Oracle Application Server Web Cache Administrator's Guide*.
- The section on creating site definitions, in the *Oracle Application Server Web Cache Administrator's Guide*.

Adding Site to Server Mappings of the New Site to the Origin Server After adding a new site definition, you must add the site to server mapping for the newly defined site to the origin server. To do this:

1. From the OracleAS Web Cache Administration page, click **Site-to-Server Mapping** under **Origin Servers**, **Sites**, and **Load Balancing**.
2. Select the first option of the Site-to-Server Mapping table.
3. Click **Insert Above..** to bring up the **Edit/Add Site-to-Server Mapping** dialog window.
4. Select the **Site** you are mapping from the drop-down list, which should be the OracleAS Web Cache site with the SSL port.
5. Check the Origin Server that you added in the previous step, entitled "Adding the SSL Origin Server", to which requests should be routed for content.
6. Click **Submit** to close the dialog box and see the new mapping, as shown in Table 6–21 of the original page.

Table 6–21 Site-to-Server Mapping

Site			Origin Server		
Host Name	**Port**	**ESI Content Policy**	**Host Name**	**Port**	**Proxy**
`www.mycompany.com`	`4444`	Unrestricted	`www.mycompany.com`	`4445`	No

For more information on the procedure in the preceding text, refer to the section on mapping sites to origin servers, in the *Oracle Application Server Web Cache Administrator's Guide*

Adding Wallet Path for the Origin Server Wallet You must enter the wallet path in the **Origin Server Wallet** page, by performing the following steps:

1. From the OracleAS Web Cache Administration page, click **Origin Server Wallet** under **Origin Servers, Sites, and Load Balancing**.
2. Select the option for the Cache Name to change its wallet path.
3. Click **Edit Selected** to display the **Edit Origin Server Wallet** dialog window.
4. Enter a valid **Wallet Directory**. For example, use the wallet directory path specified on the **Listen Ports** page.
5. Click **Submit** to close the dialog window. The new wallet directory path is displayed in the table on the **Origin Server Wallet** page.

> **Note:** After you have performed all the steps to secure OracleAS Web Cache, you must restart the OracleAS Web Cache server using the OracleAS Web Cache Manager for the changes to take effect.

Securing the Parallel Page Engine

In this configuration, SSL is used throughout as the request comes to OracleAS Web Cache through HTTPS and the PPE loops back over HTTPS as well. The server specifies the chain that goes with its certificate. As long as the chain is valid and leads to a self-signed root certificate, it is validated without determining whether it is trusted, assuming that you have not loaded any trust points into it.

To implement this configuration, perform the following steps on the OracleAS Portal middle tier:

1. Open the `web.xml` file associated with the OC4J_Portal instance on the middle tier.

   ```
   MID_TIER_ORACLE_HOME\j2ee\OC4J_Portal\applications\portal\portal\
   WEB-INF\web.xml
   ```

2. Add `httpsports` in an additional `<init-param>` block in `web.xml`. This should point to the OracleAS Web Cache HTTPS listening port. For example:

   ```
   <servlet>
   <servlet-name>page</servlet-name>
      <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
      <init-param>
        <param-name>httpsports</param-name>
        <param-value>4443</param-value>
      </init-param>
    </servlet>
   ```

   > **Note:** If your current `web.xml` file contains the useScheme and usePort directives, you need to remove them. The configuration with SSL throughout should only use the `httpsports` directive.

3. (Optional) If you want the PPE to trust only specific certificates, add `x509certfile` in an additional `<init-param>` block in `web.xml`. The value of `x509certfile` is the absolute path to the text file containing the list of certificates which defines the group of "trusted" servers. For example:

   ```
   <servlet>
   <servlet-name>page</servlet-name>
   ```

```
  <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
    <init-param>
      <param-name>x509certfile</param-name>
      <param-value>C:\mySSLconfig\trustedCerts.txt</param-value>
    </init-param>
</servlet>
```

Note: If you choose not to implement `x509certfile`, the PPE trusts any SSL certificate.

Specifying OracleAS Portal Published Address and Protocol

To specify the published address for OracleAS Portal with the modified port for SSL, you need to use Oracle Enterprise Manager as follows:

1. Navigate to the Oracle Enterprise Manager 10*g* Application Server Control Console.
2. Click the Standalone Instance with the Oracle Application Server that is running the OracleAS Portal middle tier.
3. Click the OracleAS Portal system component.
4. Under **Administration**, click **Portal Web Cache Settings**.
5. For **Listen Port**, enter the OracleAS Web Cache SSL port number.
6. For **Listening Port SSL Enabled**, choose **Yes**.
7. Click **OK**. The OracleAS Portal schema is updated with the setting and the Oracle Enterprise Manager 10*g* target instance is updated to use HTTPS for its URL tests.

   If at a later time you choose to switch to HTTP, you would perform this same procedure and return Listening Port SSL Enabled to No.

Note: This procedure updates the settings in the `iasconfig.xml` file.

See Also: See Appendix A, "Using the Portal Dependency Settings Tool and File" for more information about `iasconfig.xml`.

Re-Registering the Oracle HTTP Server Partner Application

Run `ssoreg` to register the virtual host for which mod_osso facilitates single sign-on. The specific application URLs to be defined as partner applications within this site are defined in the file `osso.conf`. `ssoreg` is located on the middle tier in *`MID_TIER_ORACLE_HOME`*`/sso/bin`.

The following example shows the usage of `ssoreg` on UNIX:

```
MID_TIER_ORACLE_HOME/sso/bin/ssoreg.sh
  -oracle_home_path MID_TIER_ORACLE_HOME
  -site_name www.abc.com
  -config_mod_osso TRUE
  -mod_osso_url https://www.abc.com:4443
  -config_file MID_TIER_ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
  -admin_info cn=orcladmin
```

On Windows you must run the `ssoreg.bat` batch file instead. Refer to the information on creating partner applications in the *Oracle Application Server Single Sign-On Administrator's Guide*.

Associating the Federated Portal Adapter with SSL

The Federated Portal Adapter uses the Oracle HTTP Server rewrite rules to simplify URLs for registering providers. By default, these rewrite rules are only specified for HTTP communication. To set up the Federated Portal Adapter with SSL, perform the following steps:

1. Edit the virtual hosts section of your `ssl.conf` file, located in the *MID_TIER_ORACLE_HOME*`/Apache/Apache/conf` directory, as follows:

```
## SSL Virtual Host Context
##
#
# NOTE: this value should match the SSL Listen directive set previously in this
# file otherwise your virtual host will not respond to SSL requests.
#
<VirtualHost _default_:3011>
  #  General setup for the virtual host
  DocumentRoot /u01/app/oracle/product/as10g/Apache/Apache/htdocs
  ServerName host1.abc.com
  ServerAdmin you@your.address
  ErrorLog /u01/app/oracle/product/as10g/Apache/Apache/logs/error_log
  TransferLog "/u01/app/oracle/product/as10g/Apache/Apache/logs/access_log"
  Port 3001
  SSLEngine on
  SSLCipherSuite
SSL_RSA_WITH_RC4_128_MD5:SSL_RSA_WITH_RC4_128_SHA:SSL_RSA_WITH_3DES_EDE_CBC_
SHA:SSL_RSA_WITH_DES_CBC_SHA:SSL_RSA_EXPORT_WITH_RC4_40_MD5:S

SL_RSA_EXPORT_WITH_DES40_CBC_SHA
  SSLWallet
file:/u01/app/oracle/product/as10g/Apache/Apache/conf/ssl.wlt/default

  <Files ~ "\.(cgi|shtml)$">
   SSLOptions +StdEnvVars
  </Files>
  <Directory /u01/app/oracle/product/as10g/Apache/Apache/cgi-bin>
   SSLOptions +StdEnvVars
  </Directory>

        SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
        CustomLog /u01/app/oracle/product/as10g/Apache/Apache/logs/ssl_request_
log "%t %h %{SSL_PROTOCOL}x
%{SSL_CIPHER}x \"%r\" %b"

        RewriteEngine on
        RewriteOptions inherit

    </VirtualHost>
```

2. Run the following command to synchronize the manual configuration changes:

```
MID_TIER_ORACLE_HOME/dcm/bin/dcmctl updateconfig
```

3. Restart your Oracle Application Server instance:

```
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl stopall
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl startall
```

Configuring and Registering Web Providers or Provider Groups Exposed over SSL

Refer to "Configuring and Registering Web Providers or Provider Groups Exposed Over SSL" for details.

Configuring and Registering WSRP Producers Exposed Over SSL

Refer to "Configuring and Registering WSRP Producers Exposed Over SSL" for details.

Augmenting the Portal Tools Trusted Certificate File

Refer to "Augmenting the Portal Tools Trusted Certificate File" for details.

Enabling Access for Oracle Ultra Search

Refer to "Enabling Access for Oracle Ultra Search" for details.

Re-Registering OracleAS Portal as an Oracle Ultra Search Content Source

If you use Oracle Ultra Search to crawl your portal and you reconfigure SSL throughout OracleAS Portal, you must re-register OracleAS Portal as a content source with Oracle Ultra Search. Make sure that you specify the SSL URL for OracleAS Portal. See Section 8.2.4.2, "Registering OracleAS Portal as a Content Source" for details.

Registering the Ultra Search Provider with OracleAS Portal

To access the Oracle Ultra Search portlet, you must register the Ultra Search provider with OracleAS Portal. Make sure that you specify the SSL URL during registration. See Section 8.2.4.3, "Registering the Ultra Search Provider with OracleAS Portal" for details.

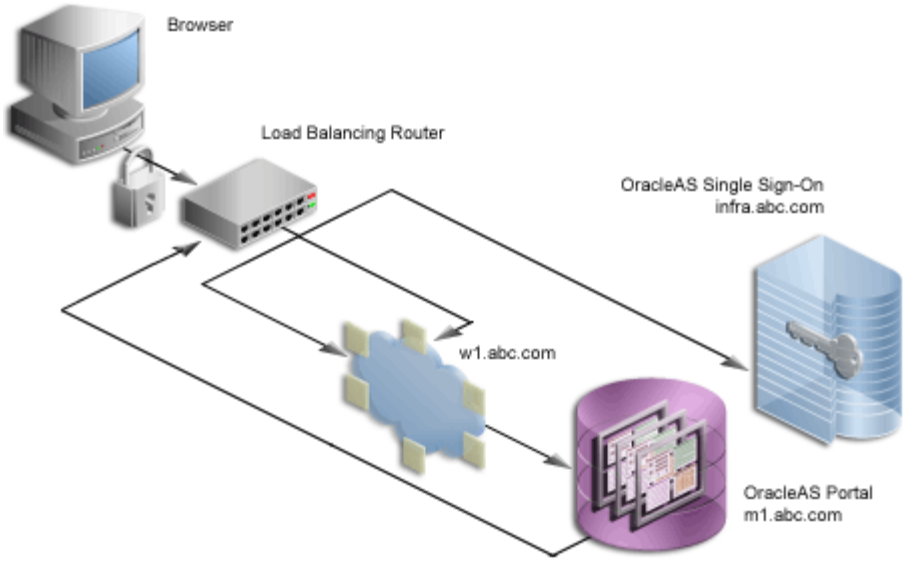
6.3.2.1.5 External SSL with Non-SSL Within Oracle Application Server

The previous configurations discuss how to configure OracleAS Portal in such a way that communications within Oracle Application Server are secured through SSL. In some cases, you may want to have OracleAS Portal set up such that the site is externally accessible through SSL URLs but the Oracle Application Server is internally running in non-SSL mode. Note that in this latter scenario, you need to have the SSL to non-SSL translation done either by a load balancing router (LBR) or an SSL accelerator. The section that follows outlines the steps you would use with an accelerator on an LBR or a reverse proxy server performing the SSL translation.

With this option, the SSL features of the OracleAS Security Framework are not used, but, instead, an external component is used for providing the SSL connection point. These external accelerators may be coupled with LBRs or reverse proxy servers. Oracle Application Server enables you to configure these external devices to provide SSL, thus allowing Oracle Application Server to use HTTP internally for the best performance.

Figure 6–19 shows a configuration where OracleAS Portal is externally accessible through SSL.

Figure 6–19 External SSL Only



Note: In a typical configuration, `w1.abc.com` and `m1.abc.com` would reside on the same computer, but for illustration purposes, they are separated here.

For the purposes of this procedure, assume the following:

- In this example, SSL acceleration is performed by the LBR, which also proxies page requests between the HTTP and HTTPS protocols and their ports. The location of the SSL end point will determine the target for the loopback requests. For example, if an SSL-enabled reverse proxy server front-ends a single protocol LBR, loopback requests will be sent to the LBR, while the published site is exposed by the reverse proxy server.
- Your load balancing router is running on `lbr.abc.com` and the LBR port used for accessing the site is `443`.
- OracleAS Web Cache is on computer `w1.abc.com` and the listening port is `7777`, the administration port is `4000`, and the invalidation port is `4001`.
- The Oracle HTTP Server is running on computer `m1.abc.com` and the port is `7778`.
- The port numbers used are example values only.

See Also: For more information, refer to:

- Section 5.3, "Configuring Multiple Middle Tiers with a Load Balancing Router"
- *Oracle Application Server Enterprise Deployment Guide*

After you have successfully performed the checks described in "Checks to Perform Before Configuring SSL", you can use either of the following two methods to configure OracleAS Portal with external SSL:

- Configuring External SSL Using SSLConfigTool
- Configuring External SSL Manually

Configuring External SSL Using SSLConfigTool

Refer to the section "SSL Configuration Tool and Its Limitations" before using `SSLConfigTool`.

To configure external SSL with non-SSL within Oracle Application Server using the SSL configuration tool, you must perform the following tasks:

1. Run SSLConfigTool
2. Configure Seeded Providers and Provider Groups
3. Enabling Access for Oracle Ultra Search

Run SSLConfigTool

Run `SSLConfigTool` in the middle-tier home, or multiple middle-tier Oracle homes, as shown here for Windows:

```
SSLConfigTool.bat -config_w_file <input_file_name> -ptl_inv_pwd <ptl_inv_pwd>
-opwd <orcladmin_pwd>
```

Where:

- `config_w_file` is used to run the tool in silent mode using the values specified in the `<input_file_name>` file. See the *Oracle Application Server Administrator's Guide* for details.
- `ptl_inv_pwd` is the portal invalidation password.
- `opwd` is the Oracle administrator password. If no password is specified, you will be prompted to enter the password.

For example:

```
SSLConfigTool.bat -config_w_file portal.config -ptl_inv_pwd invalidator -opwd administrator
```

The contents of the `portal.config` file for this example would include something similar to Example 6–3:

Example 6–3 Example Configuration File

```
<SSLConfig>
   <mid_tier>
      <virtual_address ssl="on" host="lbr.abc.com" port="443" inv_port="4001" ssl_
terminate="lbr"/>
      <lbr loopback_port="80"/>
      <wc/>
      <ohs>
         <servers>
            <server host="machine_6.us.oracle.com" port="7778" />
         </servers>
      </ohs>
   </mid_tier>
</SSLConfig>
```

Configure Seeded Providers and Provider Groups

1. To enable communication between OmniPortlet and Web Page Data Source using HTTP, configure OmniPortlet as follows:
   a. Open the `web.xml` file located in the directory, *`MID_TIER_ORACLE_HOME`*`/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF`.

b. Add the following context parameters to ensure that HTTP is used and not HTTPS that is used by the site:

```
<context-param>
    <param-name>useScheme<param-name>
    <param-value>HTTP</param-value>
</context-param>
<context-param>
    <param-name>usePort</param-name>
    <param-value>7777</param-value>
</context-param>
```

c. Save the `web.xml` file.

d. Run the following command to synchronize the manual configuration changes:

MID_TIER_ORACLE_HOME`/dcm/bin/dcmctl updateconfig`

e. Restart OC4J_Portal.

2. Update the seeded providers registration to use HTTP instead of HTTPS, by performing the following steps:

a. Log in to OracleAS Portal as the administrator (for example, `PORTAL`).

b. Click the **Administer** tab.

c. Click the **Portlets** subtab.

d. In the **Remote Providers** portlet, enter `WEBCLIPPING` in the **Name** field.

e. Click **Edit**.

f. Click the **Connection** tab.

g. In the **URL** field, change the URL from:

`https://lbr.abc.com:443/portalTools/webClipping/providers/webClipping`

to:

`http://lbr.abc.com:7777/portalTools/webClipping/providers/webClipping`

h. Click **Apply**.

i. Click **OK** to commit the change.

j. Repeat steps d through h but with the following exceptions:

– In Step d, enter `OMNIPORTLET` instead of `WEBCLIPPING`.

– In Step g, change the URL from:

`https://lbr.abc.com:443/portalTools/omniPortlet/providers/omniPortlet`

to:

`http://lbr.abc.com:7777/portalTools/omniPortlet/providers/omniPortlet`

3. Update the seeded provider group registration to use HTTP instead of HTTPS, by performing the following steps:

a. Log in to OracleAS Portal as the administrator (for example, `PORTAL`).

b. Click the **Administer** tab.

c. Click the **Portlets** subtab.

d. In the **Remote Provider Group** portlet, enter `oracle.ias.providers` in the **Name** field.

e. Click **Edit**.

f. Click the **Connection** tab.

g. In the **URL** field, change the protocol and port in the URL from `https://lbr.abc.com:443/` to `http://lbr.abc.com:7777/`.

h. Click **Apply**.

i. Click **OK** to commit the change.

j. Repeat steps d through h but with the following exception:

   In Step d, enter `oracle.sample.providers` instead of `oracle.ias.providers`.

When you register locally hosted Web providers or provider groups (such as the JPDK Sample provider), you need to register them using HTTP as the protocol, `lbr.abc.com` as the hostname, and `7777` as the port number. This restriction only applies to locally hosted Web providers or provider groups (that is, Web providers or provider groups running on the same middle tier as OracleAS Portal).

For example, to register the JPDK Sample provider, the URL is:

```
http://lbr.abc.com:7777/jpdk/providers/sample
```

However, if you want to create a new Web provider or provider group in the **Providers** tab in the **Portal Navigator**, you must first follow the steps described in the subsection "Configuring and Registering Web Providers or Provider Groups Exposed over SSL" in Section 6.3.2.1.4, "SSL Throughout OracleAS Portal".

Note: If your infrastructure is located on a separate computer than your OracleAS Portal middle tier, you need to add the following to your `/etc/host` file:

```
w1.w1.w1.w1   lbr.abc.com
```

where `w1.w1.w1.w1` is the IP Address of your LBR or reverse proxy server.

Enabling Access for Oracle Ultra Search

Refer to Enabling Access for Oracle Ultra Search for details.

Configuring External SSL Manually

To manually configure external SSL with non-SSL within Oracle Application Server, you must perform the following tasks:

- Configuring the Oracle Application Server Middle Tier
- Configuring Seeded Providers (Web Clipping and OmniPortlet) and Locally Hosted Web Providers
- Re-Registering the Oracle HTTP Server Partner Application
- Enabling Access for Oracle Ultra Search

Configuring the Oracle Application Server Middle Tier

You need to configure the Oracle Application Server middle tier to allow the underlying components to construct URLs based on the load balancing router's hostname, `lbr.abc.com`, and port (`443`). To do this:

1. Edit `httpd.conf` as follows:
   a. Access the Application Server Control Console.
   b. Click the link for the application server where OracleAS Portal is installed.
   c. Click the **HTTP Server** link.
   d. Click the **Administration** link.
   e. Click **Advanced Server Properties**.
   f. Select **httpd.conf**.
   g. Scroll to the bottom of the file and enter the following `LoadModule` directive:

      On UNIX:

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

      On Windows:

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

      For more information on mod_certheaders, refer to the *Oracle HTTP Server Administrator's Guide*.

   h. Add a Virtual Host definition. This must be inserted after the `LoadModule` directive, as follows:

      ```
      NameVirtualHost <OHS_computer_IP_address>:<OHS_listening_port>

      <VirtualHost <OHS_computer_IP_address>:<OHS_listening_port>>
        ServerName <portal_site_server_name>
        Port <ssl_listening_port_for_portal_site>
        SimulateHttps On
        RewriteEngine On
        RewriteOptions inherit
      </VirtualHost>
      ```

      For example:

      ```
      NameVirtualHost 123.45.67.8:7778

      <VirtualHost 123.45.67.8:7778>
        ServerName lbr.abc.com
        Port 443
        SimulateHttps On
        RewriteEngine On
        RewriteOptions inherit
      </VirtualHost>
      ```

   i. Define a second virtual host for internal use by Application Server Control Console. For example:

      ```
      <VirtualHost 123.45.67.8:7778>
        ServerName m1.abc.com
        Port 7777
        RewriteEngine On
        RewriteOptions inherit
      ```

```
</VirtualHost>
```

j. Click **Apply**.

k. When asked to restart Oracle HTTP Server, click **Yes**.

2. Configure the Parallel Page Engine. To do this, perform the following steps:

   a. Open the `web.xml` file, located in the directory *MID_TIER_ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF`.

   b. Make the following changes to the Page servlet section to attempt loopbacks using a different protocol and port than what is used by the site:

```
<servlet>
<servlet-name>page</servlet-name>
   <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
          <init-param>
             <param-name>useScheme</param-name>
             <param-value>http</param-value>
          </init-param>
          <init-param>
             <param-name>usePort</param-name>
             <param-value>7777</param-value>
          </init-param>
          <init-param>
             <param-name>httpsports</param-name>
             <param-value>443</param-value>
          </init-param>
</servlet>
```

   c. Save the `web.xml` file.

   d. Run the following command to synchronize the manual configuration changes:

```
MID_TIER_ORACLE_HOME/dcm/bin/dcmctl updateconfig
```

   e. Run the following commands to restart your Oracle Application Server instance:

```
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl stopall
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl startall
```

3. Ensure that DNS resolves `lbr.abc.com` to the LBR's IP address.

4. Register new URLs with OracleAS Portal using the LBR's hostname and port. To do this, edit `iasconfig.xml` (located by default in *ORACLE_HOME*`/portal/conf`) and specify a new *Farm* element that points to the LBR. A typical *Farm* element in `iasconfig.xml` looks like the bold text in Example 6–4:

Example 6–4 WebCacheDependency Entry in iasconfig.xml

```
<IASConfig XSDVersion="1.0">
    <IASFarm Name="Farm-1.lbr.abc.com" Host="lbr.abc.com">
        <WebCacheComponent ListenPort="443" InvalidationPort="4001"
        InvalidationUsername="invalidator" InvalidationPassword="welcome1"
        SSLEnabled="true"/>
    </IASFarm>
    <IASInstance Name="iAS.infra.abc.com" Host="infra.abc.com">
        <OIDComponent AdminPassword="@BVs2KPJEWC5a0l4n8lbTxUY="
        PortSSLEnabled="true" LDAPPort="3060" AdminDN="cn=orcladmin"/>
    </IASInstance>
```

```
    <IASInstance Name="iAS.w1.abc.com" Host="infra.abc.com">
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
    </IASInstance>
    <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
    SchemaPassword="@Beyh8p2bOWELQCsA5zRtuYc="
    ConnectString="cn=iasdb,cn=oraclecontext">
       <WebCacheDependency ContainerType="IASFarm" Name="Farm-1.lbr.abc.com"/>
        <OIDDependency ContainerType="IASInstance" Name="iAS.infra.abc.com"/>
       <EMDependency ContainerType="IASInstance" Name="iAS.w1.abc.com"/>
    </PortalInstance>
</IASConfig>
```

5. Run `ptlconfig` (typically located in the directory *`MID_TIER_ORACLE_HOME`*`/portal/conf`) as shown in the following example:

   ```
   ptlconfig -dad <portal name> -site -wc
   ```

6. To prevent access to Oracle Enterprise Manager from the outside, the Oracle Enterprise Manager link provided by OracleAS Portal needs to be changed back to point to the internal server. To do this, run `ptlconfig` (typically located in the directory `MID_TIER_ORACLE_HOME/portal/conf`) as shown in the following example:

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

7. From the OracleAS Web Cache administration page, click **Site Definitions** under **Origin Servers, Sites, and Load Balancing**.
8. Click **Add Site**.
9. Enter site information as follows:
   - **Host Name:** The published hostname and fully qualified domain of the external SSL accelerator device or reverse proxy server.
   - **Port Number:** The SSL port number, for example, `443` for the default SSL port.
   - **HTTPS Only Prefix:** Leave blank.
   - **Client-Side Certificate:** Not required.
   - **Default Site:** Yes.
   - **Create Alias from Site Name with/without www:** No.

   Refer to the *Oracle Application Server Web Cache Administrator's Guide* for detailed instructions on the configuration mentioned earlier.
10. Set up an alias for OracleAS Web Cache. In the configuration where the Parallel Page Engine loops back to OracleAS Web Cache and OracleAS Web Cache is listening on a different port than the load balancing router, loopback content gets cached with a URL key of `lbr.abc.com:7777`, whereas OracleAS Portal sends invalidation requests to invalidate URLs with the format `lbr.abc.com:443`. To get around this inconsistency, you need to set up an alias in OracleAS Web Cache to let it know that `lbr.abc.com:7777` and `lbr.abc.com:443` are the same, invalidation requests for one should invalidate requests for the other as well, and the cached content should also be leveraged based on this alias.
    a. Go to the Oracle Application Server Web Cache administration page and log in as the administrator.
    b. Click **Site Definitions**.

c. Select the radio button in the **Select** column that corresponds to the site for which the alias will be added, in this case `lbr.abc.com`.

d. Click **Add Alias**.

e. On the window that comes up, enter `lbr.abc.com` as the Host Name and `7777` as the port where `7777` is the value for `usePort` in the `web.xml` configuration file for the Parallel Page Engine.

f. Click **Submit**.

If the default HTTPS port `443` is chosen for a site configured with external SSL configuration, as in the preceding example, an additional alias needs to be defined in OracleAS Web Cache for the site `lbr.abc.com:443`, which should be `lbr.abc.com:80`. Follow the instructions for creating the alias as mentioned in Step 10 and replace port `7777` with `80`.

An alias for port `80` is needed because the "Host" header sent by the browser will be `lbr.abc.com` (without a port number appended to it). Because 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. It also uses this for cache key creation.

11. After adding a new site definition, you must add the site to server mapping for the newly defined site to the origin server. To do this:

a. In the navigation frame, select **Site-to-Server Mapping** under **Origin Servers, Sites, and Load Balancing**.

b. In the **Site-to-Server Mapping** page, select the first mapping in the table and click **Insert Above**.

c. In the **Edit/Add Site-to-Server Mapping** page, select the **Select from Site definitions** option and then select a site definition created in the previous step (`lbr.abc.com`).

d. In the **Select Application Web Servers** section, select the application server (`m1.abc.com`) specified in the **Origin Servers** page.

e. Click **Submit**.

f. Click **Apply Changes** on the top of the page.

g. In the **Cache Operations** page, click **Restart** to restart Web Cache.

To verify that the site has been mapped correctly, navigate to the **Site-to-Server Mapping** page, and ensure that `m1.abc.com` is mapped to the site `lbr.abc.com`.

For more information on the procedure in the preceding text, refer to the section on mapping sites to origin servers, in the *Oracle Application Server Web Cache Administrator's Guide*.

12. Configure your load balancing router, `lbr.abc.com`, to:

a. Accept requests on HTTPS port `443` and forward them to the OracleAS Web Cache port `7777` running on computer `w1.abc.com`, while converting HTTPS requests to HTTP.

b. Accept requests on HTTP port `7777` and forward them to the OracleAS Web Cache port `7777` running on computer `w1.abc.com`. This enables the loopback requests. The LBR's port `7777` should only be accessible from the middle tier. Your firewall may need to be configured separately to make this work. To test this, run the following command on the middle-tier server:

```
telnet lbr.abc.com 7777
```

You should not see any connection failure message when you run the `telnet` command.

c. Accept requests on HTTP port `4001` for the invalidation requests and forward them to the OracleAS Web Cache invalidation port `4001`. You must be able to connect to the LBR's port for invalidation requests from the OracleAS Metadata Repository. Your firewall may need to be configured separately to make this work. To test this, run the following command on the OracleAS Metadata Repository server:

```
telnet lbr.abc.com 4001
```

You should not see any connection failure message when you run the `telnet` command.

d. Facilitate communication from the middle tier to OracleAS Web Cache through the LBR. NAT-related configuration may be required for this. Refer to Section 5.3, "Configuring Multiple Middle Tiers with a Load Balancing Router" for more information on configuring NAT.

Note: Refer to Section 5.3.6, "Step 6: Configure Portal Tools and Web Providers (Optional)" for information on how this configuration might affect your Web providers.

---

e. Facilitate communication from the OracleAS Metadata Repository to OracleAS Web Cache for the invalidation requests through the LBR. NAT-related configuration may be required for this.

13. Optionally, re-register the Wireless gateway URL with the load-balancer's address. See Section 5.11, "Configuring OracleAS Wireless" for more information.

14. To enable monitoring of the load balancing router's front-end host and port settings for OracleAS Portal, you must edit `targets.xml` (located in *`MID_TIER_ORACLE_HOME`*`/sysman/emd/`) on **M1**, as follows:

a. Open `targets.xml` on **M1**, using a text editor.

b. Search for OracleAS Portal targets, that is, `TYPE="oracle_portal"`.

c. Edit the `PortalListeningHostPort` property, to point to the LBR. For example:

```
<Property NAME="PortalListeningHostPort" VALUE=https://lbr.abc.com:443/>
```

d. Save the changes to `targets.xml`.

e. Reload the targets in the Application Server Control Console:

On Solaris/Linux:

```
MID_TIER_ORACLE_HOME/bin/emctl reload
```

On Windows:

```
MID_TIER_ORACLE_HOME\bin\emctl reload
```

Configuring Seeded Providers (Web Clipping and OmniPortlet) and Locally Hosted Web Providers

Refer to the "Configure Seeded Providers and Provider Groups" subsection, in Section 6.3.2.1.5, "External SSL with Non-SSL Within Oracle Application Server" for the manual steps to be performed.

Re-Registering the Oracle HTTP Server Partner Application

Run `ssoreg` to register the virtual host for which mod_osso facilitates single sign-on. The specific application URLs to be defined as partner applications within this site are defined in the file `osso.conf`. `ssoreg` is located on the middle tier in *`MID_TIER_ORACLE_HOME`*`/sso/bin`.

The following example shows the usage of `ssoreg` on UNIX:

```
MID_TIER_ORACLE_HOME/sso/bin/ssoreg.sh
  -oracle_home_path MID_TIER_ORACLE_HOME
  -site_name lbr.abc.com
  -config_mod_osso TRUE
  -mod_osso_url https://lbr.abc.com
  -config_file MID_TIER_ORACLE_HOME/Apache/Apache/conf/osso/osso.conf
  -admin_info cn=orcladmin
  -virtualhost
```

On Windows you must run the `ssoreg.bat` batch file instead. Refer to the information on creating partner applications in the *Oracle Application Server Single Sign-On Administrator's Guide*.

Enabling Access for Oracle Ultra Search

For Oracle Ultra Search to access secure Web sites, you need to import certificates into the crawler's trust store and the Oracle Containers for J2EE (OC4J) JVM's trust store on the infrastructure instance.

By default, the OC4J JVM recognizes certificates of well-known certificate authorities. However, if the secure portal instance uses a self-signed certificate or a certificate signed by an unknown certificate authority, then you must import the portal's certificate into the OC4J JVM's truststore. The OC4J JVM default truststore is located at *`ORACLE_HOME`*`/jdk/jre/lib/security/cacerts`.

To add the required certificate to the trust store, perform the following steps on the infrastructure instance:

1. Change directory to *`ORACLE_HOME`*`/jdk/jre/lib/security/` on the infrastructure.
2. Create a backup of the truststore file `cacerts`, for example, `cacerts.bak`.
3. Execute the following command to add the required certificate to the trust store:

   ```
   $ORACLE_HOME/jdk/bin/keytool -import -alias <aliasName> -file <root_
   certificate_file_name> -trustcacerts -v -keystore $ORACLE_
   HOME/jdk/jre/lib/security/cacerts
   ```

   Note: Use any arbitrary value for `-alias`, but ensure that it is unique for each certificate.

4. Provide the trust store password, and type "Yes", when prompted for confirmation.

Note: The preceding steps also need to be performed on the OracleAS Infrastructure containing the Oracle Ultra Search crawler.

6.3.2.1.6 Configuring and Registering Web Providers, Provider Groups, and WSRP Producers Exposed Over SSL

This section describes how you can configure OracleAS Portal to provide access to Web providers, Provider groups, and WSRP producers exposed over SSL. This section contains the following topics:

- Configuring and Registering Web Providers or Provider Groups Exposed Over SSL
- Configuring and Registering WSRP Producers Exposed Over SSL

Configuring and Registering Web Providers or Provider Groups Exposed Over SSL

To register a Web provider, which is exposed over SSL, you must have a copy of the root certificate of the certificate authority used by the Web provider. If the Web provider is using an unknown or uncommon certificate authority, you need to add the appropriate root certificate (using Base-64 encoded X.509 format) to the set of trusted certificates recognized by the Oracle Database hosting the OracleAS Metadata Repository containing the OracleAS Portal schema. To configure Web providers or provider groups exposed over SSL, perform the following steps:

1. Change directory to *ORACLE_HOME*`/javavm/lib/security/` on the Oracle home containing the Oracle Database hosting the OracleAS Metadata Repository containing the OracleAS Portal schema.
2. Create a backup of the truststore file `cacerts`, for example, `cacerts.bak`.
3. Execute the following command to add the required certificate to the trust store:

```
$ORACLE_HOME/jdk/bin/keytool -import -alias <aliasName> -file <root_
certificate_file_name> -trustcacerts -v -keystore $ORACLE_
HOME/javavm/lib/security/cacerts
```

Note: Use any arbitrary value for `-alias`, but ensure that it is unique for each certificate.

4. Provide the trust store password, and type **Yes**, when prompted for confirmation.

Note: If your portal schema is located in an OracleAS RepCA database and if the release of that Oracle Database is earlier than 10*g* (10.1.0.*x*), then these steps do not need to be performed.

See Also:

- The subsection "Securing the Parallel Page Engine" in Section 6.3.2.1.4, "SSL Throughout OracleAS Portal".
- The subsection "Configure Seeded Providers and Provider Groups" in Section 6.3.2.1.5, "External SSL with Non-SSL Within Oracle Application Server".

Configuring and Registering WSRP Producers Exposed Over SSL

To configure WSRP producers exposed over SSL, perform the following steps:

1. In a Web browser, enter the WSDL URL of your WSRP producer to ensure that it is working. If the WSDL definition does not appear in the browser, then the deployment of your WAR file must have failed. Refer to the section on diagnosing Java portlet problems in the *Oracle Application Server Portal Developer's Guide*.
2. Modify your WSDL file and make it available over HTTP. To do this, perform the following steps:

   Note: The following steps are for a setup where the WSRP ports are generated with the HTTP protocol because HTTP was used for requesting the WSDL URL. However, if you are accessing WSDL using HTTP and the internal URLs are referenced using HTTPS, then you can skip Step 2.

   a. In a Web browser, enter the HTTPS WSDL URL. For example:

      ```
      https://host:port/context-root/portlets?WSDL
      ```

      Each port in the WSDL definition should be displayed with an HTTPS location. For example:

      ```
      <wsdl:port binding="bind:WSRP_v1_Markup_Binding_SOAP"
      name="WSRPBaseService">
      <soap:address
      location="https://host:port/context-root/portlets/WSRPBaseService"/>
      </wsdl:port>
      ```

   b. Save a copy of the WSDL definition to a file on your application server in a location where it can be accessed externally over HTTP. For example, the *ORACLE_HOME*/Apache/Apache/htdocs/ directory of your Oracle Application Server middle tier installation. When you register the producer in OracleAS Portal, use the location of this WSDL for your **WSDL URL** on the **Define Connection** page of the registration. The format of a WSDL URL is as follows:

      ```
      http://<host>:<port>/<Savedxml.xml>?WSDL
      ```

   c. If the ports are not listed with HTTPS locations, then you must change them manually before proceeding. You can do this by saving the XML to a file from the browser and opening it in a text editor.
3. To register a WSRP producer, which is exposed over SSL, you must have a copy of the root certificate of the certificate authority used by the WSRP producer. Check if the root certificate exists in the set of trusted certificates recognized by the Oracle Database hosting the OracleAS Metadata Repository containing the OracleAS Portal schema. To check if a root certificate already exists, register a sample JPDK provider using an SSL-enabled URL. Access the home page of the SSL-enabled portal. If a certificate security alert dialog box appears, then skip Step 4.
4. If the WSRP producer is using an unknown or uncommon certificate authority, then you need to add the appropriate root certificate (using Base-64 encoded X.509 format) to the set of trusted certificates. To add a root certificate of the certificate authority used by the WSRP producer to the truststore, perform the following steps:

a. Change directory to *ORACLE_HOME*/javavm/lib/security/ on the Oracle home containing the Oracle Database hosting the OracleAS Metadata Repository containing the OracleAS Portal schema.

b. Create a backup of the truststore file cacerts, for example, cacerts.bak.

c. Execute the following command to add the required certificate to the trust store:

```
$ORACLE_HOME/jdk/bin/keytool -import -alias <aliasName> -file <root_
certificate_file_name> -trustcacerts -v -keystore $ORACLE_
HOME/javavm/lib/security/cacerts
```

Notes:

- Use any arbitrary value for -alias, but ensure that it is unique for each certificate.
- See "Adding the Trusted Root Certificate to the Wallet (HTTPS)" for details on the location of the root certificate.

---

d. Provide the trust store password, and type **Yes**, when prompted for confirmation.

e. (Optional) If you want the PPE to trust only specific certificates, then add x509certfile in an additional <init-param> block in the file, *MID_TIER_ORACLE_HOME*\j2ee\home\applications\portletapp\wsrp-samples\WEB-INF\web.xml. The value of x509certfile is the absolute path to the text file containing the list of certificates which defines the group of *trusted* servers. For example:

```
<servlet>
   <servlet-name>page</servlet-name>
   <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
      <init-param>
         <param-name>x509certfile</param-name>
         <param-value>C:\mySSLconfig\trustedCerts.txt</param-value>
      </init-param>
</servlet>
```

The contents of a typical trustedCerts.txt file would look as shown in Example 6–5.

Example 6–5 Sample File Containing a List of Certificates

```
-----BEGIN CERTIFICATE-----
MIICPDCCAaUCEDJQM89Q0VbzXIGtZVxPyCUwDQYJKoZIhvcNAQECBQAwXzELMAkGA1UEBhMCVVMx
FzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFzcyAxIFB1YmxpYyBQcmltYXJ5
IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDTk2MDEyOTAwMDAwMFoXDTIwMDEwNzIzNTk1OVow
XzELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMTcwNQYDVQQLEy5DbGFzcyAx
IFB1YmxpYyBQcmltYXJ5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDlGb9to1ZhLZlIcfZn3rmN67eehoAKkQ76OCWvRoiC5XOooJskXQ0fzGVuDLDQ
VoQYh5oGmxChc9+0WDlrbsH2FdWoqD+qEgaNMax/sDTXjzRniAnNFBHiTkVWaR94AoDa3EeRKbs2
yWNcxeDXLYd7obcysHswuiovMaruo2fa2wIDAQABMA0GCSqGSIb3DQEBAgUAA4GBAEtEZmBoZOSY
G/OwcuaViXzde7OVwB0u2NgZ0C00PcZQmhCGjKo/O6gE/DdSlcPZydvN8oYGxLEb8IKIMEKOF1Ac
ZHq4PplJdJf8rAJD+5YMVgQlDHx8h50kp9jwMim1pN9dokzFFjKoQvZFprY2ueC/ZTaTwtLXa9ze
WdaiNfhF
-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
MIICPTCCAaYCEQC6WslMBTuS1qe2307QU5INMA0GCSqGSIb3DQEBAgUAMF8xCzAJBgNVBAYTAlVT
MRcwFQYDVQQKEw5WZXJpU2lnbiwgSW5jLjE3MDUGA1UECxMuQ2xhc3MgMiBQdWJsaWMgUHJpbWFy
eSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw05NjAxMjkwMDAwMDBaFw0wNDAxMDcyMzU5NTla
MF8xCzAJBgNVBAYTAlVTMRcwFQYDVQQKEw5WZXJpU2lnbiwgSW5jLjE3MDUGA1UECxMuQ2xhc3Mg
MiBQdWJsaWMgUHJpbWFyeSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTCBnzANBgkqhkiG9w0BAQEF
AAOBjQAwgYkCgYEAtlqLow1qI4OAa885h/QhEzMGTCWi7VUSl8WngLn6g8EgoPovFQ18oWBrfnks
+gYPOq72G2+x0v8vKFJfg31LxHq3+GYfgFT8t8KOWUoUV0bRmpO+QZEDuxWAk1zr58wIbD8+s0r8
/0tsI9VQgiZEGY4jw3HqGSRHBJ51v8imAB8CAwEAATANBgkqhkiG9w0BAQIFAAOBgQC2AB+TV6QH
p0DOZUA/VV7t7/pUSaUw1iF8YYfug5MLv7Qz8pisnwa/TqjOFIFMywROWMPPX+5815pvy0GKt3+B
uP+EYcYnQ2UdDOyxAArdG6S7x3ggKLKi3TaVLuFUT79guXdoEZkj6OpS6KoATmdOu5C1RZtG644W
78QzWzM91Q==
-----END CERTIFICATE-----
```

Note: If you choose not to implement `x509certfile`, then the PPE trusts any SSL certificate.

f. Restart your Oracle Application Server instance:

```
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl stopall
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl startall
```

g. Register the WSRP producer, by specifying the WSDL defined at the location determined in Step 2. The URL used here must be HTTP based, and not HTTPS. For example:

```
http://<host>:<port>/myProducerWsdl.xml
```

6.3.2.2 Securing the Connection to Oracle Internet Directory (Optional)

In Section 6.3.2.1, "Configuring SSL for OracleAS Portal", we were mainly concerned with the HTTP-based network hops. However, you can also secure the network connection to the Oracle Internet Directory itself, which is LDAP-based communication. In this case the Oracle Internet Directory should be configured to use LDAP over SSL (LDAPS). You can find further information about configuring the Oracle Internet Directory for LDAPS in the *Oracle Internet Directory Administrator's Guide*.

Once Oracle Internet Directory is configured to use SSL, you must update the OracleAS Portal schema to use the new port on the LDAP server. To perform this step, you run the SQL script, `secupoid.sql`, located in *`MID_TIER_ORACLE_HOME`*`/portal/admin/plsql/wwc`. This script allows for the setting of the following Oracle Internet Directory related parameters:

- Directory Host Name
- Directory Port
- Application Directory Password
- SSL Settings

When you run the script, it displays the current settings and gives you the ability to change them accordingly. In this case, you want to set the following:

```
use_ssl_to_connect_to_ldap=Y
```

The script will then give you the option of automatically refreshing OracleAS Portal's Oracle Internet Directory cache. Refer to Section C.3, "Using the secupoid.sql Script" for more information.

> **Note:** From 10*g* Release 2 (10.1.2) onwards, you can optionally install OracleAS Portal using LDAPS rather than having to implement it after installation.

6.3.2.3 Post-Installation Security Checklist

After OracleAS Portal is installed, you should consider performing the following steps to complete the security configuration:

- Safeguard Passwords for Lightweight OracleAS Portal Users
- Remove Unnecessary Objects
- Review Default Seeded Privileges
- Revoke Public Access to Provider Components
- Control Access to Administration Pages
- Protect PL/SQL Packages
- Consider SSL
- Consider LDAP over SSL for Oracle Internet Directory Connections
- Change the Application Entity Password
- Configure Oracle Enterprise Manager 10g to Monitor OracleAS Portal Running in SSL Mode

6.3.2.3.1 Safeguard Passwords for Lightweight OracleAS Portal Users Unscrupulous users might try to learn the passwords of your default users, which could result in an account lock. This lock can be released from the server, but it is far better that you not depend on the default user accounts for administrative purposes. To safeguard the passwords for these accounts do the following:

1. Immediately change the default passwords for all of the following default users:
   - `PORTAL`
   - `PORTAL_ADMIN`
2. Create new lightweight administrator accounts with the same access rights as the default users, and set the account termination date in OracleAS Single Sign-On for the default users. Alternatively, you can also deselect the **Allow User To Log In** setting in the **Edit User** page for the default users.
3. Once you have disabled login or changed the passwords for the default users, try logging in to the portal as the default users with the default passwords to ensure that your changes have been successful.

6.3.2.3.2 Remove Unnecessary Objects To prevent users from entering your portal through obsolete or unnecessary pages, you should remove any unused objects from your OracleAS Portal and database environment. For example:

- Delete page groups that are no longer in use.
- Delete OracleAS Portal providers that are no longer in use.

6.3.2.3.3 Review Default Seeded Privileges When OracleAS Portal is installed, the seeded groups listed in Table 6–2 are provisioned with a set of privileges that are typically required by users in those roles. You should review these initial set of privileges to ensure that they are consistent with your security policy.

Users or groups can obtain privileges from one of the following sources:

- OracleAS Portal access control entries
- Oracle Internet Directory privilege groups

To edit the privileges granted through OracleAS Portal access control entries, you edit the user or group profile from the **Administer** tab with the User Profile Portlet or Group Profile Portlet. Click the User or Group Profile dialog's **Privilege** tab. You can revoke or assign privileges from this list.

To edit the privileges granted through Oracle Internet Directory privilege groups, use the User Portlet or Group Portlet to edit the User or Group in Oracle Internet Directory. Select or deselect the check marks by the Privilege Assignment list to grant or revoke the appropriate privileges in Oracle Internet Directory.

Privileges granted to the AUTHENTICATED_USERS group are given to any user that logs on to OracleAS Portal through OracleAS Single Sign-On upon successful authentication. This is the group that you will want to establish with the default privileges for all your logged in users.

For example, if you do not want authenticated users to be able to create groups, then edit the AUTHENTICATED_USERS group through the Group Portlet and remove the check mark beside **Allow group creation** under Privilege Assignment.

6.3.2.3.4 Revoke Public Access to Provider Components In some cases, OracleAS Portal provider components may give users the option to view or modify records in application tables. To tighten security, you should revoke public access from such components if it is unnecessary. You can also use a menu component with specific access rights on the menu options to more tightly control application access.

6.3.2.3.5 Control Access to Administration Pages To prevent users who should not have access to administration interfaces from entering administration pages, you should ensure that you control access rights for the following page groups and the pages they contain:

- Portal Design-Time Pages is the page group that contains the OracleAS Portal Home Page, and the Builder and Navigator pages.
- Portlet Repository

To control access to the page groups mentioned earlier, perform the following steps:

1. In the Navigator, click **Page Groups**.
2. Click **Edit Properties** next to the page group for which you want to change the access settings.
3. Click the **Access** tab.
4. Grant `MANAGE ALL` to specific users or to certain groups. For example, `DBA`, `PORTAL_ADMINISTRATORS`, `PORTAL_DEVELOPERS`, and your own groups.
5. When you are done, click **OK**.

To control access to individual administration pages in these page groups, perform the following steps:

1. In the Navigator, click **Page Groups**.

2. Click **Contents** next to the page group that contains the pages on which you want to change the access settings.
3. Click **Pages**.
4. Click **Properties** next to the page for which you to change the access settings.
5. Click the **Access** tab.
6. Grant `MANAGE ALL` to specific users or to certain groups. For example, `DBA`, `PORTAL_ADMINISTRATORS`, `PORTAL_DEVELOPERS`, and your own groups.
7. When you are done, click **OK**.

Note: The **Builder** page is the root page of the Portal Design-Time Pages page group. To alter its access settings, you must click **Edit Root Page** next to the Portal Design-Time page group.

6.3.2.3.6 Protect PL/SQL Packages The default installation protects standard database procedures that are granted to `PUBLIC`, for example `dbms_*`, `utl_*`, and so on. If you write your own PL/SQL packages, which are granted to `PUBLIC`, and do not want to provide access to these packages through a Web browser, then refer to the chapter "Securing Application Database Access Through mod_plsql" in the *Oracle Application Server mod_plsql User's Guide*.

6.3.2.3.7 Consider SSL If your portal contains confidential information, you should consider configuring it with SSL. See Section 6.3.2.1, "Configuring SSL for OracleAS Portal" for the available SSL configuration options.

6.3.2.3.8 Consider LDAP over SSL for Oracle Internet Directory Connections By default, OracleAS Portal connects to the directory using LDAP without SSL. If the directory server is configured for an SSL port, though, OracleAS Portal can be configured to use LDAP over SSL, also known as LDAPS.

See Also: *Oracle Internet Directory Administrator's Guide*

To configure OracleAS Portal to use SSL to connect to the directory, you must run the `secupoid.sql` script, located in *`ORACLE_HOME`*`/portal/admin/plsql/wwc`. This script enables you to change the following OracleAS Portal configuration parameters related to the directory:

- Directory hostname
- Directory port
- Application directory password
- SSL setting

When you install OracleAS Portal, it is automatically configured with a directory server. However, you may want to change some settings, such as whether to use SSL, after installation. To change to an SSL connection for the directory, simply run the `secupoid.sql` script in the PORTAL schema to specify the LDAPS port instead of the LDAP port, and indicate that you want to use SSL.

Running the secupoid.sql script

The section that follows shows a sample execution of `secupoid.sql` from SQL*Plus.

In the example, the directory was initially configured to run LDAP on port `389`. Later, an LDAPS port was activated on `636`. Because the server name does not change, we retain the old value, update the port, and indicate that we want to use SSL by setting the **Use SSL?** value to **Y**. When you run the script, it displays the current configuration and lets you replace any of the configurable settings. The script also enables you to update OracleAS Portal's directory cache after running it. Because activating SSL does not change any of the directory information cached by OracleAS Portal, it is not usually necessary to refresh the cache in this case.

```
SQL> @secupoid
Current Configuration
--------------------
OID Host: oid.domain.com
OID Port: 389
Application DN:
orclApplicationCommonName=PORTAL.040820.123756.096286000,cn=Portal,cn=Products,cn=OracleContext
Application Password: 3E8C2D1B87CB61011757239C5AA9B390
Use SSL? N

PL/SQL procedure successfully completed.

Updating OID Configuration Entries
Press [Enter] to retain the current value for each parameter
For SSL Connection to LDAP, specify "Y"es or "N"o
-----------------------------------------------
Enter value for oid_host:
Enter value for oid_port: 636
Enter value for app_password:
Enter value for use_ssl_to_connect_to_ldap: Y
Enter value for refresh_with_new_settings: N

PL/SQL procedure successfully completed.

No errors.
```

After executing the script, OracleAS Portal is configured for LDAPS access of the directory. Refer to Section C.3, "Using the secupoid.sql Script" for more information.

6.3.2.3.9 Change the Application Entity Password OracleAS Portal never passes a user's password to the directory. Only OracleAS Single Sign-On performs that task. However, OracleAS Portal authenticates itself to the directory through its application entity and password.

If you want to change the application entity's password, you need to first change its entry in the directory, using command line utilities or the Oracle Directory Manager. To locate the application entry in the directory, you need its DN, which is reported by the `secupoid.sql` script. By default, OracleAS Portal's application entry is:

```
orclApplicationCommonName=PORTAL.040820.123756.096286000,cn=Portal,cn=Products,cn=OracleContext
```

To change the password, you set the userPassword attribute for the application entry to the new password.

After you have changed the password in the directory, you run `secupoid.sql` script in the PORTAL schema and specify the new password there, too. Running the script enables OracleAS Portal to encrypt the password and store it for retrieval when it needs to connect to the directory.

Refer to Section C.3, "Using the secupoid.sql Script" for more information about the `secupoid.sql` script.

See Also: Section 6.1.6.2.1, "Directory Entries in Oracle Internet Directory for OracleAS Portal", for more information about the application entity.

6.3.2.3.10 Configure Oracle Enterprise Manager 10*g* to Monitor OracleAS Portal Running in SSL Mode To enable Enterprise Manager to monitor OracleAS Portal running in either the mixed SSL mode or fully SSL-enabled mode, you must perform additional configuration.

To do this, follow the instructions for configuring Oracle Enterprise Manager 10*g* to monitor SSL-Enabled Targets, provided in the *Oracle Application Server Administrator's Guide*.

6.3.3 Configuring OracleAS Portal Options for Database Security

Fine-grained access controls and secure application contexts add a new dimension to your ability to secure your data in the database.

Fine-grained access control is sometimes referred to as virtual private database or row level security. Fine-grained access control in the Oracle Database 10*g* is the ability to dynamically attach, at run time, a predicate (WHERE clause) to any and all queries issued against a database table or view. This feature gives you the ability to procedurally modify the query at run time. You may evaluate who is running the query, where they are running the query from, when they are running the query and develop a predicate given those circumstances. With the use of application contexts, you may securely add additional information to the environment (such as an application role the user may have) and access it in your procedure or predicate as well.

As an example of fine-grained access control, you might have a security policy that determines what rows different groups of people may see. Your security policy will develop a predicate based on who is logged in and what group they are in.

You will find additional information about fine-grained access and application contexts on Portal Center, http://portalcenter.oracle.com/.

7

Monitoring and Administering OracleAS Portal

This chapter provides information about the monitoring and administration tools that are available, and how to use them to successfully monitor and administer OracleAS Portal.

You can monitor and administer OracleAS Portal through the Oracle Enterprise Manager 10*g* Grid Control Console, or the Oracle Enterprise Manager 10*g* Application Server Control Console. Additionally, you can view OracleAS Portal Analytics to monitor OracleAS Portal performance and analyze OracleAS Portal access characteristics.

> **See Also:** For additional OracleAS Portal monitoring and administration information, see the OracleAS Portal Administration page on the Oracle Technology Network (OTN), at `http://www.oracle.com/technology/products/ias/portal/administration_10g1014.html`.

This chapter contains the following sections:

7.1 Using the Grid Control Console

The Oracle Enterprise Manager 10*g* Grid Control Console is a full enterprise management framework consisting of the Oracle Management Service, Oracle Management Agent, and Oracle Management Repository. In the Grid Control Console, you can:

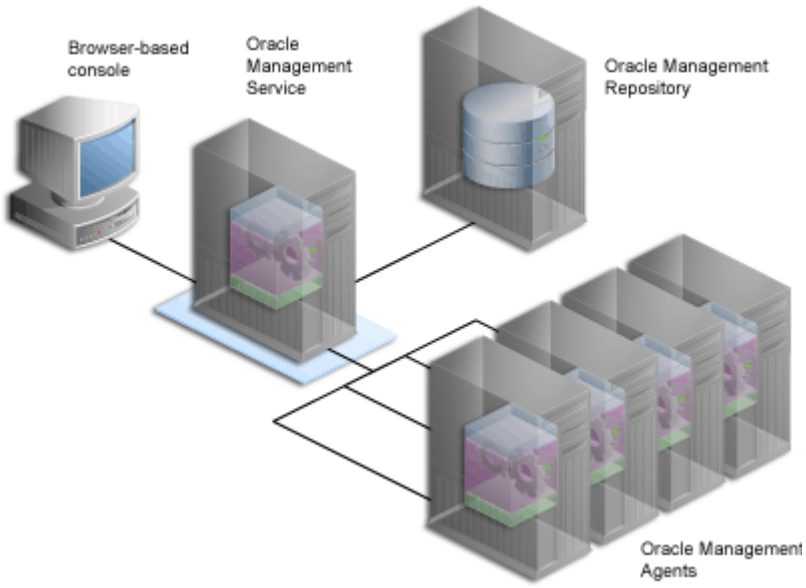
- Manage targets in your environment
- Monitor historical trends
- Configure alerts
- View diagnostics
- Monitor application performance

- Manage enterprise configuration

> **Note:** For more information, see the *Oracle Enterprise Manager Grid Control Installation and Basic Configuration* guide.

Oracle Enterprise Manager 10*g* uses a Web-based architecture that is robust, reliable, globally scalable, and easy to deploy and operate within today's Internet-enabled environments. This architecture (shown in Figure 7–1) uses four integrated software components, three of which (Oracle Management Service, Oracle Management Repository, and Oracle Management Agent) run behind the scenes, gathering, organizing, and routing management data. The Browser-based console provides a Web-based user interface so you can manage the information from a standard Web browser.

Figure 7–1 Overview of Oracle Enterprise Manager 10g Grid Control Console Components



The Oracle Enterprise Manager 10*g* Grid Control Console ships with Oracle Application Server, but must be installed separately. In the case of OracleAS Portal, the Grid Control Console can be used for monitoring, and tracking historical trends, but not for configuration.

You can access the Grid Control Console by navigating to the following URL: `http://<hostname.domain>:<port>/em/`. For example, `http://myhost.mycompany.com:7777/em/`. You can find the URL for your Grid Control Console in `setupinfo.txt`. This text file is saved to the following location after you install the Oracle Application Server:

On UNIX: *`ORACLE_HOME`*`/install/setupinfo.txt`

On Windows: *`ORACLE_HOME`*`\install\setupinfo.txt`

You must then log in using a valid Grid Control Console user name and password combination with privileges to access the OracleAS Portal targets you intend to monitor.

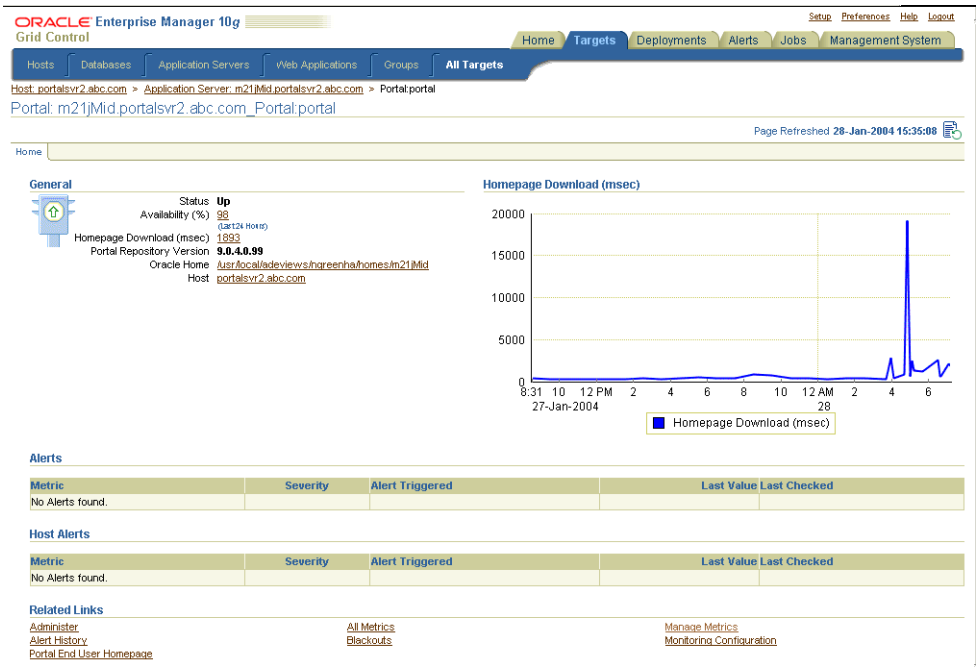
After logging on successfully, the Oracle Enterprise Manager 10*g* Grid Control Console home page is displayed.

To drill down to the application server level, click the **Targets** tab, and then the **Application Servers** subtab. Select the application server that you want to monitor from the list of available application servers. The home page for the selected application server is then displayed.

Note: In the **Related Links** section at the bottom of Oracle Application Server component home pages, you can click the **Administer** link to go to the Oracle Enterprise Manager 10*g* Application Server Control Console and perform monitoring, and administrative tasks. When you go to the Application Server Control Console, you are prompted to login as the `ias_admin` user.

From the application server home page, you can click any of the components listed to get detailed information. For example, if you click the **Portal** component, if listed, the OracleAS Portal target page is displayed as shown in Figure 7–2.

Figure 7–2 Grid Control Console - Portal Target Page



On the Portal target page, in addition to availability information, you can monitor the average home page download time on a chart.

The Grid Control Console helps you in:

- Monitoring Historical Trends
- Comparing Metrics from Multiple Portal Targets
- Setting Up Notifications for OracleAS Portal Metrics
- Setting OracleAS Portal Metric Thresholds
- Viewing Recent Alerts

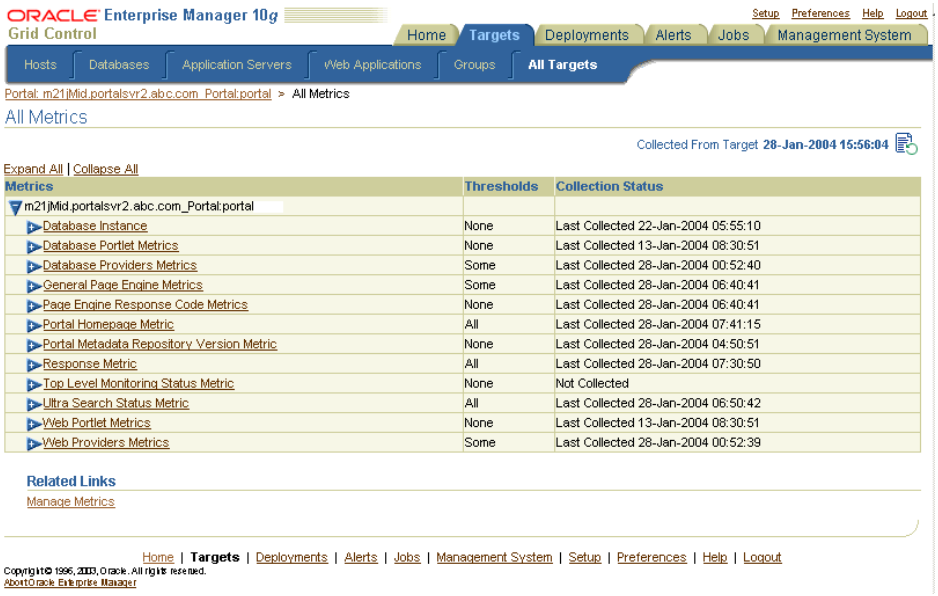
- Using Web Applications for Application Performance Monitoring

7.1.1 Monitoring Historical Trends

In the Grid Control Console, you can look at various OracleAS Portal metrics collected over a specific time period. The range of metrics which are collected are configured (by default) when the Management Agent is installed.

Figure 7–3 shows a list of the kinds of OracleAS Portal metrics you can monitor.

Figure 7–3 Grid Control Console - OracleAS Portal Metrics

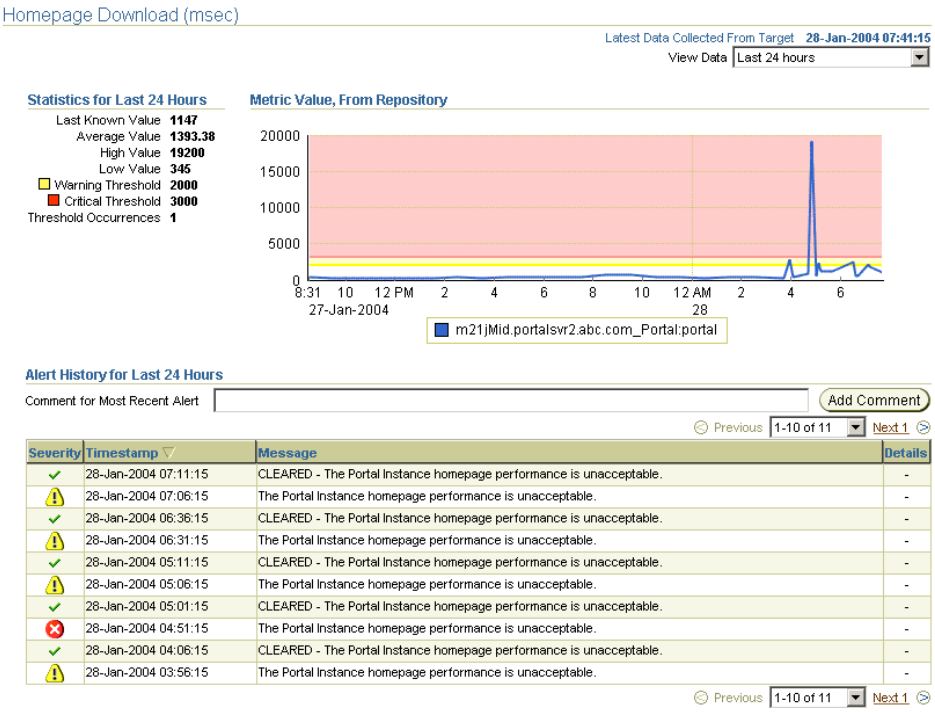


You can use the OracleAS Portal metrics to monitor historical trends. For example, if you want to see how your site has performed over the previous 31 days, follow these steps:

1. Navigate to the Grid Control Console home page.
2. Click the **Targets** tab, and then the **Application Servers** tab.
3. Choose the application server of interest.
4. From the **Components** table, select the **Portal** target.
5. Click the **All Metrics** link.
6. Expand the **Portal Homepage Metric** node.
7. Click the **Homepage Download (msec)** metric.

   A table lists all the collected data for this metric over the last 24 hours, by default. Figure 7–4 shows an example of the information displayed.

Figure 7–4 Grid Control Console - OracleAS Portal Metric Information



8. To change the duration to 31 days, select **Last 31 days** from the **View Data** drop-down list (top right hand corner).

> **Note:** For more information, see *Oracle Enterprise Manager Concepts* manual.

7.1.2 Comparing Metrics from Multiple Portal Targets

You can compare the details of an OracleAS Portal metric (the currently selected metric), with the details of the same metric on a different OracleAS Portal target.

For example, you can compare the Homepage Download (ms) metric on portal1 with the Homepage Download (ms) metric on portal2 and portal3. The comparisons are plotted on a Metric Value History chart.

To compare metrics:

1. Navigate to the OracleAS Portal metrics page, as shown in Figure 7–3.
2. Expand the metric node of interest and click the relevant metric link.
3. From the **View Data** drop-down list (top right hand corner), choose a suitable time period for analyzing this metric.
4. Click the **Compare Targets** link in the Related Links section.
5. Choose the OracleAS Portal targets that you want to compare, move them to the **Selected Targets** list box and then click OK.

   The comparisons are plotted on the Metric Value History chart.

7.1.3 Setting Up Notifications for OracleAS Portal Metrics

In the Grid Control Console, you can set up notification alerts to report that certain metrics exceed pre-set thresholds.

1. Check that the Oracle Enterprise Manager 10*g* administrator has setup at least one Notification Method for an Outgoing Mail Server, a Script (Operating System Command or PL/SQL) or an SNMP Trap:
    1. Click the **Setup** link (top right hand corner).
    2. Click **Notification Methods**.
2. Once a notification method exists, setup a Notification Rule:
    1. Click the **Preferences** link (top right hand corner).
    2. Click the **Notification Rules** link.

From this page you can create a notification rule and choose targets and conditions for which you want to receive notifications in Oracle Enterprise Manager 10*g*.

> **Note:** For more information, see *Oracle Enterprise Manager Concepts* manual.

7.1.4 Setting OracleAS Portal Metric Thresholds

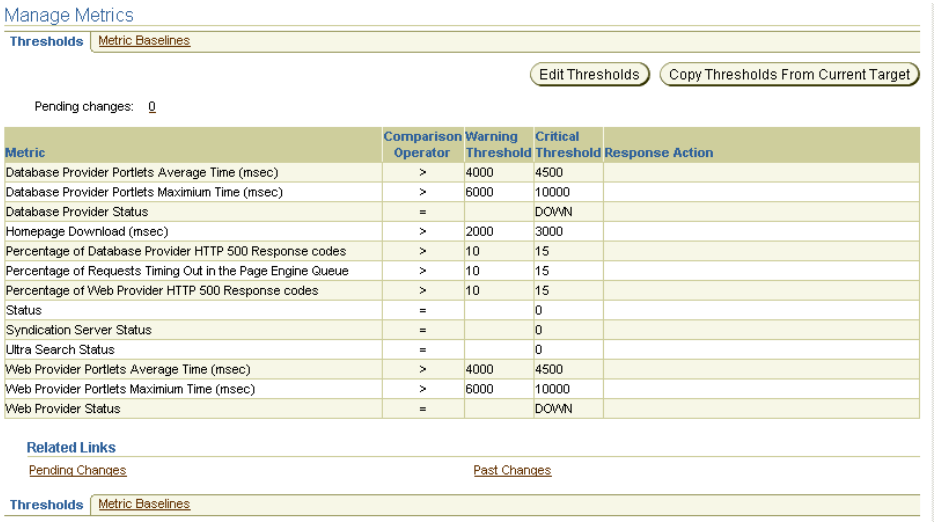
In the Grid Control Console, you can define and adjust the thresholds for OracleAS Portal metrics. Thresholds are boundary values against which monitored metric values are compared. You can specify a warning threshold so that when a monitored metric value crosses that threshold, a warning alert is generated. Alerts can notify you of impending problems which you can address in a timely manner.

Editing metric thresholds is useful because you can add or change the thresholds to fit the monitoring needs of your organization. When defining a threshold, choose a value that will not generate too many unnecessary alerts.

To edit OracleAS Portal related metrics, click the **Manage Metrics** link at the bottom of any OracleAS Portal target page, as shown in Figure 7–2. The metrics listed on the *Manage Metrics* page are either default metrics provided by Oracle, or metrics with thresholds you can change. For an example, see Figure 7–5.

> **Note:** For more information, see the *Oracle Enterprise Manager Concepts* manual.

Figure 7–5 Grid Control Console - OracleAS Portal Edit Metric Thresholds


Manage Metrics

Thresholds | Metric Baselines

Edit Thresholds | Copy Thresholds From Current Target

Pending changes: 0

Metric	Comparison Operator	Warning Threshold	Critical Threshold	Response Action
Database Provider Portlets Average Time (msec)	>	4000	4500	
Database Provider Portlets Maximium Time (msec)	>	6000	10000	
Database Provider Status	=		DOWN	
Homepage Download (msec)	>	2000	3000	
Percentage of Database Provider HTTP 500 Response codes	>	10	15	
Percentage of Requests Timing Out in the Page Engine Queue	>	10	15	
Percentage of Web Provider HTTP 500 Response codes	>	10	15	
Status	=		0	
Syndication Server Status	=		0	
Ultra Search Status	=		0	
Web Provider Portlets Average Time (msec)	>	4000	4500	
Web Provider Portlets Maximium Time (msec)	>	6000	10000	
Web Provider Status	=		DOWN	

Related Links

Pending Changes | Past Changes

Thresholds | Metric Baselines

7.1.5 Viewing Recent Alerts

A list of the most recent alerts are displayed on the OracleAS Portal target page, in the section called **Alerts** and **Host Alerts** (see Figure 7–2).

When an alert is generated to notify you of availability or performance problems, you can check the Grid Control Console for more information about the metric that triggered the alert. This includes information on the metric's historical values that might show trends over the past week or month.

7.1.6 Using Web Applications for Application Performance Monitoring

In the Grid Control Console, you can use the Web Applications feature to monitor the performance of OracleAS Portal sites. You can monitor the end-user response time, or the performance of representative transactions.

- **End-User Response Time Monitoring -** All URLs based on the application home page (specified in the Web Application properties) are monitored. URLs of particular importance can be identified in a URL Watchlist.
- **Representative Transaction Monitoring -** Recorded application activity (transactions) is played back at regular intervals through client robots (or beacons). The availability of the application is defined as the availability of a selected subset of representative transactions, replayed through selected beacons.

You model the topography of your OracleAS Portal application by adding all the relevant targets to a single Web Application group. For example, you could add the Database and OracleAS Single Sign-On targets used by your OracleAS Portal application to the same Web Application group.

Note: For more information, see the *Oracle Enterprise Manager Concepts* manual.

7.2 Using the Application Server Control Console

The Oracle Enterprise Manager 10*g* Application Server Control Console is included when you install Oracle Application Server. From OracleAS Portal's perspective,

consider this to be the administration console for the Oracle Application Server. In the Application Server Control Console you can:

- Enable and disable components
- Administer clusters
- Start and stop services
- View logs and ports
- Perform real-time monitoring
- Modify infrastructure services used by an Oracle Application Server middle tier

This section contains information about:

7.2.1 Accessing the Application Server Control Console

You can access the Application Server Control Console by navigating to the following URL: `http://<hostname.domain>:<port>`. For example, `http://myhost.mycompany.com:1810`. You can find the URL for your Application Server Control Console in `setupinfo.txt`. This text file is saved to the following location after you install the Oracle Application Server:

On UNIX: *`ORACLE_HOME`*`/install/setupinfo.txt`

On Windows: *`ORACLE_HOME`*`\install\setupinfo.txt`

Your start page for the Application Server Control Console is the Oracle Application Server **Farm** home page. Clicking an instance, takes you to the Oracle Application Server instance home page. This page contains a table of *System Components*. From this table you can display the home page for each component of the application server for monitoring and administrative purposes.

If OracleAS Portal is configured, **Portal:<portal schema name>** appears in this table. The default portal schema name is **portal**.

You can also navigate to Application Server Control Console directly from OracleAS Portal. Click the **Administer** tab on the **Portal Builder** page and then click **Portal Service Monitoring** (by default, this link is located in the **Services** portlet on the **Portal** subtab).

Note: If any Application Server Control Console details change, for example, the port or protocol, you must update the **Portal Service Monitoring** link otherwise it will not work. See Section 7.3.11.1, "Updating Oracle Enterprise Manager Link in OracleAS Portal" for instructions.

7.2.2 Using Application Server Control Console to Configure OracleAS Portal

If **Portal:portal** is not listed in the **System Components** table, it means that OracleAS Portal is not yet configured. The **Configure Component** button appears above the System Components table if you have installed, but not configured, some Oracle Application Server components.

> **Note:** Only components that have the check box selected can be started or stopped.

To configure OracleAS Portal perform these steps:

1. On the Oracle Application Server home page, click the **Configure Component** button.

   > **Note:** By default, an OracleAS Portal middle tier is made up of one portal instance. Both the DAD name and the OracleAS Metadata Repository schema name for this instance are **portal**. You cannot use the **Configure Component** button to configure additional OracleAS Portal instances for a given middle tier if **Portal:portal** is already listed in the **System Components** table.

2. Select **Portal** from the drop-down list on the **Select Component** page, and click **Continue**.
3. In the **Login** page, enter the administration (`ias_admin`) password for the Oracle Application Server instance in the **Administration Password** field.
4. Click **Finish**.

   It may take some time to complete this process (10-20 minutes), depending upon the speed and configuration of your hardware.
5. When the configuration is complete, click **OK**.

   The Oracle Application Server home page is displayed.
6. Verify that *OC4J_Portal* and *Portal:portal* appear in the **System Components** table.
7. Restart Oracle HTTP Server and start OC4J_Portal.
   a. In the **System Components** table, select **HTTP_Server**, and click the **Restart** button.
   b. Select **OC4J_Portal**, and then click the **Start** button.

   The home OC4J instance will be *Down* after configuring OracleAS Portal through Oracle Enterprise Manager. If you wish to start this service, click the **home** component in the **System Components** table and then click the **Start** button.
8. Verify that the status of *OC4J_Portal* and *Portal:portal* are both *Up*:
   - Click **OC4J_Portal** and verify that the OC4J_Portal page is displayed.
   - Click **Portal:portal** and verify that the Portal page is displayed.

   Initially, the **Portal:portal** status may appear *Down*. This is normal. The status should be updated approximately five minutes after configuration.
9. If this is the *first* instance of OracleAS Portal to use the OracleAS Metadata Repository, run the following command in the middle-tier Oracle home (make sure the `ORACLE_HOME` environment variable is set before you run this command):

   On Unix: `ORACLE_HOME/portal/conf/ptlconfig -dad portal [-pw PORTAL_schema_password]`

   On Windows: `ORACLE_HOME\portal\conf\ptlconfig -dad portal [-pw PORTAL_schema_password]`

This script writes OracleAS Portal configuration entries into the OracleAS Metadata Repository. Do *not* run this script if there are other OracleAS Portal instances using the OracleAS Metadata Repository as this script will overwrite any existing OracleAS Portal configuration entries in the OracleAS Metadata Repository.

Note: The PORTAL schema password is stored in the Oracle Internet Directory and the entry may be viewed by an administrator using the `oidadmin` utility with the following path under Entry Management:

```
OrclResourceName=PORTAL,orclReferenceName=iasdb.myho
st.au.oracle.com,cn=IAS Infrastructure
Databases,cn=IAS,cn=Products,cn=OracleContext
```

---

10. Verify that you can access OracleAS Portal at the following URL:

    `http://<hostname.domain>:<port>/portal/pls/portal`

    In the URL, `hostname.domain` is the OracleAS Portal host, and `port` is the OracleAS Web Cache HTTP port number for the OracleAS Portal instance. For example, `http://myhost.mycompany.com:7777/portal/pls/portal`.

    You can log in to OracleAS Portal as the user `portal`.

    - If this is the first OracleAS Portal instance to use the OracleAS Metadata Repository, the password is the *original* `ias_admin` password you supplied for this middle tier during installation. The *original* `ias_admin` password is required, even if you changed the `ias_admin` password, after installation.
    - If this is *not* the first OracleAS Portal instance to use the OracleAS Metadata Repository, the password is either the:
      - original `ias_admin` password for the first middle tier associated with the OracleAS Metadata Repository, or
      - current `portal` password, if the administrator changed the `portal` user password after the first OracleAS Portal instance was installed.

For more information, see the *Oracle Application Server Administrator's Guide*.

Note: When OracleAS Portal is configured using Oracle Enterprise Manager, the Oracle Ultra Search instance is not configured automatically and therefore the **Ultra Search Administration** link in OracleAS Portal will not work. See Section 8.2.4.1, "Accessing the Oracle Ultra Search Administration Tool" for details.

7.3 Using Application Server Control Console to Monitor and Administer OracleAS Portal

To monitor and administer OracleAS Portal, click **Portal:<portal schema name>** in the System Components list on the Oracle Application Server instance home page. The default portal schema name is **portal**. Note that **OC4J_Portal** is the container for portal servlets, and not the actual portal servlet to monitor.

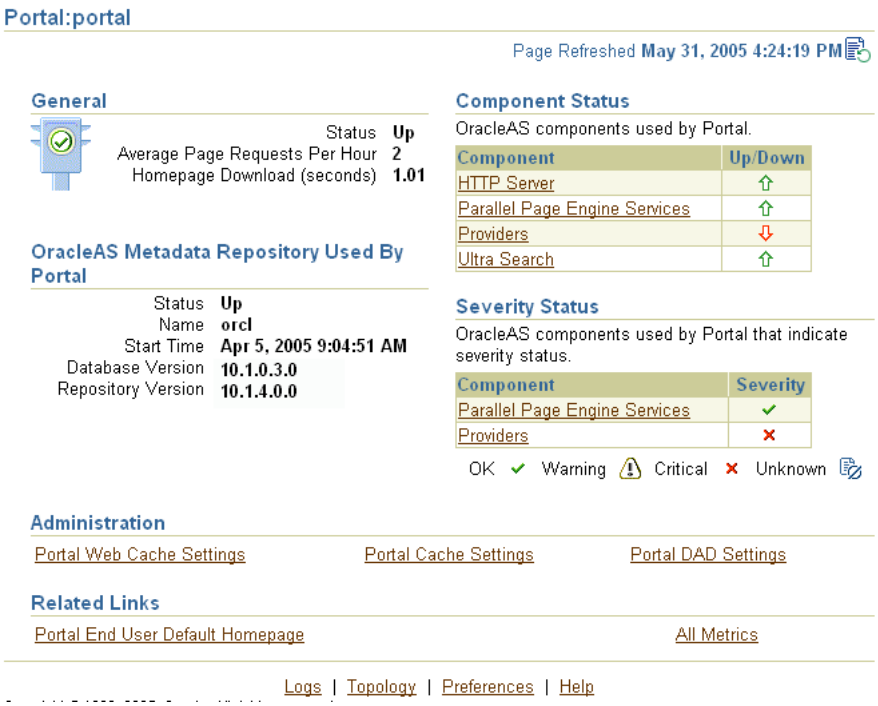
Figure 7–6 shows the main page for monitoring OracleAS Portal.

You can access the Oracle Application Server instance home page for your portal directly from OracleAS Portal. Click the **Administer** tab on the **Portal Builder** page

and then click **Portal Service Monitoring** (by default, this link is located in the **Services** portlet on the **Portal** subtab).

Note: If any Oracle Enterprise Manager 10*g* Application Server Control Console details change, for example, the port or protocol, you must update the **Portal Service Monitoring** link in OracleAS Portal otherwise it will not work. See Section 7.3.11.1, "Updating Oracle Enterprise Manager Link in OracleAS Portal" for more information.

Figure 7–6 Application Server Control Console - Main OracleAS Portal Monitoring Page



The main OracleAS Portal monitoring page, shown in Figure 7–6, displays various information and links which are described in the following section:

- General Status Information
- OracleAS Metadata Repository Information
- Component Status Table
- Severity Status Table
- Portal Web Cache Settings Link
- Portal Cache Settings Link
- Portal DAD Settings Link
- Related Links
- Logs Link
- Topology Link

For performance reasons, less critical data, that is, non-response metrics, is collected by the Application Server Control Console metric cache and it can become slightly out of date. To display the most up to date metric data, click the **Refresh Data** icon.

In addition to the portal metrics available in the Application Server Control Console, the Dynamic Monitoring Service (DMS) collects portal-related metric data from the Parallel Page Engine, and the Providers. DMS starts collecting metric information when the middle tier starts up. DMS metrics are cleared whenever the middle tier is restarted, and a new set of metric data is collected when the middle tier starts up again. To view these metrics, use the DMS monitoring tool, `dmstool`, which is documented in the *Oracle Application Server Performance Guide*.

> **See Also:** The appendix titled "Performance Metrics" in *Oracle Application Server Performance Guide*.

7.3.1 General Status Information

On the main page for OracleAS Portal, in Application Server Control Console, there is a *General* section that displays important status and performance-related information. Use this section to establish the status of an OracleAS Portal instance, either *Up* or *Down*.

You can also see the average number of page requests for each hour, and the current home page download speed.

7.3.2 OracleAS Metadata Repository Information

On the main page for OracleAS Portal, in Application Server Control Console, there is a section called *OracleAS Metadata Repository Used By OracleAS Portal*. Use this section to view metrics relating to the OracleAS Metadata Repository. This is the repository that contains the OracleAS Portal schema.

You can see if the database that contains the OracleAS Metadata Repository is up and running, and the date and time the database was started. You can also find out the name and version of the database, and the version of the OracleAS Metadata Repository.

7.3.3 Portal Web Cache Settings Link

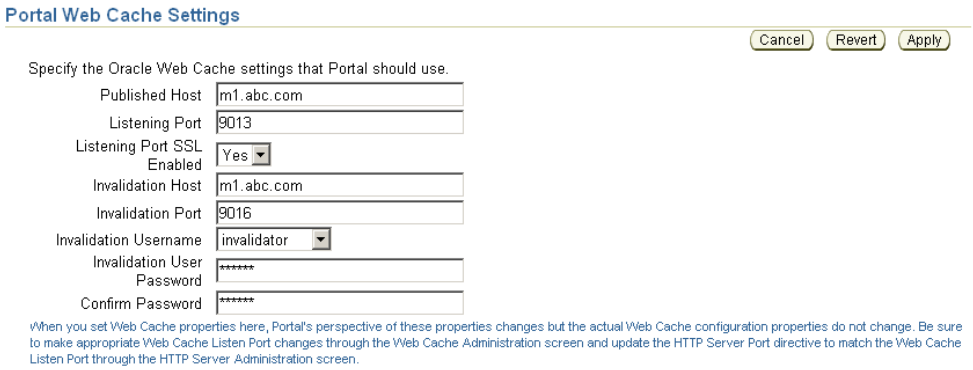
In Application Server Control Console, you can specify the OracleAS Web Cache settings that OracleAS Portal should use. From the main OracleAS Portal monitoring page, click the **Portal Web Cache Settings** link, under **Administration**, to display the *Portal Web Cache Settings* page shown in Figure 7–7.

When you set OracleAS Web Cache properties on this page, the Portal Dependency Settings file (`iasconfig.xml`) located on this middle tier is updated automatically, and the OracleAS Portal schema is also updated. See Appendix A, "Using the Portal Dependency Settings Tool and File" for more details.

Notes:

- When you change OracleAS Web Cache properties in the **Portal Web Cache Settings** page, the properties are saved to `iasconfig.xml`, but not to `webcache.xml`. Navigate to the *Web Cache Administration* page, in Application Server Control Console, to make the appropriate changes to `webcache.xml`. Refer to the *Oracle Application Server Web Cache Administrator's Guide* for more information about OracleAS Web Cache.
- Changing OracleAS Web Cache settings (for example, Listening Port) can change the OracleAS Portal URL. If you do this, your mobile settings need to be updated. See Section C.8, "Using the cfgiasw Script to Configure Mobile Settings" for more information.
- Changing OracleAS Web Cache settings can impact Web Providers (such as OmniPortlet and Web Clipping) if the OracleAS Web Cache and the Web Provider are running on the same middle tier. In this case, you must make corresponding cache configuration changes for the Web Providers. See *"Defining the OracleAS Web Cache Invalidation Port"* in the *Oracle Application Server Portal Developer's Guide*.

Figure 7–7 Application Server Control Console - Portal Web Cache Settings



In the **Portal Web Cache Settings** page, you can modify the settings detailed in Table 7–1:

Table 7–1 Portal Web Cache Settings

Setting	Description
Published Host	The published host name. This is the published server host name used in a browser to connect to OracleAS Portal. For example, `www.company.com`.
Listening Port	The port on which OracleAS Web Cache listens. For example, `7778`.
Listening Port SSL Enabled	Indicates whether OracleAS Web Cache is SSL enabled. Valid values are *Yes* and *No*.

Table 7–1 (Cont.) Portal Web Cache Settings

Setting	Description
Invalidation Host	The name of the OracleAS Web Cache host to which invalidation messages are sent. Use this only if it is different from the published host name. For example, `www.internal.company.com`. The published host name is used if you leave this field empty. Typically, the host name and port number used to connect to OracleAS Portal are the OracleAS Web Cache host name and port number. This is because, in a simple configuration, browser requests go directly to OracleAS Web Cache. However, if you have a reverse proxy server front-ending OracleAS Web Cache or a load balancing router (LBR), you can still send invalidation messages directly to the OracleAS Web Cache host instead of the reverse proxy server, or LBR. In this case the published host name should be that of the reverse proxy server, or LBR and the invalidation host name should be that of OracleAS Web Cache.
Invalidation Port	The number of the OracleAS Web Cache invalidation port, to which invalidation messages are sent. For example, `4001`.
Invalidation Username	The user name used for sending the invalidation messages. Either `invalidator` or `administrator`.
Invalidation User Password	The invalidation password. The default is the same as the `ias_admin` password chosen at the time of the OracleAS Portal middle-tier installation.
Confirm Password	Repeat the invalidation password.

Example 7–1 shows how to configure OracleAS Portal to use OracleAS Web Cache on a different host, using the **Portal Web Cache Settings** page.

Example 7–1 Configuring OracleAS Portal to Use OracleAS Web Cache on a Different Host

To configure OracleAS Portal to use OracleAS Web Cache on a different host from the one on which the OracleAS Portal middle tier is installed:

1. Access the Application Server Control Console on the middle tier where OracleAS Portal is installed.
2. Select the portal instance you want to configure, typically, **Portal:portal**.
3. Select **Portal Web Cache Settings**.
4. Update the **Published Host** property with the new host name, along with any other property changes you want to make.
5. Click **Apply**.

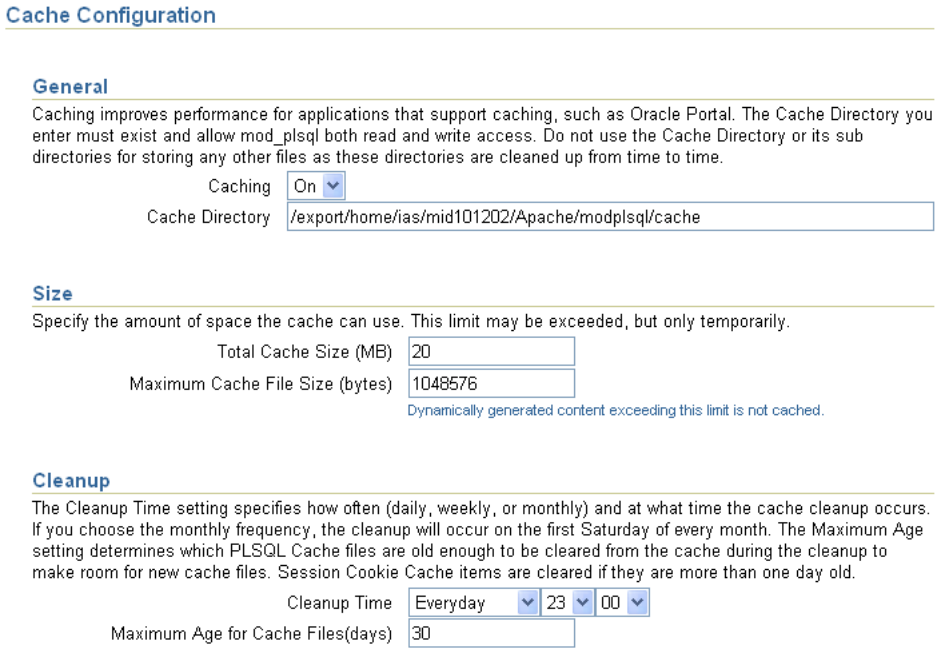
7.3.4 Portal Cache Settings Link

In Application Server Control Console, you can specify *content cache* and *session cache* settings for an OracleAS Portal instance. From the main OracleAS Portal monitoring page, click the **Portal Cache Settings** link, under **Administration**, to display the **Cache Configuration** page shown in Figure 7–8.

See Section 1.3.2, "Understanding Portal Cache" for more information.

Note: If you change any values on the **Cache Configuration** page, you must restart Oracle HTTP Server and OC4J_Portal. To do this, navigate to the Oracle Application Server home page, select **HTTP_Server** and **OC4J_Portal** in the **System Components** table, and then click the **Restart** button.

Figure 7–8 Application Server Control Console - Portal Cache Settings



In the **Cache Configuration** page, you can modify the settings detailed in Table 7–2:

Table 7–2 Portal Cache Settings

Setting	Description
Caching	Enables (*On*) or disables (*Off*) portal content and session caching.
Cache Directory	The directory where cached content is stored. **Note:** Ensure that this directory exists and that it is accessible (read/write access required).
Total Cache Size (MB)	The total amount of disk space (in megabytes) that the portal cache may use. The maximum value allowed is 4 GB. **Note:** This setting is not a hard limit. The cache may exceed this limit temporarily.
Maximum Cache File Size (bytes)	Enter the maximum size (in bytes) for all cached files. The maximum value allowed is 4 GB. Any dynamically generated content that exceeds this limit is not cached.

Table 7–2 (Cont.) Portal Cache Settings

Setting	Description

Cleanup Time	The time at which to start the cleanup of the cache storage. Use the format: `[Sunday-Saturday, Everyday, Everymonth][hh:mm]` to define the exact day and time in which cleanup should occur.
	The frequency can be set as daily, weekly, and monthly. The default is `Everyday 23:00`.An infrequent cleanup improves performance but total cache size may be exceeded. A frequent cleanup decreases performance, but total cache size is not exceeded.
	■ To define daily frequency, use the keyword `Everyday`. A cleanup will start everyday at the time defined. For example:
	`Everyday 2:00`
	Cleans up the cache everyday at 2 am (local time) in the morning.
	■ To define weekly frequency, enter the day of the week [Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday]. For example:
	`Wednesday 15:30`
	Cleans up the cache every Wednesday at 3:30 pm (local time) in the afternoon.
	■ To define monthly frequency, use the keyword `Everymonth`. The cleanup starts on the first Saturday of the month at the time defined. For example:
	`Everymonth 23:00`
	Cleans up the cache on the first Saturday of every month at 11:00 pm (local time) at night.

Maximum Age for Cache File (days)	The maximum age for cached files. This setting ensures the cache system does not contain any old content. Old cache files are removed to make space for new cache files. The default is `30 days`. **Note:** This setting only affects items in the portal content cache. Session cookie cache items are cleaned if they are in the cache for more than 1 day.

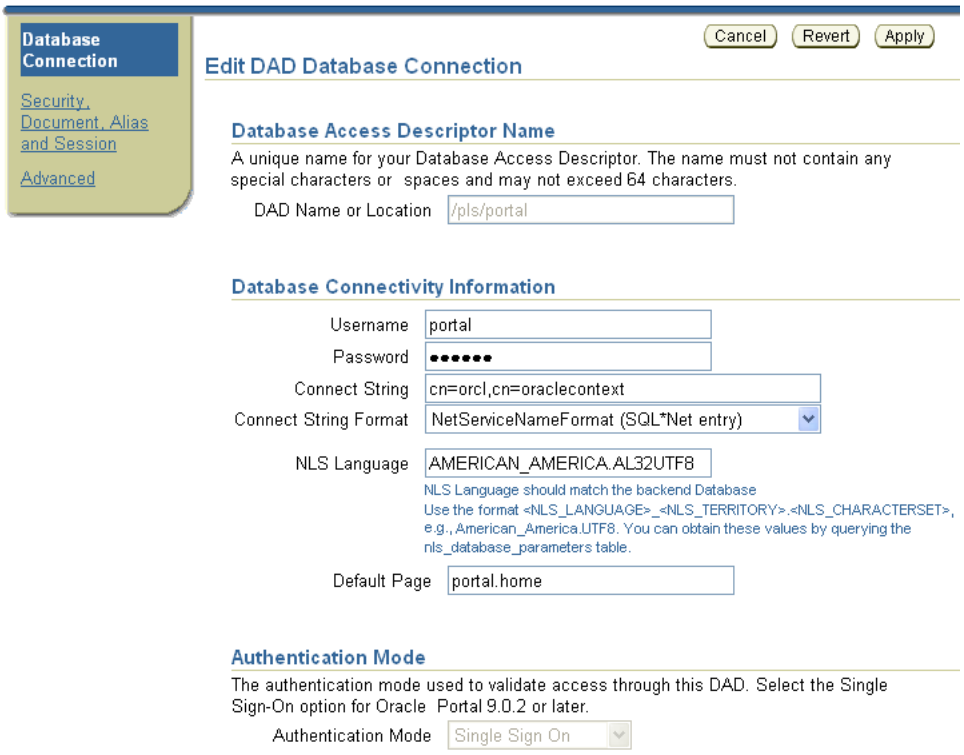
7.3.5 Portal DAD Settings Link

In Application Server Control Console, you can edit the DAD (Database Access Descriptor) for an OracleAS Portal instance. Click the **Portal DAD Settings** link, under **Administration**, to display the **Edit DAD Database Connection** page shown in Figure 7–9.

> **Note:** If you make any changes on the **Edit DAD** pages, you must restart Oracle HTTP Server and OC4J_Portal. To do this, navigate to the Oracle Application Server home page, select **HTTP_Server** and **OC4J_Portal** in the **System Components** table, and then click the **Restart** button.

See Section 4.5.3, "Configuring a Portal DAD" for more information.

Figure 7–9 Application Server Control Console - Portal DAD Settings



In the **Edit DAD Database Connection** page, you can modify the DAD settings detailed in Table 7–3:

Table 7–3 DAD Settings

Setting	Description
Username	The Oracle Database account username.
Password	The Oracle Database account password. The password is typically set at installation but you can change it by typing a new password in this field. Use this field to set the password for a *nondefault* OracleAS Portal instance. For the *default* OracleAS Portal instance, we recommend that you set the password through the *Infrastructure* page in Application Server Control Console. This way, password changes are propagated to the database, Oracle Internet Directory, and the DAD. See Section 5.12, "Changing the OracleAS Portal Schema Password" for more information.
Connect String	The connection string (if the database is remote) and then use the **Connection String Format** as described subsequently to specify the format of the connect string you have entered here.

Table 7–3 (Cont.) DAD Settings

Setting	Description

Connect String Format	The format used for the **Connect String** property. The options are:
	■ **ServiceNameFormat (host:port:service_name)** - Use this format when the connect string is in the format `host:port:service_name`. For example:
	`myhost.oracle.com:1521:mydb.oracle.com`
	■ **SIDFormat (host:port:sid)** - Use this format when the connect string is in the format `host:port:sid`. For example:
	`myhost.oracle.com:1521:mydb`
	The SIDFormat is provided for backward compatibility only and it will be deprecated in a future release.
	■ **TNSFormat (TNS alias or the whole TNS entry)** - Use this format when the connect string is resolved through tnsnames.ora or when the complete tnsnames.ora entry is specified in the Portal Services configuration file. For example:
	`myhostdb.oracle.com` or
	`DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=myhost.oracle.com)(Port=1521))(CONNECT_DATA=(SID=mydb))"`
	■ **NetServiceNameFormat (SQL*Net entry)** - Use this format when the connect string is resolved by SQL*Net, For example, `cn=oracle,cn=mydb` for name resolution through LDAP, or `mydb.oracle.com` for name resolution through `tnsnames.ora`. Refer to the SQL*Net documentation for a more detailed description of *Net Service Names*.
	If the **Connection String Format** is not specified, the connect string format is assumed to be either **SIDFormat (host:port:sid)** or, resolvable as **NetServiceNameFormat**. The differentiation between the two is made by the presence of colon characters (:) in the connect string.
	For database installations like Real Application Clusters (RAC), the recommended connect string format is **NetServiceNameFormat** so that the lookup is through LDAP. This allows database nodes to be for added/removed without having to reconfigure each Oracle Application Server middle tier separately to recognize added/removed nodes.

Table 7–3 (Cont.) DAD Settings

Setting	Description
NLS Language	The Globalization Support language of the OracleAS Portal database that is represented by this DAD. This setting overrides the NLS_LANG environment variable for a database session and defines some important Globalization Support properties of the response, including the response character set. For OracleAS Portal, this setting should match the NLS_LANG of the back-end database. For example, if you set this parameter is to `JAPANESE_JAPAN.JA16SJIS`, content is transferred from the database in the `JA16SJIS` character set. **Tip:** To obtain the settings of this parameter, query the *nls_database_parameters* table as follows: `select value, parameter from nls_database_parameters where parameter in ('NLS_LANGUAGE', 'NLS_TERRITORY','NLS_CHARACTERSET');` Refer to the *Oracle HTTP Server Administrator's Guide*, for more details on the parameter *PlsqlNLSLanguage*.

Default Page	The PL/SQL procedure that is invoked when one is not specified as part of the URL. For example, if you specify a default home page of `myapp.home` and an end user enters this URL in a browser:
	
	`http://myapp.myserver.com:<port>/portal/pls/myapp/`
	
	The URL is automatically updated to:
	
	`http://myapp.myserver.com:<port>/portal/pls/myapp/myapp.home`

Security, Document, Alias and Session	Links to additional DAD settings: Request Validation Function, Exclusion List, Document Access Information (Document Table, Document Path, Document Access Procedure, Long Raw), Path Alias, Session Cookie Name, Session State Management. For more information, click the **Help** link on the *Edit DAD Security, Document, Alias and Session* page.
Advanced	Links to advanced DAD settings: Always Describe Procedures, Before Procedure, After Procedure, Fetch Buffer Size (rows). For more information, click the **Help** link on the *Edit DAD Advanced* page.

7.3.6 Component Status Table

The Component Status table, on the main page for OracleAS Portal in Application Server Control Console, lists the Oracle Application Server components used by OracleAS Portal and indicates their current status. You can drill down and find more information about individual Oracle Application Server components, by clicking on a link. The listed components are:

- HTTP Server
- Parallel Page Engine Services
- Providers
- Ultra Search

7.3.6.1 HTTP Server

Clicking the **HTTP Server** link, in the Component Status table, takes you to the home page for the Oracle HTTP Server. This is the starting point for managing a single instance of Oracle HTTP Server. For example, you can restart the Oracle HTTP Server from here.

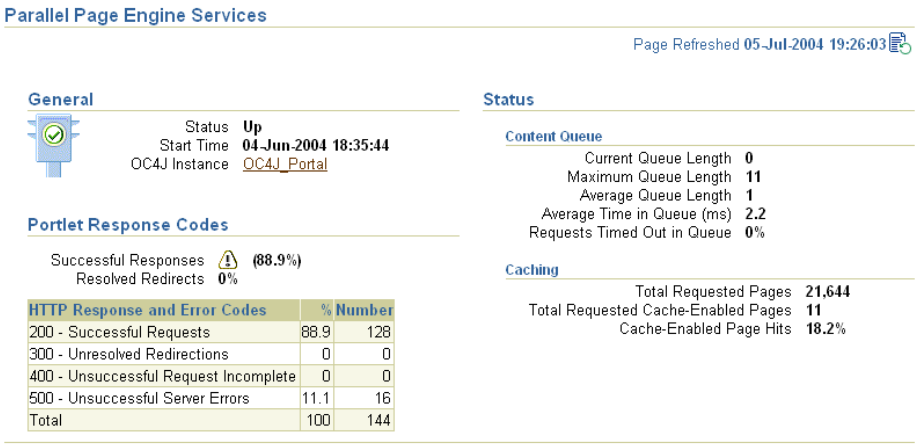
> **See Also:** For more information, refer the *Oracle HTTP Server Administrator's Guide*.

7.3.6.2 Parallel Page Engine Services

Clicking the **Parallel Page Engine Services** link, in the Component Status table, takes you to the Parallel Page Engine Services page shown in Figure 7–10. Detailed Parallel Page Engine (PPE) metrics are displayed on this page.

> **See Also:** Refer to Appendix D, "Configuring the Parallel Page Engine" for more information about the PPE.

Figure 7–10 Application Server Control Console - Parallel Page Engine Services Monitoring Page



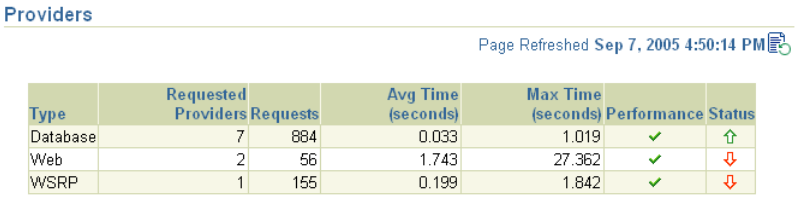
7.3.6.3 Providers

In the Component Status table, the status of Providers is shown to be *Up* when none of the portlets' last response codes indicated a failure, and the slowest average portlet performance is quicker than the portlet performance warning threshold.

Clicking the **Providers** link, in the takes you to the Providers monitoring page. From here you can get an overview of the performance, status, and HTTP response codes (portlets only) of providers and portlets that are requested by the Parallel Page Engine (PPE) in the Application Server Control Console.

The performance of Web Providers, Database Providers, and WSRP Providers are summarized at the top of the Providers page, as shown in Figure 7–11.

Figure 7–11 Application Server Control Console - Provider Summary



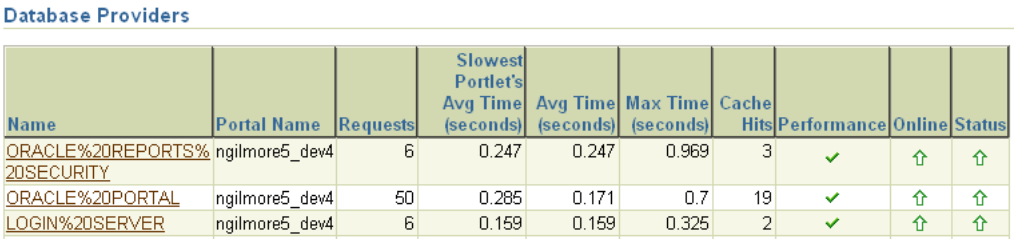
Providers

Page Refreshed Sep 7, 2005 4:50:14 PM

Type	Requested Providers	Requests	Avg Time (seconds)	Max Time (seconds)	Performance	Status
Database	7	884	0.033	1.019	✔	⇧
Web	2	56	1.743	27.362	✔	⇩
WSRP	1	155	0.199	1.842	✔	⇩

You can click a provider to get details about individual portlets that have been accessed through that provider.

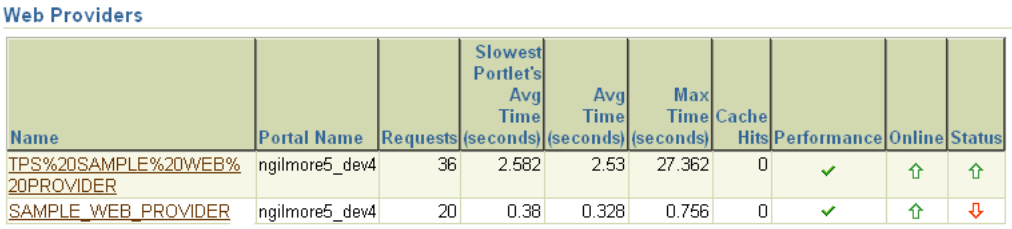
Figure 7–12 Application Server Control Console - Database Providers



Database Providers

Name	Portal Name	Requests	Slowest Portlet's Avg Time (seconds)	Avg Time (seconds)	Max Time (seconds)	Cache Hits	Performance	Online	Status
ORACLE%20REPORTS%20SECURITY	ngilmore5_dev4	6	0.247	0.247	0.969	3	✔	⇧	⇧
ORACLE%20PORTAL	ngilmore5_dev4	50	0.285	0.171	0.7	19	✔	⇧	⇧
LOGIN%20SERVER	ngilmore5_dev4	6	0.159	0.159	0.325	2	✔	⇧	⇧

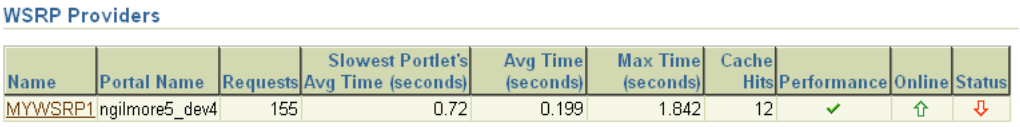
Figure 7–13 Application Server Control Console - Web Providers



Web Providers

Name	Portal Name	Requests	Slowest Portlet's Avg Time (seconds)	Avg Time (seconds)	Max Time (seconds)	Cache Hits	Performance	Online	Status
TPS%20SAMPLE%20WEB%20PROVIDER	ngilmore5_dev4	36	2.582	2.53	27.362	0	✔	⇧	⇧
SAMPLE_WEB_PROVIDER	ngilmore5_dev4	20	0.38	0.328	0.756	0	✔	⇧	⇩

Figure 7–14 Application Server Control Console - WSRP Providers



WSRP Providers

Name	Portal Name	Requests	Slowest Portlet's Avg Time (seconds)	Avg Time (seconds)	Max Time (seconds)	Cache Hits	Performance	Online	Status
MYWSRP1	ngilmore5_dev4	155	0.72	0.199	1.842	12	✔	⇧	⇩

Metrics you can monitor include:

- Requests - The number of requests serviced by this provider.
- Avg Time (seconds) - The average response time to request a portlet.
- Max Time (seconds) - The maximum response time to request a portlet.
- Slowest Portlets Avg Time (seconds) - The average performance of a provider's slowest portlet (in seconds).
- Cache Hits - The number of times the cache has been accessed.
- Performance - Indicates whether the providers are performing as expected.
- Online - Indicates if a provider is currently online.
- Status - Indicates whether a specific provider is Up or Down.

The status of a portlet is considered to be *Up* if no portlets' last response code indicated a failure, and the slowest average portlet performance is quicker than the portlet performance warning threshold set for the target. The status appears *Down* if the last response code for at least one portlet indicates a failure.

7.3.6.4 Ultra Search

Clicking the **Ultra Search** link, in the Component Status table, takes you to the Oracle Ultra Search administration pages. From here you can configure Oracle Ultra Search. See Chapter 8, "Configuring the Search Features in OracleAS Portal" for more information.

7.3.7 Severity Status Table

The Severity Status table, on the main page for OracleAS Portal in Application Server Control Console, lists the Oracle Application Server components used by OracleAS Portal that indicate severity status. Table 7–4 describes the severity status levels that are reported.

Note: Severity level thresholds are set in `targets.xml`. See section "Setting Severity Thresholds Manually in targets.xml" for more information.

Table 7–4 Severity Level Status Descriptions

Item	Description
OK	The component is running normally.
Warning	There is some problem with the component.
Critical	The component is having critical problems.
Unknown	There is not enough information to establish the status as the component is down.

Setting Severity Thresholds Manually in targets.xml

In the Grid Control Console, you can define and adjust the thresholds for OracleAS Portal metrics using the **Manage Metrics** link (see Section 7.1.4, "Setting OracleAS Portal Metric Thresholds" for information). This feature is not available in the Application Server Control Console but you can edit the severity level thresholds manually, in the Oracle Enterprise Manager configuration file `targets.xml`, if necessary.

For the Application Server Control Console, this file is located at *`ORACLE_HOME`*`/sysman/emd/targets.xml`, where *`ORACLE_HOME`* is the Oracle Application Server Home. You can edit the following threshold values:

- **PPESuccessfulResponsesCriticalThreshold** - A threshold value used by OracleAS Portal to determine whether the status of the Parallel Page Engine Services is reported to be critical, that is, when the percentage of successful responses drops below this value. The default is 80%.
- **PPESuccessfulResponsesWarningThreshold** - Parallel Page Engine Services are considered to be in a warning state if the percentage of successful responses is higher than PPESuccessfulResponsesCriticalThreshold and lower than this value. The default is 90%.

- **portletResponseCriticalThreshold** - A threshold value used by OracleAS Portal to determine whether the status of a Provider's Portlet Timing is reported to be critical, that is, when the average portlet response time (in milliseconds) is greater than this value (in milliseconds). The default is 4500 milliseconds.
- **portletResponseWarningThreshold** - Portlet response times are considered to be in a warning state if the average response time for Provider portlets is less than PPESuccessfulResponsesCriticalThreshold and greater than this value (in milliseconds). The default is 4000 milliseconds.

If you do edit metric thresholds manually in `targets.xml`, you must reload the configuration file in the appropriate Application Server Control Console. Follow these steps to update `targets.xml` in the Oracle Application Server 10*g* home (*`ORACLE_HOME`*) and load the new target information in the Application Server Control Console:

1. Create a backup copy of *`ORACLE_HOME`*`/sysman/emd/targets.xml`. For example, *`ORACLE_HOME`*`/sysman/emd/targets.xml.30thJune2005`.
2. Open `ORACLE_HOME/sysman/emd/targets.xml` in your favorite Text Editor.
3. Edit the XML segment describing your OracleAS Portal target thresholds.

```
<Target TYPE="oracle_portal" NAME="Name" DISPLAY_NAME="Display_Name"
VERSION="1.0">
   ...
   <Property NAME='PPESuccessfulResponsesCriticalThreshold' VALUE='80' />
   <Property NAME='PPESuccessfulResponsesWarningThreshold' VALUE='90' />
   <Property NAME='portletResponseCriticalThreshold' VALUE='4500' />
   <Property NAME='portletResponseWarningThreshold' VALUE='4000' />
    ...
  </Target>
```

4. Reload the targets in the Application Server Control Console:

   On Solaris/Linux:

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

   On Windows:

   ```
   ORACLE_HOME\bin\emctl reload
   ```

7.3.8 Related Links

The Related Links section, on the main page for OracleAS Portal in Application Server Control Console, contains several links:

- **Portal End User Default Homepage** - This link takes you to the home page for the OracleAS Portal instance.
- **All Metrics** - This link displays a single comprehensive list of all the metrics available for this OracleAS Portal instance. It is not possible to view historical metric data from the Application Server Control Console but you can *start* collecting metric data and display the results graphically in real-time.

  To analyze historical trends, use the Grid Control Console. See Section 7.1.1, "Monitoring Historical Trends" for more information.

7.3.9 Logs Link

A **Logs** link is displayed at the top and bottom of every Oracle Application Server component home page. Click the **Logs** link on any OracleAS Portal instance home page to view detailed diagnostic information for that OracleAS Portal instance.

See Appendix K, "Troubleshooting OracleAS Portal" for more information.

See Also: *Oracle Application Server Administrator's Guide*

7.3.10 Topology Link

A **Topology** link is displayed at the top and bottom of every Oracle Application Server component home page. Click the **Topology** link to display a graphical representation of your Oracle Application Server environment, including the Oracle Application Server processes that are running OracleAS Portal instances, such as, Web Cache, OC4J_Portal and HTTP Server.

See Also: *Oracle Application Server Administrator's Guide*

7.3.11 Additional Configuration Requirements

This section contains the following sections:

- Updating Oracle Enterprise Manager Link in OracleAS Portal
- Monitoring OracleAS Portal in an SSL Environment

7.3.11.1 Updating Oracle Enterprise Manager Link in OracleAS Portal

In OracleAS Portal there is a link to the Application Server Control Console that is monitoring and managing the portal. To access the **Portal Service Monitoring** link, click the **Administer** tab in OracleAS Portal (Builder page) and then locate the **Services** portlet.

If any details relating to the Application Server Control Console change, for example, the port or protocol, the link in OracleAS Portal must be updated otherwise it will not work.

To do this, follow these steps:

1. Edit the file `iasconfig.xml` on the OracleAS Portal middle tier.

   This is usually located in *ORACLE_HOME*`/portal/conf`. See Appendix A, "Using the Portal Dependency Settings Tool and File" for more information.

2. Update the **EMComponent** element for your OracleAS Portal instance, as required.

3. Run the following script to update the Oracle Application Server Metadata Repository with the new settings:

   ```
   ORACLE_HOME/portal/conf/ptlconfig -dad <dad> -em
   ```

4. Clear the content of the portal cache directory `ORACLE_HOME/Apache/modplsql/cache/plsql/sys`:

   a. Navigate to the portal cache directory (default path is `ORACLE_HOME/Apache/modplsql/cache`).

   b. Perform a recursive delete of all the files under `ORACLE_HOME/Apache/modplsql/cache/plsql/sys`. For example, on UNIX platforms, issue the following command:

```
rm -rf ORACLE_HOME/Apache/modplsql/cache/plsql/sys
```

5. In OracleAS Portal, clear the content of OracleAS Web Cache to update the **Portal Service Monitoring** link.
   a. In the **Services** portlet, click **Global Settings**.

      By default, the **Services** portlet is located on the **Administer** tab of the **Portal Builder** page.

   b. Click the **Cache** tab and then select **Clear The Entire Web Cache**.

   c. Click **OK**.
6. In the **Services** portlet, click the updated **Portal Service Monitoring** link to access the Application Server Control Console.

7.3.11.2 Monitoring OracleAS Portal in an SSL Environment

The configuration file `targets.xml` defines attributes for all the targets managed by Oracle Enterprise Manager, including your OracleAS Portal targets. For the most part, the content of `targets.xml` is maintained automatically, so there is no need to edit this file manually. However, when the Oracle HTTP Server in your environment is SSL enabled, you must maintain Oracle HTTP Server port changes manually (in `targets.xml`) for all OracleAS Portal targets running on that server. If the HTTP Port information in `targets.xml` does not match the actual listening port of the Oracle HTTP Server, the targets appear to be *down* in the Oracle Enterprise Manager 10*g* Application Server Control Console.

So, whenever Oracle HTTP Server listening port details are changed, either in the configuration file `httpd.conf`, or using the Oracle HTTP Server's administration pages in Oracle Enterprise Manager, update the corresponding property (HTTPPort) in `targets.xml` manually as described in the following sections.

If OracleAS Web Cache is SSL-enabled, the `PortalListeningHostPort` property must be changed too.

Follow the steps appropriate to your SSL environment:

- SSL-enabled Oracle HTTP Server and non-SSL OracleAS Web Cache
- Non-SSL Oracle HTTP Server and SSL-enabled OracleAS Web Cache
- SSL-enabled Oracle HTTP Server and SSL-enabled OracleAS Web Cache

> **Note:** Manual configuration is not required in non-SSL environments.

SSL-enabled Oracle HTTP Server and non-SSL OracleAS Web Cache

1. Make a backup copy of `targets.xml` located in the *`MID_TIER_ORACLE_HOME`*`/sysman/emd` directory.
2. Open *`MID_TIER_ORACLE_HOME`*`/sysman/emd/targets.xml` in a text editor.
3. Search for OracleAS Portal targets, that is, `TYPE="oracle_portal"`.
4. Change the `HTTPProtocol` property value from `http` to `https`. For example:

   ```
   <Property NAME="HTTPProtocol" VALUE="https"/>
   ```
5. Change the `HTTPPort` property value so that it matches the SSL-enabled Oracle HTTP Server listening port value. For example:

```
<Property NAME="HTTPPort" VALUE="7782"/>
```

To verify the current listening port of the Oracle HTTP Server, look in `httpd.conf` and determine the value of the `Listen` property. Alternatively, click the Site link on the OracleAS Web Cache Administration page and determine the port number displayed in the Origin Servers field.

6. Save the changes to `targets.xml`.
7. Configure Enterprise Manager to monitor the SSL-Enabled targets. To do this, follow the instructions provided in the *Oracle Application Server Administrator's Guide*.

Non-SSL Oracle HTTP Server and SSL-enabled OracleAS Web Cache

1. Make a backup copy of `targets.xml` located in the *`MID_TIER_ORACLE_HOME`*`/sysman/emd` directory.
2. Open *`MID_TIER_ORACLE_HOME`*`/sysman/emd/targets.xml` in a text editor.
3. Edit the `PortalListeningHostPort` property, which is the OracleAS Web Cache route for some of the status metrics. Replace `http` with `https`. For example:

   ```
   <Property NAME="PortalListeningHostPort" VALUE=https://<host>:<Web Cache port>/>
   ```

4. Save the changes to `targets.xml`.
5. Configure Enterprise Manager to monitor the SSL-Enabled targets. To do this, follow the instructions provided in the *Oracle Application Server Administrator's Guide*.

SSL-enabled Oracle HTTP Server and SSL-enabled OracleAS Web Cache

1. Make a backup copy of `targets.xml` located in the *`MID_TIER_ORACLE_HOME`*`/sysman/emd` directory.
2. Open *`MID_TIER_ORACLE_HOME`*`/sysman/emd/targets.xml` in a text editor.
3. Search for OracleAS Portal targets, that is, `TYPE="oracle_portal"`.
4. Change the `HTTPProtocol` property value from `http` to `https`. For example:

   ```
   <Property NAME="HTTPProtocol" VALUE="https"/>
   ```

5. Change the `HTTPPort` property value so that it matches the SSL-enabled Oracle HTTP Server listening port value. For example:

   ```
   <Property NAME="HTTPPort" VALUE="7782"/>
   ```

   To verify the current listening port of the Oracle HTTP Server, look in `httpd.conf` and determine the value of the `Listen` property. Alternatively, click the Site link on the OracleAS Web Cache Administration page and determine the port number displayed in the Origin Servers field.

6. Edit the `PortalListeningHostPort` property, which is the OracleAS Web Cache route for some of the status metrics. Replace `http` with `https`. For example:

   ```
   <Property NAME="PortalListeningHostPort" VALUE=https://<host>:<Web Cache port>/>
   ```

7. Save the changes to `targets.xml`.

8. Configure Enterprise Manager to monitor the SSL-Enabled targets. To do this, follow the instructions provided in the *Oracle Application Server Administrator's Guide*.

7.4 Viewing OracleAS Portal Activity Reports

You can choose how objects and actions are logged in OracleAS Portal and generate reports for analyzing the data. For example, you can add an entry into the Activity Log tables every time OracleAS Portal users create, edit or delete a particular page.

Any authorized user can view the OracleAS Portal Log Registry records. However, only the portal administrator can set up what information is to be logged. See Section 7.4.2, "Choosing Which Events Are Logged" for more information.

Note: With the introduction of OracleAS Web Cache into the OracleAS Portal architecture, some actions logged in OracleAS Portal Activity Log tables have become inaccurate. These actions include View, Execute (for Reports, Charts, and Hierarchies), and Show. The Activity Log tables and views still remain in the OracleAS Metadata Repository, as all other logged actions remain accurate.

7.4.1 Logged Events

Table 7–5 lists the events that can be logged for portal objects.

Table 7–5 Logged Events for OracleAS Portal Objects

Portal Object	Event
Pages	Create, Edit, Delete, Personalize
Items	Create, Edit, Delete, Move, Check Out, Check In
Application Components	Create, Edit, Delete, Execute (except for Reports, Charts, and Hierarchies), Copy, Export, Rename, Generate, Access Control, Manage, Insert, Update, Save
Portlets	Add to Page, Delete from Page
Portlet Instances	Hide, Personalize
Searches	Search

Note: User and Group actions such as Create, Edit, and Delete are logged by Oracle Internet Directory and may be viewed from Oracle Directory Manager, if logging is enabled. For more information, refer to the *Oracle Internet Directory Administrator's Guide*.

7.4.2 Choosing Which Events Are Logged

You can choose which events are logged in OracleAS Portal Log Registry records.

1. In the **Services** portlet, click **Log Registry Administration**.

Note: By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

The Administer Log Registry page is displayed as shown in Figure 7–15.

Figure 7–15 Administer Log Registry Page

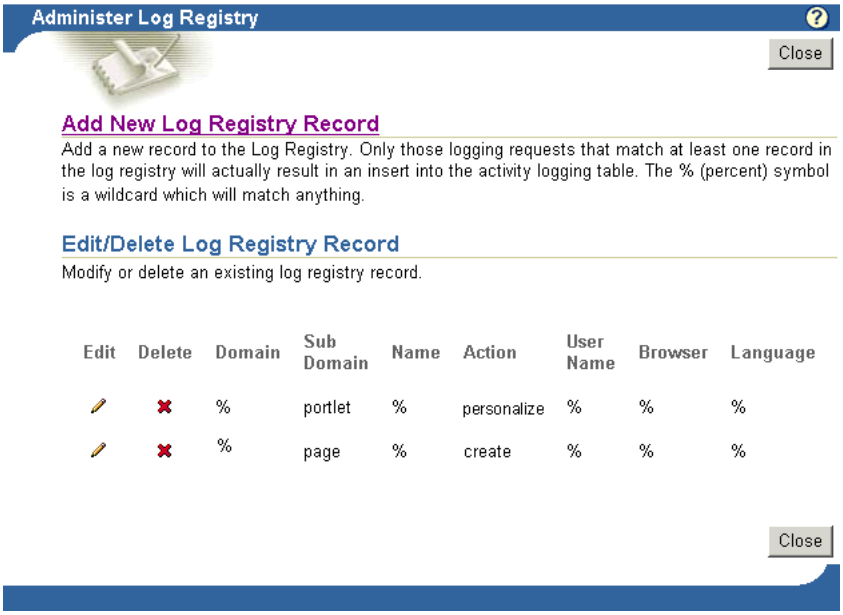


Figure 7–15 shows two logging requests. The first creates an entry in the Activity Log every time a portlet is personalized. The second creates an entry every time a page is created. If you want to log all possible requests, choose % for each field.

2. Do one of the following:

   Click **Add New Log Registry Record** to create a new Log Registry record and specify logging criteria.

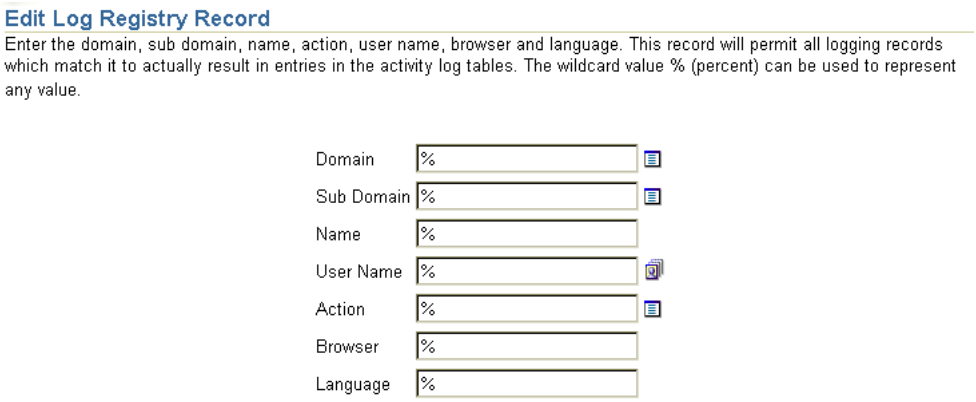
   Or,

   Edit logging criteria for an existing Log Registry record. To do this, perform the following steps:

   a. Click the **Edit** icon to edit logging criteria for an existing Log Registry record (under **Edit/Delete Log Registry Record**).

      The Edit Log Registry Record page is displayed as shown in Figure 7–16.

Figure 7–16 Edit Log Registry Record page



b. Choose the objects that you wish to log, from the **Sub Domain** list. Valid objects are listed in Table 7–5.

c. Choose which actions (or events) you want to log, from the **Action** list. Valid actions are listed in Table 7–5.

d. Specify other logging criteria as required.

e. Click **OK**.

7.4.3 Activity Log Views

Several Activity Log views are available (named wwlog_*). These views exist in the schema in which OracleAS Portal is installed. These views are granted to public; however, the logs are secure according to the object's security. For example, information about pages is available only on pages for which the user has access privileges.

Table 7–6 lists all the Activity Log views and their descriptions. You can create simple OracleAS Portal DB Provider reports and charts based on these views if required.

Table 7–6 Activity Log Views

Log View	Description
wwlog_portal_admin_logs	All logs (only has records if the user is the portal administrator).
wwlog_user_logs	All logs created by current user.
wwlog_all_portlet_logs	Portlet instances on pages that the current user can view.
wwlog_all_document_logs	Documents that the current user can view.
wwlog_all_search_logs	Searches that the current user can view.
wwlog_all_item_logs	Items that the current user can view.
wwlog_all_component_logs	Components that the current user can view.
wwlog_all_object_logs	Summary view, which encompasses all the preceding views.

7.4.4 Accessing Activity Log Views Externally

You can also access information in the Activity Log views from outside of the OracleAS Portal browser-based interface, that is, using SQL*Plus, OracleAS Reports

Services, and so on. To do this, you must first set the portal security context for your database session using the **`wwctx_api.set_context`** API:

```
wwctx_api.set_context (
   p_user_name => 'portal_username',
   p_password  => 'portal_pw'
);
```

7.5 Viewing Oracle Application Server Port Information

In Application Server Control Console, the **Application Server Ports** page shows a list of all the ports currently in use by the components of a particular Oracle Application Server instance. This page is important when you are troubleshooting port conflicts among the various Oracle Application Server components.

Whenever possible, Application Server Control Console provides a link to the appropriate Oracle Enterprise Manager 10*g* configuration page where you can modify the port settings for the component.

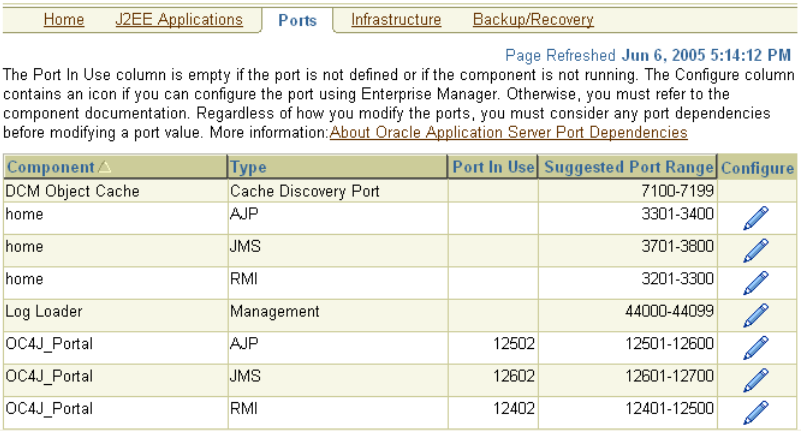
To access port information for your Oracle Application Server:

1. Access the Application Server Control Console. See Section 7.2.1, "Accessing the Application Server Control Console" for details.

   If there is more than one standalone application server instance, your start page for the Application Server Control Console is the Oracle Application Server **Farm** home page.

2. Click an instance to take you to the Oracle Application Server instance home page.
3. Click the **Ports** link to display port information, as shown in Figure 7–17.

Figure 7–17 Oracle Application Server Ports Page



For information on managing ports, refer to the *Oracle Application Server Administrator's Guide*.

8

Configuring the Search Features in OracleAS Portal

This chapter provides information on setting up the search capabilities in OracleAS Portal. This includes how to set up Oracle Text and maintain Oracle Text indexes.

This chapter contains the following sections:

8.1 Search Options in OracleAS Portal

OracleAS Portal offers powerful search capabilities that you can customize according to your needs. A robust set of built-in search portlets enables you to perform searches on the portlet repository, portal pages and external sites.

Furthermore, you can perform searches against more than 100 document types including HTML, XML, PDF, word processing formats, spreadsheets formats, presentation formats, and other common business formats.

This section introduces the search options that are available in OracleAS Portal and gives some guidance on how you can select which option is best for you:

8.1.1 OracleAS Portal Search

OracleAS Portal includes a set of built-in features tuned for searching content stored and managed within the OracleAS Portal Repository. These features are incorporated within four search portlets and they can be configured in a variety of ways:

- **Basic Search** — this portlet allows simple keyword searches.
- **Advanced Search** — this portlet enables you to enter more detailed search criteria, including operators on multiple attributes values.

- **Custom Search** — this portlet is fully customizable and enables you to design a search portlet to suit your needs, including pre-defined searches that display results in place. As this portlet is a superset of the Basic and Advanced search portlets, it can be configured to look and behave like these portlets if required.
- **Saved Searches** — this portlet enables you to repeat saved searches.

These portlets search *all* text-type metadata associated with content in the OracleAS Portal Repository. For example, display name, keyword, description, and similar attributes.

In addition to metadata, the portlets can search the portal content and this is possible as Oracle Text is enabled in OracleAS Portal by default. This means that OracleAS Portal search portlets also search in:

- **Documents/files and URL items** — file and URL items in binary format can be indexed providing the file format is filterable by Oracle Text.
- **Web pages that URLs (in URL attributes) point to** — the content must be plain text or HTML.

> **Note:** If more than one search term is specified along with an AND search operator (like *Contain All of the Terms*, *Partially Match All of the Terms*, *Sound Like All of the Terms*, and so on), then the terms must all appear within the same search index to result in a match. For example, if you enter '`weights aerobics`' and choose the *Contains All* operator, then search results are returned only when both these terms are found in item metadata, URL content, or document content. If the term `weights` is found in URL content and the term `aerobics` is found in document content, then this does not result in a match.

To find out how to configure Oracle Text for use in OracleAS Portal, see Section 8.2.2, "Configuring Oracle Text Options in OracleAS Portal". To learn more about Oracle Text, how to maintain Oracle Text indexes, and for troubleshooting information, see Section 8.3, "Oracle Text".

To find out how you can configure OracleAS Portal search portlets, see Section 8.2.1, "Configuring OracleAS Portal Search Portlets". To learn more about OracleAS Portal search portlets and how to add search functionality to OracleAS Portal pages, refer to the *Oracle Application Server Portal User's Guide*.

Disabling Oracle Text

Out-of-the-box, Oracle Text is enabled. Although Oracle does not recommend that you disable Oracle Text, it is possible to do so, if your portal does not require or would not benefit from full text searches for OracleAS Portal Repository content. For more information, see Section 8.3.1.1, "Searching With Oracle Text Disabled".

Search Results and Content Security

OracleAS Portal search result pages can display items, pages, categories, or perspectives that meet your search criteria. Refer to Section 8.1.3, "Default Search Functionality" for more information. Search results do not include:

- Content that you are not authorized to view
- Content that has expired, or is not yet published
- Page content that is derived from a template

- Portlet instance items, Portal Smart Links, and Navigation pages

Page designers can choose whether to display links to associated objects with each search result. For example, users may see links to the page group, page, category and perspective associated with an item. However, users who click such links are denied access to the object, if they do not have the required access privileges.

8.1.2 Oracle Ultra Search

Oracle Ultra Search is an application built on Oracle Text that provides an enterprise search capability over a variety of content repositories and data sources, including the OracleAS Portal Repository. Oracle Ultra Search is installed and preconfigured for use within OracleAS Portal and includes a search portlet that can be embedded in OracleAS Portal pages.

From this portlet, a user can enter a search term and launch a search that returns a single result set that includes content from all configured data sources. When OracleAS Portal is configured as one of the data sources, the search can return only *public* OracleAS Portal content.

The *Oracle Ultra Search User's Guide* provides detailed configuration instructions for Oracle Ultra Search and is available on the Oracle Technology Network (OTN) at `http://www.oracle.com/technology/`.

See Section 8.2.4, "Configuring Oracle Ultra Search Options in OracleAS Portal" to learn how to register OracleAS Portal as an Oracle Ultra Search content source and make the Ultra Search portlet available in OracleAS Portal.

8.1.3 Default Search Functionality

After a standard OracleAS Portal installation you can start using the search features in OracleAS Portal right away. Without any additional configuration, you can place one of the built-in OracleAS Portal search portlets on a page and use it to search portal content.

During installation, Oracle Text indexes are created and synchronized and Oracle Text searching is enabled in your portal. However, it is important to note that new or modified content (items, pages, categories, perspectives) is not returned in search results until the Oracle Text indexes are next synchronized. See Section 8.3.5.1, "Synchronizing Oracle Text Indexes".

By default, Oracle Text indexes are scheduled to synchronize hourly by a job that calls `wwv_context.sync`. If you find that the default synchronization interval is not suitable for your portal you can modify it at any time. For details, see.Section 8.3.5.5, "Deciding How Often to Synchronize Oracle Text Indexes".

If you are using Oracle Database 10*g*, you can specify that Oracle Text indexes synchronize automatically whenever portal objects are added, modified, or deleted. This feature is useful for portal applications where newly added or altered content must be searchable immediately. To find out more, see Section 8.3.5.2, "Synchronizing an Oracle Text Index On Commit". This feature is not available on databases earlier than Oracle Database 10*g*.

Note: If you do not want to make use of the additional features provided by Oracle Text, then you can disable this feature. See Section 8.2.2.1, "Enabling and Disabling Oracle Text in OracleAS Portal".

Table 8–1 shows some other default search settings. See Section 8.2, "Configuring OracleAS Portal Search Options" for information about how to change the values listed here.

Table 8–1 Default Search Settings

Search Setting Option	Default
Results Page - Basic Search Portlets and Basic Search Box Items	Basic Search Results Page
Results Page - Advanced, Custom and Saved Search Portlets	Search Results Page
Advanced Search Link	Advanced Search Page
Internet Search Engine Link	None
Hits per Page	20
Oracle Text	Enabled
Oracle Text - Themes And Gists	Disabled
Oracle Text - Highlight Text Color	Default
Oracle Text - Highlight Text Style	Plain
Oracle Text - Base URL	http://<host>:<port>/portal/pls/<dad>

The following images show default search portlets and pages:

Figure 8–1 OracleAS Portal Basic Search Portlet

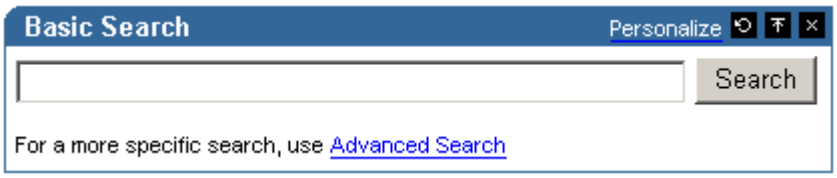


Figure 8–2 OracleAS Portal Basic Search Results Page

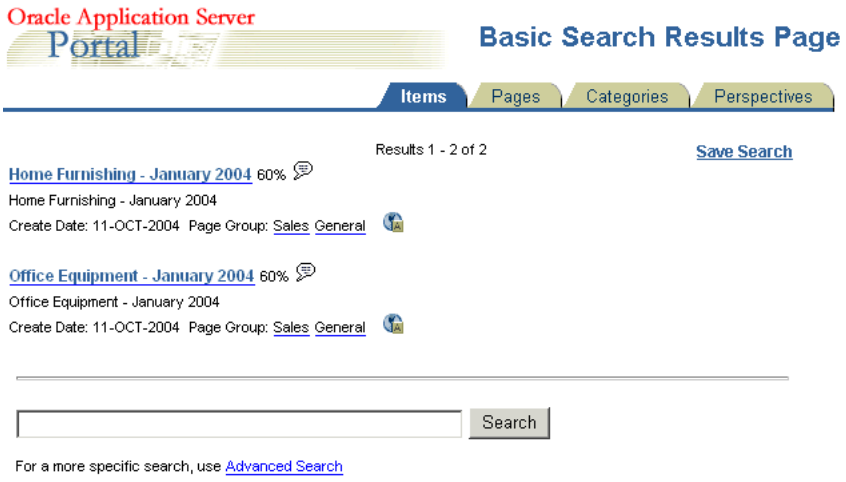


Figure 8–3 OracleAS Portal Advanced Search Portlet

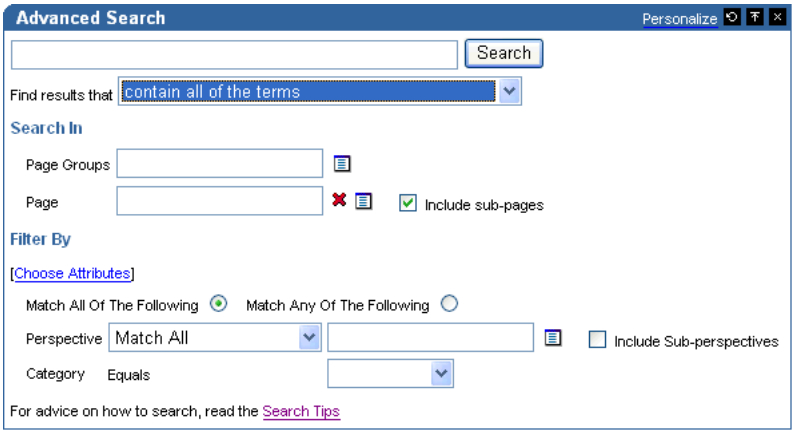


Figure 8–4 OracleAS Portal Custom Search Portlet

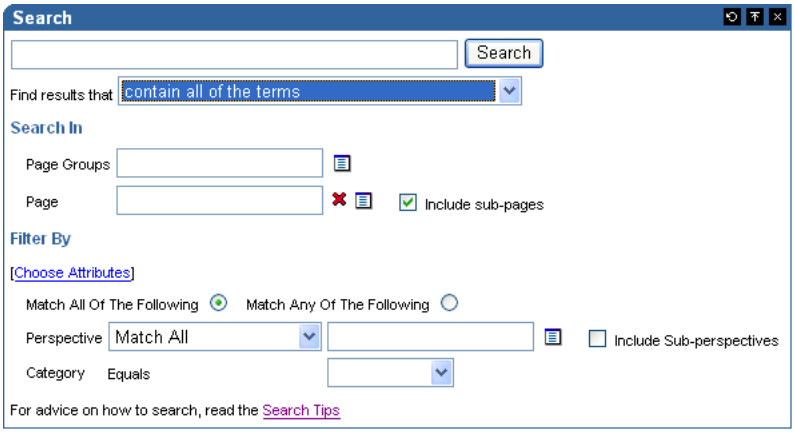


Figure 8–5 OracleAS Portal Search Results Page

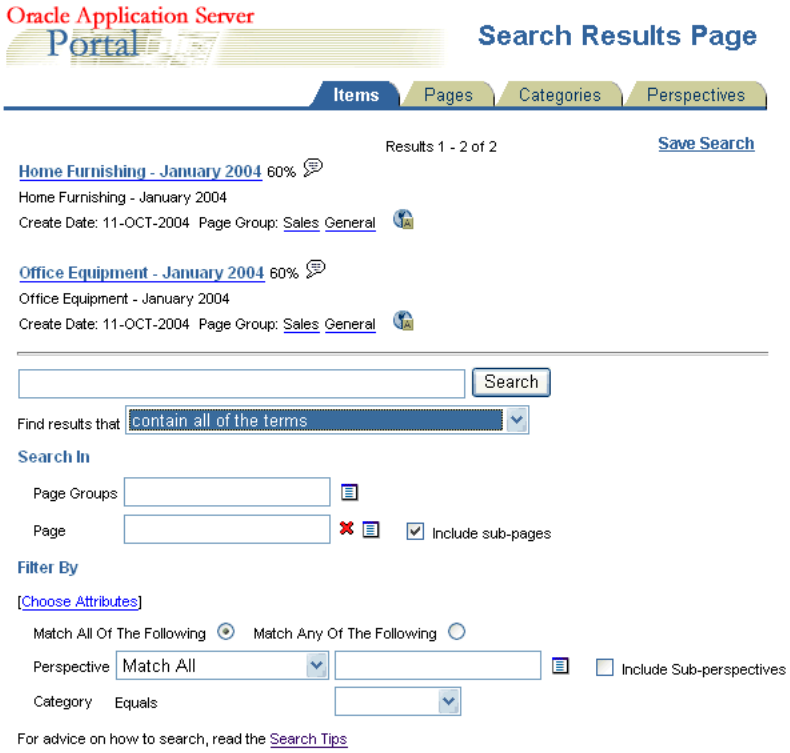


Figure 8–6 OracleAS Portal Saved Searches Portlet

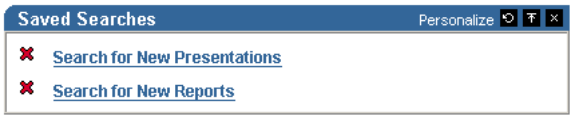
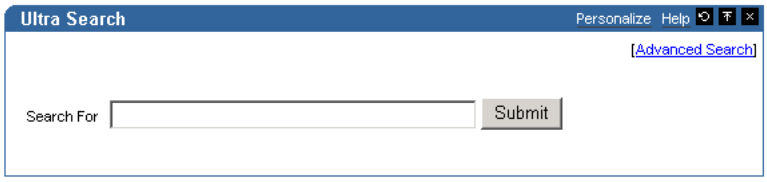


Figure 8–7 Oracle Ultra Search Portlet



8.1.4 Deciding Which Search Options to Use

Choosing how to configure searching within OracleAS Portal begins with a careful examination of your goals for the search experience and understanding of your portal content. Some key questions include:

- **Searching 'breadth'** - do you wish to limit the results returned from your portal search to content managed within the OracleAS Portal Repository, or do you want to return results from other repositories?
- **Searching 'depth'** - is full text indexing of document content a key requirement, or is a metadata only index sufficient?

- **Content security policies and portal user profiles** - is your search experience targeted at primarily public, unauthenticated users searching public content or is it more targeted at individual users who have various levels of access privileges to the content?
- **Advanced searching features** - is the ability to order results by relevancy, view document themes and gists, and other features of Oracle Text an important capability to offer your users?
- **Administration** - how much time are you willing to invest in administering and maintaining indexes, data sources, and so on?

Use Table 8–2 to help match your search requirements to the most appropriate search configuration:

Table 8–2 OracleAS Portal Search Options

	OracleAS Portal (Oracle Text disabled)	OracleAS Portal (Oracle Text enabled)	Oracle Ultra Search
Searching 'Breadth'	OracleAS Portal Repository only	OracleAS Portal Repository only	OracleAS Portal Repository and other repositories
Searching 'Depth'	OracleAS Portal metadata only	Full text index	Full text index. For OracleAS Portal, public content only.
Content security and user profiles	Returns secure and public content in search results	Returns secure and public content in search results	Returns public content only
Advanced searching features	No	Yes	Yes
Administration	Minimal	Maintain full text indexes	Maintain full text indexes and configure data sources

8.1.5 Differences Between Oracle Ultra Search and OracleAS Portal Search

This section highlights the main differences between Oracle Ultra Search and OracleAS Portal Search.

- Oracle Ultra Search only crawls public content

  OracleAS Portal is exposed to Oracle Ultra Search as a file system, and to see content in a folder, the folder must be public. If it is not public, none of the content from the folder or the sub-folder hierarchy is crawled. If you create a piece of content and make it public, then it is only indexed if all the containing folders are also public.

- Oracle Ultra Search returns a single list of pages and items

  To Oracle Ultra Search, both OracleAS Portal pages and items are resources with metadata and content, or a visual representation that can be crawled, indexed, and returned in search results. This means that, Oracle Ultra Search can return a search result list that contains both pages and items. OracleAS Portal Search searches for distinct types of data (pages, items, categories and perspectives) and only one type of data can be searched at a time. Whilst Oracle Ultra Search does not treat categories and perspectives as separate searchable entities, it can (like OracleAS

Portal Search), search for items and pages that have a particular perspective or category.

- Oracle Ultra Search searches content of displayed pages in addition to metadata

  OracleAS Portal Search searches page and item metadata. The Oracle Ultra Search crawler sees the rendered content plus the metadata. This means that Oracle Ultra Search can return results when OracleAS Portal search does not return any.

- OracleAS Portal Search excludes some item types

  OracleAS Portal Search can only return items of the following base item types:

  – <None> that is, no base item type
  – Base File
  – Base URL
  – Base Text
  – Base PL/SQL
  – Base Page Link
  – Base Image
  – Base Image Map

  Oracle Ultra Search indexes the visualization of any item type that appears on a page, irrespective of the base item type, as it is the page rendition that is indexed. This means that all the content on the page, static and dynamic, is indexed by Oracle Ultra Search including banners and template items, login/logout links, and so on.

- Oracle Text and scoring systems

  Both Oracle Ultra Search and OracleAS Portal Search use Oracle Text to index their content, however their implementations are different. Furthermore, Oracle Ultra Search uses a slightly different scoring system to OracleAS Portal Search, and it may be customized. See the *Oracle Ultra Search User's Guide* available from OTN at `http://www.oracle.com/technology/`.

  Both scoring systems give weighting when a search term is found in the title, so title hits will score more highly than hits in the document content. In OracleAS Portal searches, the score ranks even higher when there are multiple terms in the title, and weighting is also given when multiple terms are found close together or to search results that contain the most hits.

- Oracle Ultra Search crawls external content

  Oracle Ultra Search can crawl content outside of OracleAS Portal, that is, external Web sources. OracleAS Portal searches are restricted to internal content.

8.2 Configuring OracleAS Portal Search Options

The OracleAS Portal search feature is installed with defaults so you can start using the search features right away. Refer to Section 8.1.3, "Default Search Functionality" for a description of these initial defaults.

This section describes how you, the portal administrator, can configure aspects of the search feature that affect *all* search portlets:

- Configuring OracleAS Portal Search Portlets

- Configuring Oracle Text Options in OracleAS Portal
- Configuring Enterprise Search Engine Options
- Configuring Oracle Ultra Search Options in OracleAS Portal

8.2.1 Configuring OracleAS Portal Search Portlets

This section describes how to configure aspects of the search feature that affect *all* OracleAS Portal search portlets:

- Choosing Search Result Pages
- Limiting the Number of Search Results on a Page
- Choosing an Advanced Search Link (Basic/Custom Search Portlets)
- Choosing an Internet Search Engine (Advanced/Custom Search Portlets)

8.2.1.1 Choosing Search Result Pages

You can determine which page is used to display search results from:

- Basic Search portlets and Basic Search Box items
- Advanced, Custom, and Saved Searches portlets

If you choose a different search result page, then it is applied to both new and existing search portlets.

You can override this setting for a particular Custom Search portlet, if required. A Custom Search portlet only uses the result page specified here, if the **Where should the search results be displayed?** option (on Edit Defaults: Results Display tab) is set to the Default Search Results Page. For more information on how to set options for the Custom Search portlet, refer to the *Oracle Application Server Portal User's Guide*, available from OTN at `http://www.oracle.com/technology/products/ias/portal/documentation.html`.

To specify a search result page for OracleAS Portal search portlets:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Search** tab.
3. In the Search Results Pages section, for **Basic Search Portlets and Basic Search Box Items**, choose a suitable search results page.

   You can choose any portal page that contains a search portlet. If you select a page without a search portlet, then no results are displayed. The default is the *Basic Search Results Page.*

4. For **Advanced, Custom and Saved Search Portlets**, choose a suitable search results page.

   You can choose any portal page that contains a search portlet. If you select a page without a search portlet, then no results are displayed. The default is the *Search Results Page*.

5. Select **OK**.

> **Note:** If page caching is enabled, then the change may not be seen in existing search portlets immediately. The cache is cleared automatically every 24 hours for all search portlets. Alternatively, clear the cache manually using OracleAS Web Cache Manager, accessible though the Web Cache Administration page in Application Server Control Console. See Section 7.2, "Using the Application Server Control Console".

If a page you select is subsequently deleted, then the associated **Page** field is empty. Choose another page and then click **OK**. If you click **Cancel**, then you will see **Page Not Found** errors after search operations.

8.2.1.2 Limiting the Number of Search Results on a Page

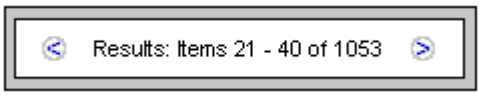
You can limit the number of results that search portlets can display. The limit is applied to Basic, Advanced, and Custom Search portlets.

You cannot change the limit for individual Basic or Advanced Search portlets. However, you can override this setting for a Custom Search portlet, if required.

If the number of results returned by a search exceeds this limit, the search results pages include Next and Previous icons so that users can view all of the results as shown in Figure 8–8. On a Custom Search portlet, these icons may be hidden, if required.

For more information on how to set options for the Custom Search portlet, refer to the *Oracle Application Server Portal User's Guide*, available on OTN at `http://www.oracle.com/technology/products/ias/portal/documentation.html`.

Figure 8–8 Hits per Page Setting on Search Portlets



For example, if you limit *Hits Per Page* to 10, the first 10 results are displayed on the first search results page, the next 10 on the second page, and so on.

> **Note:** If you change the limit, the new value does not effect existing search portlets, only new ones.

To specify the number of search results per page:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Search** tab.
3. In the **Search Properties** section, for **Hits Per Page**, enter the number of search results to display on a page.
4. Click **OK**.

8.2.1.3 Choosing an Advanced Search Link (Basic/Custom Search Portlets)

Typically, advanced searches allow a user to specify additional search criteria. For example, see Figure 8–9.

Figure 8–9 Advanced Search Link on Basic/Custom Search Portlets

The advanced search link can be to an external site, another portal page, or a package call within OracleAS Portal.

An advanced search link is displayed on Basic Search portlets. Optionally, this link can be displayed on Custom Search portlets. For more information on how to set options for the Custom Search portlet, refer to the *Oracle Application Server Portal User's Guide*, available on OTN at `http://www.oracle.com/technology/products/ias/portal/documentation.html`.

You can determine the destination of the Advanced Search Link, for all Basic/Custom Search portlet instances. When you specify a new Advanced Search Link, it is applied to both new and existing search portlets that display an Advanced Search link.

To enter advanced search link details:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Search** tab.

3. In the **Advanced Search Link** section, do one of the following:

   - Specify a destination **Page Name** for the **Advanced Search** link.

     The default is the *Advanced Search Page*, which contains the built-in **OracleAS Portal Advanced Search** portlet. However, you can select any portal page displaying advanced search options, the page does not have to contain one of the OracleAS Portal search portlets. For example, you can use a JSP page containing advanced search options if one existed in your portal.

     If the page you select is subsequently deleted, this field is empty. Choose another page and then **OK**. If you click **Cancel**, the advanced search links will all still point to the deleted page.

   - Specify a **URL** for the **Advanced Search** link.

     Enter the URL you want to use. If you have created a customized search engine that you want to use for advanced searches throughout the portal, you can specify its link here.

     You can specify an absolute URL, or a relative URL. For example, `http://www.myfavoritesearchengine.com` creates a link directly to this Internet search site.

     If you enter a relative URL (that is, a portal package), the value specified here is appended to the OracleAS Portal schema URL and this results in a call to the portal package. Note how the value is appended, depending on whether the value specified begins with '/':

`/value` results in this URL: `http://<webserver>:<port>/<value>`

`value` results in this URL:
`http://<webserver>:<port>/portal/pls/<dad>/<value>`

4. Select **OK**.

Note: If page caching is enabled, the change may not be seen in existing search portlets immediately. The cache is cleared automatically every 24 hours for all search portlets. Alternatively, clear the cache manually using OracleAS Web Cache Manager, accessible though the Web Cache Administration page in Application Server Control Console. See Section 7.2, "Using the Application Server Control Console".

8.2.1.4 Choosing an Internet Search Engine (Advanced/Custom Search Portlets)

An Internet search engine link is displayed on Advanced Search portlets. So, if users do not find the information they need when they search OracleAS Portal, they can extend their search using an Internet Search Engine. See Figure 8–10.

Figure 8–10 Internet Search Engine Link on Advanced/Custom Search Portlets

Optionally, this link can be displayed on Custom Search portlets. For more information on how to set options for the Custom Search portlet, refer to the *Oracle Application Server Portal User's Guide*, available on OTN at `http://www.oracle.com/technology/products/ias/portal/documentation.html`.

When you specify the URL of an Internet search engine and the link text that users click to access the specified Internet search engine, it applies to all new and existing Advanced/Custom Search portlet instances that display an Internet search link.

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Search** tab.

3. In the **Internet Search Engine** section, for **URL**, enter the URL of an Internet search engine. For example, `http://www.myinternetsearch.com`.

   The URL must be fully formed. It must include http:// and any associated parameters.

4. For **Link Text,** enter the text that users click to access the specified Internet search engine. For example: `MySearch`

   If you enter `MySearch`, this text is displayed as a link in Advanced Search portlets and optionally in Custom Search portlets. See Figure 8–10.

5. Select **OK**.

If the Internet Search Engine properties (URL and Link Text) are not specified, none of the Advanced or Custom Search portlets display an Internet search engine link.

Note: If page caching is enabled, the change may not be seen in existing search portlets immediately. The cache is cleared automatically every 24 hours for all search portlets. Alternatively, clear the cache manually using OracleAS Web Cache Manager, accessible though the Web Cache Administration page in Application Server Control Console. See Section 7.2, "Using the Application Server Control Console".

8.2.2 Configuring Oracle Text Options in OracleAS Portal

This section describes how to configure Oracle Text features in OracleAS Portal:

- Enabling and Disabling Oracle Text in OracleAS Portal
- Setting Oracle Text Search Result Options
- Setting a Base URL for Oracle Text
- Configuring Proxy Settings for Oracle Text

Note: If page caching is enabled, changes to Oracle Text search settings may not be seen in existing search portlets immediately. The cache is cleared automatically every 24 hours for all search portlets. Alternatively, clear the cache manually using OracleAS Web Cache Manager, accessible though the Web Cache Administration page in Application Server Control Console. See Section 7.2, "Using the Application Server Control Console".

See Section 8.3, "Oracle Text" for more information about Oracle Text, how to maintain Oracle Text indexes, and troubleshooting information. Refer to Appendix H, "Using TEXTTEST to Check Oracle Text Installation" for checking that Oracle Text is installed and working correctly.

8.2.2.1 Enabling and Disabling Oracle Text in OracleAS Portal

Oracle Text extends the searching capabilities of OracleAS Portal. See Section 8.1.1, "OracleAS Portal Search". Out-of-the-box, Oracle Text is always enabled. Although Oracle does not recommend that you disable Oracle Text, it is possible to do so, if your portal does not require or would not benefit from full text indexing content within the OracleAS Portal Repository. See also Section 8.2.2.1, "Enabling and Disabling Oracle Text in OracleAS Portal".

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Search** tab.

3. Select **Enable Oracle Text Searching** to make use of Oracle Text when searching OracleAS Portal.

   Deselect this option at any time to disable the use of Oracle Text.

Note: If you see the message `Oracle Text is not installed`, Oracle Text is not installed in the database and is not available in OracleAS Portal. Arrange with your database administrator to have Oracle Text installed. Once installed, you must run the following command in SQL* Plus to create the Oracle Text role:

```
inctxgrn.sql
```

This file is located in the directory *ORACLE_HOME*`/portal/admin/plsql/wws`.

Log on using the user name and password for the PORTAL schema. You must also create Oracle Text indexes. See Section 8.3.4, "Creating and Dropping Oracle Text Indexes" for more information.

---

4. Click **OK**.

8.2.2.2 Setting Oracle Text Search Result Options

When Oracle Text is enabled, you can display additional information alongside items (documents/files) when they are returned as search results. For each item returned you can view:

- Major **themes** in a chart. A theme shows the nouns and verbs that occur most frequently.
- A short summary about the content (**gist**). Gists are derived from how frequently those nouns and verbs appear.
- An HTML version.
- An HTML version of the file with search terms highlighted in a specific color and font.

Themes and gists are optional and HTML highlighting can be customized as follows:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Search** tab.
3. Select **Enable Themes And Gists** to create a theme and gist for each item returned by the search.

Note: Themes and gists are not available for all languages.

---

4. For **Highlight Text Color**, select the color to highlight search terms found in the HTML version of items returned by the search.
5. For **Highlight Text Style**, select the style to apply to search terms found in the HTML version of the items returned by the search.
6. Click **OK**.

8.2.2.3 Setting a Base URL for Oracle Text

Oracle Text needs a *base URL* to resolve relative URLs into fully qualified absolute URLs. See Section 8.3.6.1, "Relative URLs" for more information.

To specify the Base URL for Oracle Text:

1. In the **Services** portlet, click **Global Settings**.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Click the **Search** tab.

3. Enter the **Oracle Text Base URL** in the format:
   `http://<host>:<port>/portal/pls/<dad>`

   For example: `http://myportal.com:4000/portal/pls/design`

   If no value is specified, no relative URLs are indexed and therefore, any URL content that relative URLs points to, cannot be searched.

4. Click **OK**.

8.2.2.4 Configuring Proxy Settings for Oracle Text

Oracle Text uses OracleAS Portal proxy server settings to access URL content. This is necessary when OracleAS Portal lies behind a firewall and URL items point to content beyond this firewall. See Section 8.3.6.4, "URL Index Proxy Settings" for more information.

Refer to Section 5.5, "Configuring OracleAS Portal to Use a Proxy Server" for information about configuring the global proxy settings for OracleAS Portal.

8.2.3 Configuring Enterprise Search Engine Options

Several *Enterprise Search Engine* options are displayed on the Global Settings: Search tab. These options are reserved for future use.

In the future, other enterprise search engines, such as Oracle Ultra Search and FAST (partner technology), will be able to search secure portal content. When this technology becomes available, more documentation will be published on OTN at `http://www.oracle.com/technology/products/ias/portal/documentation.html`.

8.2.4 Configuring Oracle Ultra Search Options in OracleAS Portal

This section describes how to set up Oracle Ultra Search for use in OracleAS Portal. You must complete the tasks in this section, before you can add the Ultra Search portlet to a portal page and use this feature:

- Accessing the Oracle Ultra Search Administration Tool
- Registering OracleAS Portal as a Content Source
- Registering the Ultra Search Provider with OracleAS Portal

> **Note:** Before using Oracle Ultra Search features in OracleAS Portal, also ensure that all necessary database and middle-tier configuration is complete. For detailed information, see the *Oracle Ultra Search User's Guide*.

You will find additional information in the paper "*Setting Up Oracle Ultra Search for OracleAS Portal 10g*" on OTN, `http://www.oracle.com/technology/products/ias/portal/`.

8.2.4.1 Accessing the Oracle Ultra Search Administration Tool

1. Click **Ultra Search Administration** in the **Services** portlet.

   By default, the **Services** portlet is on the **Portal** subtab of the **Administer** tab on the **Portal Builder** page.

2. Log in.

If OracleAS Portal was configured using Oracle Enterprise Manager, the Oracle Ultra Search instance is not configured automatically and therefore the **Ultra Search Administration** link in OracleAS Portal does not work. To set this up you must create an Oracle Ultra Search instance. For instructions, see the *Oracle Ultra Search User's Guide*, available on OTN at `http://www.oracle.com/technology/`.

8.2.4.2 Registering OracleAS Portal as a Content Source

1. Access the Oracle Ultra Search administration tool. Refer to Section 8.2.4.1, "Accessing the Oracle Ultra Search Administration Tool" for details.
2. On the **Instances** tab, click **Apply** to set the instance.

   If you have more than one instance make sure to select the instance you want to manage first.
3. On the **Crawler** tab, enter the **Cache Directory Location** and the **Crawler Log File Directory**.

   These directory locations are on the computer where Oracle Application Server middle tier is installed.
4. On the **Sources** tab, click the **Oracle Sources** subtab, choose **Oracle Portal (Crawlable)** from the **Create Source** drop-down list and click **Go**.

   (Optional) Edit the OracleAS Portal data source and customize the types of documents the Oracle Ultra Search crawler should process. HTML and plain text are the default document types that the crawler will always process. You can add other document types such as MS Word Doc, MS Excel Doc, PDF and so on.
5. Enter OracleAS Portal registration details:
   a. Enter the **Portal Name**.
   b. For **URL base**, enter the base URL for the portal.

      Use the format:

      ```
      http://<host>:<port>/portal/pls/<portal_DAD>/<portal_schema>
      ```

      For example:

      ```
      http://myserver.abc.com:7778/portal/pls/portal/myschema
      ```

   c. Click **Register Portal**.
6. Select the page groups that you would like to create data sources for and then click **Create Portal Data Sources**.

   You can optionally edit each of the portal data sources to add content types for processing. For example, you can add the MS Word Doc, MS Excel Doc, PDF Doc types.

Note: A page group is available as a crawlable data source, when either:

- The option **Display Page to Public Users** is set on its root page (**Edit Page:Access** tab).
- The *View* privilege is granted to PUBLIC (**Edit Page Group: Access** tab).

For more information, see the *Oracle Application Server Portal User's Guide*.

7. Finally, on the **Schedules** tab, schedule the indexing of the portal data sources:
   a. Click **Create New Schedule** and enter a **Name** for the schedule.
   b. Click **Proceed to Step 2** and specify synchronization schedule details.
   c. Click **Proceed to Step 3**, select **Portal** from the drop down list and then click **Get Sources**.
   d. Move the sources over to the **Assigned Sources** box and click **Finish**.

   Clicking the **Status** link for the source enables you to optionally run the synchronization immediately.

Once you have registered OracleAS Portal as an Oracle Ultra Search content source, you can register the Ultra Search provider with OracleAS Portal.

8.2.4.3 Registering the Ultra Search Provider with OracleAS Portal

OracleAS Portal comes with a pre-built sample portlet for Oracle Ultra Search. To access the portlet the provider must be registered with OracleAS Portal.

1. In the **Remote Providers** portlet, click **Register a Provider**.

   By default, the **Remote Providers** portlet is on the **Portlet** subtab of the **Administer** tab on the **Portal Builder** page.
2. Specify a **Name** and the following:
   - **Timeout** - how long pages take to render if the portlet is not responding, so do not set it too high.
   - **Implementation Style** - leave as Web.
   - Click **Next** to continue.
3. Enter the **URL** for the Ultra Search provider.

   By default this is:

   ```
   http://computer.domain:7778/provider/ultrasearch/servlet/soaprouter
   ```
4. Set the **Service ID** to be `ultrasearch`.
5. Change the **Login Frequency** to **Once per User Session** and then click **Next**.
6. Click the **Browse Groups icon**, select AUTHENTICATED_USERS and grant Execute privileges.
7. Click **Finish**.

Once the provider is registered with OracleAS Portal, you can add the Ultra Search portlet to portal pages.

Note: Check an entry exists for Oracle Ultra Search in the OC4J_Portal configuration file `data-sources.xml`. For detailed instructions, see the *Oracle Ultra Search User's Guide*, available on OTN at `http://www.oracle.com/technology/`.

If the entry is missing, the Ultra Search portlet cannot access the Oracle Ultra Search instance and you will see the following error when the portlet is placed on the page:

```
oracle.ultrasearch.query.SearchException: WKG17005: connection
failure: data source does not exist: jdbc/UltraSearchPooledDS not
found
```

When you create or register a new provider, a page is created in the Portlet Repository under *Portlet Staging Area* to display portlets for that provider. This page is not visible to all logged in users. It is only visible to the user who published the provider, and the portal administrator. The publisher or portal administrator can change the provider page properties to grant privileges to appropriate users and groups, as required.

8.3 Oracle Text

Oracle Text adds powerful text search and intelligent text management to the Oracle Database. OracleAS Portal uses the Oracle Text functionality to extend its search capabilities.

Out-of-the-box, Oracle Text is always enabled. However, use of Oracle Text with OracleAS Portal is an optional feature that can be disabled by a portal administrator. For more information, see Section 8.3.1, "Understanding OracleAS Portal Searches with Oracle Text Enabled/Disabled".

The use of Oracle Text with OracleAS Portal is described in the following sections:

You will find additional information in the Oracle Text documentation, available on OTN at `http://www.oracle.com/technology/`.

8.3.1 Understanding OracleAS Portal Searches with Oracle Text Enabled/Disabled

Out-of-the-box, Oracle Text is always enabled. Although Oracle does not recommend that you disable Oracle Text, it is possible to do so, if your portal does not require or would not benefit from full text indexing OracleAS Portal Repository content.

See Section 8.2.2.1, "Enabling and Disabling Oracle Text in OracleAS Portal" for information about disabling Oracle Text.

8.3.1.1 Searching With Oracle Text Disabled

If Oracle Text is disabled and you perform a basic search (enter a search term only), the following metadata is searched:

- Item attributes (Display Name, Description, Keywords, Author)
- Page attributes (Display Name, Description, Keywords)
- Category and perspective attributes (Display Name, Description)

A basic search does not search custom attributes.

If more than one search term is specified along with the search operator *Contains All of the Terms*, then the terms must all appear within the same attribute to result in a match. For example, if you enter `weights aerobics`, search results are returned only when both these terms are found in a single attribute, such as Description. If the term `weights` is found in Description and the term `aerobics` is found in Display Name, then this does not result in a match.

It is also worth noting that fewer search operators are available when Oracle Text is disabled. Only three search operators are available for the main search term: *Contain All of the Terms*, *Contain Any of the Terms*, and *Contain these Terms Exactly*. There are fewer operators for attribute searches too.

Searches that specify criteria against selected attributes (advanced searches), matches against the selected attributes. No file- or URL- based attributes (including the seeded attributes URL and File Name) appear on advanced search forms as these are not searchable when Oracle Text is disabled. Similarly, page designers editing Custom Search portlets are prevented from selecting file- and URL- based attributes as search criteria. If any search portlet specified a file- or URL- based attribute before Oracle Text was disabled, the attribute appears greyed out and italicized if Oracle Text is subsequently disabled.

8.3.1.2 Searching With Oracle Text Enabled

If Oracle Text is enabled when you perform a basic search, all text-type attributes, including custom text attributes are searched. Furthermore, the content of file and URL items are searched.

Documents/file and URL items in binary format can be searched providing that the file format is filterable by Oracle Text. In addition, Web pages that URLs (in URL attributes) point to can also be searched, providing that the content is plain text or HTML. For more information, see Section 8.3.3.1, "Oracle Text Index Overview".

8.3.2 Oracle Text Prerequisites

Oracle Text is a standard component of the Oracle Database 10*g*. If you want to use the Oracle Text functionality in OracleAS Portal, it is essential that the Oracle Text component is correctly installed and functioning properly.

Ensure that:

- **Oracle Text is installed in the OracleAS Portal Repository database**. Since OracleAS Portal 9.0.2.2 and from the 3.0.9.8.4 patchset onwards, the Oracle Text component is required to be in the OracleAS Portal Repository database before the OracleAS Portal Repository can be installed. This is because some OracleAS Portal packages make reference to the `ctx_ddl` packages in the `CTXSYS` schema in which the Oracle Text component resides.
- **Oracle Text upgrade steps are complete.** In particular, during database upgrades, it is essential that any manual steps that pertain to Oracle Text are completed correctly.
- **Library path for the Oracle Text AUTO_FILTER is set correctly.** For AUTO_FILTER to function correctly, the `ctxhx` executable (called during indexing) needs to be able to load the appropriate shared libraries. See also Section 8.3.3.1, "Oracle Text Index Overview".
  - For UNIX platforms, ensure that the library path used by `ld` includes *ORACLE_HOME*`/ctx/lib` for both the TNS listener and the environment where the database is started. The library path environment variable for the different UNIX platforms are as follows:

    Solaris, Tru64 UNIX, Linux -> $LD_LIBRARY_PATH

    HP/UX -> $SHLIB_PATH and $LD_LIBRARY_PATH

    IBM AIX -> $LIBPATH

    For detailed information, see the *Oracle Text Reference*, available on OTN at `http://www.oracle.com/technology/documentation/`.

    Whenever you change the library path you must restart both the database and the listener for Oracle Text indexing operations to work. If one or both environment variables are not set, documents are not indexed as expected and the table `ctx_user_index_errors` may be full of DRG-11207, status 137 errors. See also Section 8.3.12.1, "Common Document Indexing Errors".
  - On Windows platforms, the Oracle Text DLLs are located in *ORACLE_HOME*`\bin`. Ensure that this path is included in the PATH environment variable, that is, in the environment from where the Oracle server is started.

You can use the TEXTTEST utility to check that Oracle Text is installed and working correctly. The TEXTTEST utility is located at *ORACLE_HOME*`/portal/admin/texttest/textest`. For more information, see Appendix H, "Using TEXTTEST to Check Oracle Text Installation".

8.3.3 Oracle Text Indexes

If you want to use the Oracle Text functionality in OracleAS Portal, several Oracle Text indexes are required in the OracleAS Portal schema. Details of these indexes are described in the following sections:

- Oracle Text Index Overview
- Oracle Text Index Preferences
- Datastore Procedures
- Granting CTXAPP Role to the OracleAS Portal Schema
- Multilingual Functionality (Multilexer)
- STEM Searching
- Maximizing AUTO_FILTER Performance

8.3.3.1 Oracle Text Index Overview

All required Oracle Text indexes are built automatically during OracleAS Portal installation by procedures in the package `wwv_context`.

> **See Also:** Appendix G, "Using the wwv_context APIs"

After portal installation, you can use the procedures in this package to manage the indexes, and this includes removing and re-creating the indexes. For more information, see Section 8.3.4.3, "Dropping All Oracle Text Indexes Using ctxdrind.sql" and Section 8.3.4.1, "Creating All Oracle Text Indexes Using ctxcrind.sql".

> **Note:** Oracle Text can be disabled, even when Oracle Text indexes are present. See Section 8.2.2.1, "Enabling and Disabling Oracle Text in OracleAS Portal".

Table 8–3 describes the Oracle Text indexes that are required.

Table 8–3 Oracle Text Indexes In the OracleAS Portal Schema

Index	Table.column	Purpose	Datastore type	Filter Type	Optional?
WWSBR_CORNER_CTX_INDX	wwpob_page$.ctxtxt	Index page metadata	user datastore	-	No
WWSBR_DOC_CTX_INDX	wwdoc_document$.blob_content	Index document content	direct datastore	AUTO_FILTER	Yes
WWSBR_PERSP_CTX_INDX	wwv_perspectives.ctxtxt	Index perspective metadata	user datastore	-	No
WWSBR_THING_CTX_INDX	wwv_things.ctxtxt	Index item metadata	user datastore	-	No
WWSBR_TOPIC_CTX_INDX	wwv_topics.ctxtxt	Index category metadata	user datastore	-	No
WWSBR_URL_CTX_INDX	wwsbr_url$.absolute_url	Index URL content	URL datastore	AUTO_FILTER	Yes

Most of the Oracle Text indexes use a user datastore. The exceptions are the indexes WWSBR_DOC_CTX_INDX (Document index) and WWSBR_URL_CTX_INDX (URL index):

- **Document index:** Uses a direct datastore. It indexes the document content held directly in the BLOB type `blob_content` column of the wwdoc_document$ table.
- **URL index:** Fetches the content to be indexed for each row in the wwsbr_url$ table from the location pointed to by the `absolute_url$` column.

It is possible to disable Document and URL indexing. This can improve the speed and efficiency of portal searches as searches are limited to item, page, category, and perspective metadata only. See Section 8.3.7, "Disabling Document and URL Indexing".

The Document and URL indexes both use a filter (AUTO_FILTER), to convert documents into a plain text format that is suitable for indexing:

- Binary documents are converted into plain text (providing the binary format is supported by the AUTO_FILTER).
- Plain text, HTML, XHTML, SGML, and XML documents bypass the filter and are indexed directly.
- Documents that do not need to be indexed, such as graphics, are ignored by the filter.

See also Section 8.3.3.7, "Maximizing AUTO_FILTER Performance".

Note: If OracleAS Portal is installed into a database that does not have a functional AUTO_FILTER, document and URL searching is automatically disabled as this functionality does not work without the AUTO_FILTER. See also Section 8.3.7, "Disabling Document and URL Indexing".

8.3.3.2 Oracle Text Index Preferences

Oracle Text uses preferences to configure the Oracle Text indexes used by OracleAS Portal and these preferences are created and owned by the OracleAS Portal schema. The preferences are created using the `ctx_ddl` package, which resides in the `CTXSYS` schema, and the data representing the preferences is actually stored in relational tables in the `CTXSYS` schema.

Oracle Text index preferences must exist before the indexes are created. Subsequent changes to these preferences do not take effect until the Oracle Text indexes are dropped and re-created.

The Oracle Text index preferences that are used during OracleAS Portal installation to create Oracle Text indexes can be re-created using the package `wwv_context`. Some Oracle Text index preferences can also be configured by you, the portal administrator, after installation. For example, global OracleAS Portal proxy settings are used by Oracle Text to populate the proxy preferences used in Oracle Text indexes.

See Also: Appendix G, "Using the wwv_context APIs"

Oracle Text indexes also use a number of Lexer preferences to control the linguistic aspects of the indexing. Lexer preferences are created by the script `sbrimtlx.sql` and you can run this script at any time to re-create the Lexer preferences. The script is located in the directory *ORACLE_HOME*`/portal/admin/plsql/wws`.

You will find additional information in the Oracle Text documentation on OTN, `http://www.oracle.com/technology/`.

8.3.3.3 Datastore Procedures

In an Oracle9*i* Database Server, for each of the Oracle Text indexes that use *user datastores*, a datastore procedure is created in the `CTXSYS` schema where Oracle Text is installed. The procedures are called for each row that is to be indexed for the given index. These procedures in turn call procedures in the OracleAS Portal schema.

The datastore procedures are named as follows:

- `WWSBR_THING_CTX_<user_id>`
- `WWSBR_CORNER_CTX_<user_id>`
- `WWSBR_PERSP_CTX_<user_id>`

- WWSBR_TOPIC_CTX_<*user_id*>

Where <*user_id*> is the user_id (as found in the ALL_USERS view) of the OracleAS Portal Repository schema. This suffix is required so that the procedure names do not clash, in the case where multiple OracleAS Portal repositories exist in the same database.

If for any reason these procedures do not exist, Oracle Text does not work. This might happen, for example, if the CTXSYS schema is dropped and reinstalled. In this situation, the procedures can be reinstalled by running the script inctxgrn.sql as the OracleAS Portal schema owner:

```
SQL> @inctxgrn.sql
```

This script also grants the CTXAPP role to the OracleAS Portal schema. See Section 8.3.3.4, "Granting CTXAPP Role to the OracleAS Portal Schema" for details. The script is located in the directory *ORACLE_HOME*/portal/admin/plsql/wws.

In Oracle Database 10*g*, the datastore procedures are not created in the CTXSYS schema. Instead, the procedures are owned by the index owning schema, that is, the OracleAS Portal schema. In this case, all the procedures are in the package wwsbr_ctx_procs:

- wwsbr_ctx_procs.thing_ctx
- wwsbr_ctx_procs.corner_ctx
- wwsbr_ctx_procs.perspective_ctx
- wwsbr_ctx_procs.topic_ctx

As the procedures are in the OracleAS Portal schema, <*user_id*> suffixes are not required.

8.3.3.4 Granting CTXAPP Role to the OracleAS Portal Schema

To use Oracle Text functionality, the role CTXAPP must be granted to the OracleAS Portal schema. This happens automatically during OracleAS Portal installation and normally no further action is required.

If for any reason this grant is revoked, Oracle Text does not work. For example, this may occur if the CTXAPP role is dropped when the CTXSYS schema is reinstalled.

To restore the necessary grants, run the script inctxgrn.sql as the OracleAS Portal schema owner:

```
SQL> @inctxgrn.sql
```

The script is located in the directory *ORACLE_HOME*/portal/admin/plsql/wws. This script also creates the OracleAS Portal user datastore procedures, which are required in the CTXSYS schema. See Section 8.3.3.3, "Datastore Procedures".

8.3.3.5 Multilingual Functionality (Multilexer)

OracleAS Portal uses the Oracle Text Multilexer to enable language-specific searching in OracleAS Portal. The Multilexer:

- Controls the way that the linguistic aspects of searching are carried out.
- Allows content, items, pages, categories, perspectives, and their translations, to be treated in a way that is appropriate to their language.

Lexer preferences are used to configure the Multilexer used for all the Oracle Text indexes. The lexer preferences are created by the script file sbrimtlx.sql. You can

modify these preferences if required, but if you do, you must drop and re-create the Oracle Text indexes for the changes to take a effect.

For more information on the Multilexer, refer to Oracle Text documentation on OTN, http://www.oracle.com/technology/.

8.3.3.6 STEM Searching

By default, STEM searching is used when Oracle Text is enabled in OracleAS Portal. STEM searching enables you to search for words that have the same root as the specified term. For example, a stem of $sing expands into a query on the words sang, sung, sing.

However, STEM searching is used only when logged in to OracleAS Portal in one of the following languages, where STEM searching is supported in Oracle Text:

```
AMERICAN ENGLISH
CANADIAN FRENCH
DUTCH
UK ENGLISH
FRENCH
GERMAN DIN
GERMAN
ITALIAN
LATIN AMERICAN SPANISH
MEXICAN SPANISH
SPANISH
```

In all other languages, the STEM operator is not used.

8.3.3.7 Maximizing AUTO_FILTER Performance

AUTO_FILTER is a universal filter that converts most document formats, such as PDF documents, into a plain text format that is suitable for indexing. In OracleAS Portal, only the Document and URL index make use of the AUTO_FILTER.

During the indexing process, AUTO_FILTER converts documents and URL content according to the following AUTO_FILTER_FORMAT settings:

- BINARY - these documents are converted into plain text (providing the binary format is supported by the AUTO_FILTER).
- TEXT - these documents bypass AUTO_FILTER and get indexed directly; for example, plain text, HTML, XHTML, SGML, and XML documents.
- IGNORE - these documents, such as images, are not filtered or indexed.

Filtering content unnecessarily can impact the speed and efficiency of portal searches, so it is important that you optimize the filtering process. The best way to optimize the use of AUTO_FILTER, is to ensure that *all* document and URL content uploaded to your portal is classified with the correct MIME type and character set:

- MIME type - In OracleAS Portal, it is the MIME type of a document that determines the AUTO_FILTER_FORMAT (the setting that AUTO_FILTER uses to determine whether a document gets filtered). For example, a document uploaded with the MIME type application/PDF gets mapped to `BINARY` and is filtered, whereas a document with the MIME type text/HTML gets mapped to `TEXT` and is indexed directly. Other documents, like images with the MIME type image/GIF, are mapped to `IGNORE`.
- Character set - AUTO_FILTER can convert documents from a non-database character set to the character set used by the database. This enables you to index

and search for documents in other character sets. If AUTO_FILTER cannot determine the character set or it is not one of the supported character sets, the document gets indexed without any character set conversion.

A default MIME type is allocated to file items and file attributes when they are uploaded to your portal. The browser determines the default MIME type but it does not always determine the setting correctly and consequently, AUTO_FILTER may filter some documents unnecessarily. Unlike file items, a default MIME type is *not* allocated to URL content. Instead, the AUTO_FILTER_FORMAT for URL content defaults to `TEXT` when no MIME type is specified. Clearly, not all URL content is plain text and therefore AUTO_FILTER may filter some URL content incorrectly too.

To ensure that all portal content gets classified and filtered properly, OracleAS Portal provides two special attributes for file- and URL- based item types: *MIME Type* and *Character Set*. By extending a built-in Base File and Base URL item type to include these attributes, users can enter the correct information when they upload portal content.

Note: Whilst the MIME Type and Character Set attributes enable you to specify the correct MIME type and character set for file- and URL-based *items*, it is not possible to specify these for File and URL *attributes*:

- File attributes - browser always determines the MIME type.
- URL attributes - MIME type is always text/html, so AUTO_FILTER always processes URL attributes as plain text.

If speed and efficiency of portal searches are important in your portal or your portal stores/references non-database character set documents, ask page group administrators to add *MIME Type* and *Character Set* attributes to all file- and URL-based item types available in their page groups. See also, *Adding Attributes to an Item Type* in the *Oracle Application Server Portal User's Guide.*

Note: When you search for portal content by MIME type or character set, you will only find content based on an item type that includes the corresponding attribute (MIME Type or Character Set).

You will find additional information in the Oracle Text documentation on OTN, `http://www.oracle.com/technology/`.

8.3.4 Creating and Dropping Oracle Text Indexes

All the required Oracle Text indexes are created automatically during OracleAS Portal Repository installation. However, if the indexes are subsequently dropped, it may be necessary to re-create them.

Creating and dropping indexes is a very time-consuming and resource-intensive operation, so plan this task during non-business hours. Also, dropping and re-creating Oracle Text indexes may affect the search functionality in your portal as all the indexes must be present and valid for the search feature to operate normally. When an index is dropped, Oracle Text functionality, such as extra search operators, special search result attributes, and so on, are temporarily unavailable even though Oracle Text searching is enabled. This is another good reason for planning this task during non-business hours.

> **Note:** Dropping or creating Oracle Text indexes does not invalidate OracleAS Web Cache. Therefore, search results published automatically and existing search forms, are still displayed until they expire from the cache, or someone edits the search portlet (using the Edit Defaults page).

The following sections describe how to create and drop Oracle Text indexes:

- Creating All Oracle Text Indexes Using ctxcrind.sql
- Creating a Single Oracle Text Index
- Dropping All Oracle Text Indexes Using ctxdrind.sql
- Dropping a Single Oracle Text Index

8.3.4.1 Creating All Oracle Text Indexes Using ctxcrind.sql

You can re-create all the Oracle Text indexes using scripts and packages provided with OracleAS Portal. The primary script for creating the Oracle Text indexes is `ctxcrind.sql` and it is located in the directory *`ORACLE_HOME`*`/portal/admin/plsql/wws`.

When you run the script `ctxcrind.sql` as the OracleAS Portal Repository schema owner:

- All the required Oracle Text indexes and preferences are created. See also, Section 8.3.3, "Oracle Text Indexes".
- If there are existing Oracle Text indexes, all existing preferences and valid indexes are dropped and re-created. Indexes are judged to be valid if:
  - The row in view `user_indexes` for the relevant index has index_status, domidx_status, and domidx_opstatus all set as 'VALID'.
  - The index has an entry in `ctx_user_indexes` with the idx_status set to 'INDEXED'.
- Any indexes that are not present are also created.

This process can take several hours.

To create Oracle Text indexes using the script ctxcrind.sql:

1. Navigate to the directory *`ORACLE_HOME`*`/portal/admin/plsql/wws`.
2. In SQL*Plus, log on using the user name and password for the PORTAL schema.
3. In SQL*Plus, enter this command:

```
ctxcrind.sql
```

If the operation is successful, all the Oracle Text indexes and preferences are created in the OracleAS Portal Repository schema. If it fails, check that your system has met all the requirements. See Section 8.3.2, "Oracle Text Prerequisites".

> **Note:** The time it takes to create the Oracle Text indexes, depends on how many items and page groups exist in your portal.

The script `ctxcrind.sql` makes a call to the procedure:

```
wwv_context.createindex( p_message => l_message );
```

Where `p_message` is an `out` parameter that passes a completion message. The call `wwv_context.createindex()` is in turn equivalent to:

```
wwv_context.drop_prefs;  /* Drop all Oracle Text preferences for the indexes,
except Lexer preferences */
wwv_context.drop_invalid_indexes; /* Drop all invalid indexes */
wwv_context.create_prefs; /* Create all Oracle Text preferences,except Lexer
preferences */
wwv_context.create_missing_indexes(l_indexes);  /* Create missing indexes and
record them in l_indexes */
wwv_context.touch_index(l_indexes); /* Mark all rows for created indexes as
requiring synchronization */
wwv_context.sync;     /* Synchronize indexes */
wwv_context.optimize; /* Optimize indexes */
```

See Also: Appendix G, "Using the wwv_context APIs"

8.3.4.2 Creating a Single Oracle Text Index

If you want to create a specific index, use the procedure `wwv_context.create_index(p_index)`. See also Appendix G.1.3, "create_index".

Use `p_index` to specify which index you want to create. One of:

```
wwv_context.PAGE_TEXT_INDEX
wwv_context.DOC_TEXT_INDEX
wwv_context.PERSPECTIVE_TEXT_INDEX
wwv_context.ITEM_TEXT_INDEX
wwv_context.CATEGORY_TEXT_INDEX
wwv_context.URL_TEXT_INDEX
```

This procedure creates an empty index. Search results cannot be returned from an empty index, so you'll need to populate the index too. See Section 8.3.5.6, "Synchronizing All the Index Content" for information about marking an index for update and synchronizing an index.

8.3.4.3 Dropping All Oracle Text Indexes Using ctxdrind.sql

You can drop all of the Oracle Text indexes and preferences (except for the Lexer preferences), using the script `ctxdrind.sql`. This script is located in the directory *ORACLE_HOME*`/portal/admin/plsql/wws`.

To drop all the Oracle Text indexes using the script ctxdrind.sql:

1. Navigate to the directory *ORACLE_HOME*`/portal/admin/plsql/wws`.
2. In SQL*Plus, log on using the user name and password for the PORTAL schema.
3. In SQL*Plus, enter this command:

   ```
   ctxdrind.sql
   ```

   This script makes a call to:

   ```
   wwv_context.dropindex(p_message  =>l_message);
   ```

   Where `p_message` is an `out` parameter that passes a completion message.

Note: When the Oracle Text indexes are dropped, any views and packages that reference tables on which the indexes were created will become invalid.

These views and packages are automatically validated when they are next accessed. Alternatively, it is possible to validate the views and packages manually.

8.3.4.4 Dropping a Single Oracle Text Index

You may want to drop a specific Oracle Text index. For example, you may want to drop the URL index so that it can be re-created with a different proxy setting, without having to drop and re-create all the other indexes.

To do this, drop the index directly using the command:

```
SQL> drop index <index_name> force;
```

For example, to drop the URL index, enter:

```
SQL> drop index WWSBR_URL_CTX_INDX force;
```

Alternatively, you can drop an index using `wwv_context.drop_index`:

```
SQL>exec wwv_context.drop_index('<index_name>');
```

See also, Appendix G.1.8, "drop_index".

8.3.5 Maintaining Oracle Text Indexes

It is important that you maintain Oracle Text indexes properly as this ensures that portal search results are returned accurately and efficiently. If you are maintaining Oracle Text indexes you'll need to consider index synchronization and optimization:

- **Synchronization —** Updates an Oracle Text index based on a queue.
- **Optimization** — Compacts fragmented rows and removes old data in an Oracle Text index. As an index is synchronized, it grows in such a way as to consume more disk space than necessary and this reduces the efficiency of queries.

Oracle Text gives you full control over how often each index is synchronized and optimized. For more information about synchronization, see:

- Synchronizing Oracle Text Indexes
- Synchronizing an Oracle Text Index On Commit
- Synchronizing All Oracle Text Indexes Manually
- Scheduling Index Synchronization
- Deciding How Often to Synchronize Oracle Text Indexes
- Synchronizing All the Index Content

For more information about optimization, see:

- Optimizing Oracle Text Indexes
- Scheduling Index Optimization
- Choosing the Optimization Interval

8.3.5.1 Synchronizing Oracle Text Indexes

When new content is added to your portal (items, pages, categories, perspectives) it must be indexed before it can be searched. Furthermore, when any row in a table on which the indexes are created are modified, that row is marked as needing synchronization. These are referred to as *pending rows* and they are not returned in search results until the index is synchronized.

You can see which rows are marked pending, using the view `ctx_user_pending`. You can also use the script `textstat.sql` to see the number of rows that need to be synchronized for each index. See also, Section 8.3.8, "Viewing the Status of Oracle Text Indexes".

During installation, Oracle Text indexes are created and synchronized and by default, all indexes are scheduled to synchronize hourly by a job that calls `wwv_context.sync`. If hourly synchronization is not acceptable for your portal you may modify the default synchronization schedule. For example, you can choose to synchronize every five seconds, if it is important to reflect text changes quickly in the index. Alternatively, you can choose to synchronize once a day, for more efficient use of computing resources and a more optimal index. See Section 8.3.5.4, "Scheduling Index Synchronization".

If you are running Oracle Database 10*g* or later, you can specify that Oracle Text indexes synchronize immediately after portal content is added, updated, or deleted, and this can be configured on an index-by-index basis. This feature is not available on databases earlier than Oracle Database 10*g* as earlier versions do not support the `sync` property. Oracle recommend that the page, item, category, and perspective indexes are configured to synchronize *on commit* as this configuration does not impact search performance. You can also configure the document and URL indexes to synchronize *on commit* but as this configuration can impact the speed and efficiency of portal searches, you will need to evaluate its use on a portal-by-portal basis. Table 8–4 summarizes the recommended synchronization schedule for Oracle Text indexes on Oracle Database 10*g*:

Table 8–4 Recommended Synchronization Schedule for Oracle Text Indexes on Oracle Database 10g

Oracle Text Index	Index Synchronization on Oracle Database 10*g*
Page, Item, Category, Perspective	Synchronize immediately after a commit—whenever an associated portal object is added, modified or deleted. See Section 8.3.5.2, "Synchronizing an Oracle Text Index On Commit".
Document, URL	Synchronization scheduled hourly (or some other regular interval) by a job that calls `wwv_context.sync`. See Section 8.3.5.4, "Scheduling Index Synchronization".

The following sections describe your synchronization options:

- Synchronizing an Oracle Text Index On Commit
- Synchronizing All Oracle Text Indexes Manually
- Scheduling Index Synchronization

8.3.5.2 Synchronizing an Oracle Text Index On Commit

Use the procedure `wwv_context.commit_sync()` to specify whether an Oracle Text index synchronizes immediately after data is committed to your portal. This feature is available on Oracle Database 10*g* or later, see also, Appendix G.1.2, "commit_sync".

The commit does not return until the synchronization is complete. Since the synchronization is performed as a separate transaction, there may be a period, usually small, when the data is committed but index changes are not. The operation uses the memory specified with the memory parameter. See also, Appendix G.1.14, "set_sync_memory".

> **Note:** Use `textstat.sql` to determine the current status of this setting. For more information, Section 8.3.8, "Viewing the Status of Oracle Text Indexes"

To specify that an Oracle Text index synchronizes on commit:

Execute `wwv_context.commit_sync` as the OracleAS Portal schema owner from SQL*Plus, using the command:

```
exec wwv_context.commit_sync('<Index_name>', true);
```

To specify that an Oracle Text index synchronizes manually:

Execute `wwv_context.commit_sync` as the OracleAS Portal schema owner from SQL*Plus, using the command:

```
exec wwv_context.commit_sync('<Index_name>', false);
```

To verify the status of on commit synchronization for an Oracle Text index:

Execute `wwv_context.get_commit_sync` as the OracleAS Portal schema owner from SQL*Plus, using the command:

```
set serveroutput on
begin
        dbms_output.put_line(
           case wwv_context.get_commit_sync('<index_name>')
               when true then
                 'Index synchronizes automatically when data commits'
               when false then
                  'Index needs to be synchronized manually'
            end
       );
end;
```

8.3.5.3 Synchronizing All Oracle Text Indexes Manually

Use the procedure `wwv_context.sync()` to synchronize the Oracle Text indexes manually. This procedure indexes all pending rows. See also, Appendix G.1.17, "sync".

With manual synchronization you can also specify memory size and parallel synchronization. See also, Appendix G.1.13, "set_parallel_degree" and Appendix G.1.14, "set_sync_memory".

> **Note:** `wwv_context.sync` ignores any index that synchronizes immediately after data is committed (`wwv_context.commit_sync` is set to true).

To synchronize Oracle Text indexes manually:

Execute this procedure as the OracleAS Portal schema owner from SQL*Plus, using the command:

```
exec wwv_context.sync();
```

Use the following syntax to specify the degree of parallelism used during synchronization:

```
exec wwv_context.set_parallel_degree('<index_name>', <parallel_degree>);
```

For example: `exec wwv_context.set_parallel_degree('WWSBR_CORNER_CTX_INDX', 2);`

Use the following syntax to specify the amount of memory used during synchronization:

```
exec wwv_context.set_sync_memory('<index_name>', <sync_memory);
```

For example: `exec wwv_context.set_parallel_degree('WWSBR_CORNER_CTX_INDX', 12582912);`

This procedure operates across all virtual private portal subscribers.

8.3.5.4 Scheduling Index Synchronization

In most installations, it is desirable to schedule index synchronization to run automatically at regular intervals so that newly added or updated content gets indexed periodically. You can schedule a job using the script `textjsub.sql`. This uses `dbms_job` to call `wwv_context.sync` at regular intervals.

The script takes three parameters and it can also be used to alter or remove a synchronization job:

```
start_time        - a valid date or 'START' or 'STOP'
start_time_fmt    - start time format mask.
                    Ignored if start_time is 'START' or 'STOP'
interval_minutes  - minutes between each run. Ignored if 'STOP'
```

If you set start_time to `START`, the second argument is ignored and the next job is scheduled to run immediately. Subsequent jobs are run after the interval specified.

If you set start_time to `STOP`, the job is removed and other arguments are ignored.

To schedule Oracle Text index synchronization:

Run the script `textjsub.sql`. For example, to schedule index synchronization every 60 minutes, enter:

```
SQL> @textjsub.sql START NOW 60
```

8.3.5.5 Deciding How Often to Synchronize Oracle Text Indexes

The appropriate interval between index synchronization jobs depends on:

- How often new content is added to your portal site.
- Whether it matters that newly added or altered content is not searchable immediately.
- How long is it reasonable to have to wait before added or updated content is searchable.

Depending on your requirements, the synchronization interval could be anything from a few minutes to several days.

It is more efficient to synchronize a larger number of rows on a single occasion than to repeatedly synchronize a smaller number of rows, as the index becomes less fragmented. If an index is less fragmented, then it needs to be optimized less frequently. See Section 8.3.5.7, "Optimizing Oracle Text Indexes" for more information.

However, indexing a larger number of rows at once places a heavier load on the server. Synchronizing more frequently increases the total amount of work but spreads the load on the server. The job only synchronizes the rows that are pending, however, there is always some overhead, however small, in starting up the synchronization job.

8.3.5.6 Synchronizing All the Index Content

You can synchronize *all* the content for a particular Oracle Text index by marking every row in that index as *requiring synchronization*.

For example, when an index is initially created it is empty, so you would need to update the entire index content. This involves performing an update for the column that the index is created on. For every row in the indexed table use the procedure `wwv_context.touch_index(p_index)` to update the column.

After running this procedure, there is an entry in the table `ctx_user_index_pending` for every row in the table upon which the index was created.

Note also that this procedure works across all virtual private portal subscribers.

To synchronize all the content of an index:

Use the procedure `wwv_context.touch_index(p_index)`. Where `p_index` enables you to specify one of these index names:

```
wwv_context.PAGE_TEXT_INDEX
wwv_context.DOC_TEXT_INDEX
wwv_context.PERSPECTIVE_TEXT_INDEX
wwv_context.ITEM_TEXT_INDEX
wwv_context.CATEGPRY_TEXT_INDEX
wwv_context.URL_TEXT_INDEX
```

To synchronize all the content of multiple indexes:

Use the procedure `wwv_context.touch_index(p_indexes)`. Where `p_indexes` enables you to specify a varray of index names to be synchronized (`wwsbr_array`).

8.3.5.7 Optimizing Oracle Text Indexes

Synchronizing Oracle Text indexes causes them to become fragmented. Each Oracle Text index is an inverted index where search terms are listed in a form that is efficient to look up. Each search term references the location of the term.

When new terms are added during synchronization, duplicate terms are not removed, so the index may contain the same term several times. This inflates the size of the index and causes the performance of search queries to deteriorate.

The solution is to optimize the Oracle Text indexes. This process compacts the indexes and (optionally) removes old data.

To optimize all of the Oracle Text indexes:

To optimize all of the Oracle Text indexes, use the procedure `wwv_context.optimize()`. See also, Appendix G.1.12, "optimize".

This procedure takes the following parameters:

```
wwv_context.optimize
(
  p_optlevel in varchar2 default CTX_DDL.OPTLEVEL_FULL, -- FULL, FAST, TOKEN
  p_maxtime in number default null,  -- Maximum time for full optimization, in
minutes
  p_token in varchar2 default null -- Token to optimize (when TOKEN)
);
```

Internally, this procedure calls the Oracle Text procedure `ctx_ddl.optimize_index` for each Oracle Text index and passes these parameters. It performs full index optimization as opposed to *fast* or *token* optimization.

You will find additional information in the Oracle Text documentation on OTN, http://www.oracle.com/technology/.

> **Note:** If no Oracle Text indexes exist, the procedure `wwv_context.optimize` has no effect.

`wwv_context.optimize` only optimizes an Oracle Text index if it is sufficiently fragmented to require optimization. The measure of the fragmentation used is the average number of times a token that appears more than once, is found in the index. If this average is greater than 10, the index is judged to require optimization. The fragmentation query used is as follows:

```
SELECT AVG(COUNT(*)) FROM DR$<index_name>$I
GROUP BY TOKEN_TEXT HAVING COUNT(*) > 1
```

Where `<index_name>` is the name of the index to be measured.

8.3.5.8 Scheduling Index Optimization

In most installations it is desirable to schedule the index optimization process to run automatically at regular intervals. You can schedule a job using the script `optjsub.sql`. This uses `dbms_job` to call `wwv_context.optimize` at regular intervals.

This script `optjsub.sql` takes three parameters and it can also be used to alter or remove an optimization job:

```
start_time       - A valid date or 'START' or 'STOP'
start_time_fmt   - Start time format mask.
                   Ignored if start_time is 'START' or 'STOP'
interval_minutes - Minutes between each run. Ignored if 'STOP'
```

If you set `start_time` to 'START', the second argument is ignored and the next job is scheduled to run immediately. Subsequent jobs are run after the interval specified.

If you set `start_time` to 'STOP', the job is removed and other arguments are ignored.

During OracleAS Portal installation, a job is set up to optimize all of the Oracle Text indexes, every 24 hours.

To schedule Oracle Text index optimization:

Run the script `optjsub.sql`. For example, to schedule index optimization every 60 minutes, enter:

```
SQL> @optjsub.sql START NOW 60
```

This script is located in the directory *ORACLE_HOME*/portal/admin/plsql/wws. If no Oracle Text indexes are present when you run this optimization job, the procedure has no effect.

8.3.5.9 Choosing the Optimization Interval

It is difficult to predict how often Oracle Text indexes need to be optimized as the frequency depends on the amount of content that is being loaded, the type of content being loaded, the synchronization schedule, and many other factors.

However, if you measure the index fragmentation at regular intervals, you can determine how rapidly it is becoming fragmented. Using this information, you can set an appropriate optimization interval.

The procedure wwv_context.optimize only optimizes the index if it is judged to be fragmented. So, other than the minimal overhead of calling the job, it is quite safe to run this job more often than perhaps is required.

During OracleAS Portal installation, a job is set up to optimize all of the Oracle Text indexes, every 24 hours.

8.3.6 Indexing and Searching URL Content

When Oracle Text is enabled in OracleAS Portal, the content of URL attributes attached to items or pages are indexed by default. Once the URL content is indexed, it is searchable. When you enter search criteria for URL attributes, it is this URL content that is searched.

Note: If you do not want portal users to search URL content you can disable the URL index. See Section 8.3.7, "Disabling Document and URL Indexing" for more information.

8.3.6.1 Relative URLs

In OracleAS Portal you can enter a relative URL for a URL attribute. When these URLs display as links on a portal page they are relative to the base HREF that is set in the HTML <head> section for a portal page. The format of the base HREF is:

```
<protocol>://<server>:<port>/portal/pls/<dad>/
```

For example, in the HTML <head> section you might see:

```
<base href="http://myserver.abc.com/portal/pls/portal/">
```

In this example:

- The relative URL /help/index.html is resolved by the browser to:

  http://myserver.abc.com/help/index.html

- The relative URL!PORTAL.mypackage.proc (with no leading /) is resolved by the browser to:

  http://myserver.abc.com/portal/pls/portal/!PORTAL.mypackage.proc

The base HREF on a page is dependent on the URL used to request the page. As it is possible to use more than one URL to access the page, the base HREF reflects the URL used to access the page.

Oracle Text Base URL Setting

When indexing URL content, Oracle Text needs to know how to resolve relative URLs into fully qualified absolute URLs. As Oracle Text does not have the context of an initial request from which to determine the correct base HREF, you must specify the base HREF that is used. You set this option, by specifying the **Oracle Text Base URL** property on the **Global Settings: Search** page. See Section 8.2.2.3, "Setting a Base URL for Oracle Text" for details.

During OracleAS Portal installation, this option is set automatically.

The format of the Oracle Text Base URL is:

```
<protocol>://<server>:<port>/portal/pls/<dad>/
```

For example: `http://myserver.abc.com/portal/pls/portal/`

> **Note:** Do not specify an Oracle Text Base URL beginning with `https`, as HTTPS URLs are not indexed by Oracle Text. If you do this, no relative URLs are indexed.

If you change the Oracle Text Base URL, it does not take effect immediately. When a URL is edited, it is marked as requiring synchronization and Oracle Text will use the new preference the next time the index is synchronized. If you want to force all URLs to immediately use a new Oracle Text Base URL value, you can mark the entire content of the URL Index as *requiring synchronization*, using the procedure:

```
SQL> wwv_context.touch_index(wwv_context.URL_TEXT_INDEX);
```

This procedure acts across all subscribers. In a single virtual private portal subscriber, this is equivalent to:

```
SQL> update wwsbr_url$ set absolute_url = null;
...
SQL> commit;
```

8.3.6.2 Unsupported URLs

Oracle Text cannot index URLs that use these protocols:

- https
- javascript

If a URL item specifies one of these protocols it is not indexed. You will not see a corresponding error in the Oracle Text error logs.

8.3.6.3 Supported URLs

Oracle Text can index URLs that use these protocols:

- **http**
- **file** - File URLs must be accessible from the database server.
- **ftp** - FTP URLs must point to locations that do not require authentication as Oracle Text is not able to authenticate — even as an anonymous user.

8.3.6.4 URL Index Proxy Settings

When indexing URL content, Oracle Text can use proxy servers to access URLs. This may be necessary when OracleAS Portal lies behind a firewall and URLs point to content beyond this firewall. As indexing takes place from the OracleAS Portal Repository server, it is the proxy settings required on this computer that are important.

The URL index uses the same proxy settings that are used globally for OracleAS Portal. These are set on the Proxy Settings page, available from the **Services** portlet. See Section 8.2.2.4, "Configuring Proxy Settings for Oracle Text" for details.

The proxy settings are used when Oracle Text indexes are created. So, if you change the proxy settings the indexes must be re-created. If you need to drop all your indexes and re-create them, use the scripts `ctxdrind.sql` (drop indexes) and `ctxcrind.sql` (create indexes). See Section 8.3.4, "Creating and Dropping Oracle Text Indexes" for more information:

```
SQL> @ctxdrind.sql
...
SQL> @ctxcrind.sql
...
```

These scripts drop and re-create *all* of the indexes and this can take a long time if your indexes are large. Alternatively, you can drop and re-create the Oracle Text preferences and URL index only:

```
begin
   -- Drop and re-create the Oracle Text preferences
   -- to pick up the new proxy settings.
   wwv_context.drop_prefs();
   wwv_context.create_prefs();
end;
/
-- Check that the proxy settings used by the index are correct
select prv_attribute attribute, prv_value value
  from ctx_user_preference_values
  where prv_attribute in ('TIMEOUT','HTTP_PROXY','NO_PROXY')
/

begin
   -- Drop and re-create the URL index
   wwv_context.drop_index(wwv_context.URL_TEXT_INDEX);
   wwv_context.create_index(wwv_context.URL_TEXT_INDEX);

   -- Mark all of the rows for the index as pending
   wwv_context.touch_index(wwv_context.URL_TEXT_INDEX);

   -- Syncronize and optimize
   wwv_context.sync();
   wwv_context.optimize();
end;
/
```

8.3.7 Disabling Document and URL Indexing

By default, the content of files uploaded to the OracleAS Portal Repository and the content referenced in URL items or custom URL attributes is indexed. This allows users to search and find terms in document and URL content and for most cases, this is desirable.

When portal users do not need to search within file and URL content you may wish to disable the Document and URL indexes. In this case, searching is limited to item, page, category, and perspective metadata, including Title, Author, Keywords, Description, Update Date and all custom Text, Boolean and Date attributes. Metadata-only searching is more efficient and therefore faster than searches that include file and URL content.

Note: If the OracleAS Portal Repository is installed into a database that does not have a functional AUTO_FILTER, document and URL searching is automatically disabled as this functionality does not work without the AUTO_FILTER.

Use the following procedures to specify whether the document and URL indexes are required:

- set_use_doc_index
- set_use_url_index

Both procedures belong to the package `wwv_context`. For more detail, see Appendix G, "Using the wwv_context APIs".

If you disable Document and URL indexes, the script `ctxcrind.sql` (which normally creates missing Oracle Text indexes) removes existing Document/URL indexes as they are no longer required. If you do not remove Document/URL indexes, the indexes are still updated when the synchronization and optimization jobs are run. Therefore, it is more efficient to remove the unused indexes by running `ctxcrind.sql`. See Section 8.3.4, "Creating and Dropping Oracle Text Indexes".

Whenever you make changes to these Document and URL index settings, the appearance and behavior of search portlets in OracleAS Portal are affected. If portlets are being cached, such changes might not appear immediately. Therefore, you should clear the portal cache manually, after making any index changes. See Section 5.8.3.4, "Clearing the Cache for a Particular Portal Object".

For example, when you disable the Document index, search portlets display fewer search operators for file- based attributes, that is, only *Match All Within File Name* and *Match Any Within File Name*. Similarly, if the URL index is disabled, the only operators available for URL-based attributes are *Match All Within URL* and *Match Any Within URL*. Other search operators (such as *Content Contains All*) are not displayed as file and URL content is not searchable when these indexes are disabled.

When you disable the Document index, Themes, Gists and View as HTML features are no longer available, so you must disable *Themes and Gists* on the **Global Settings: Search** page. See Section 8.2.2.2, "Setting Oracle Text Search Result Options" for details.

To enable or disable Document and URL indexes:

Use the following procedures:

```
-- To enable the document index
execute wwv_context.set_use_doc_index(true);

-- To disable the document index
execute wwv_context.set_use_doc_index(false);

-- To enable the URL index
execute wwv_context.set_use_url_index(true);
```

```
-- To disable the URL index
execute wwv_context.set_use_url_index(false);
```

8.3.8 Viewing the Status of Oracle Text Indexes

You can determine the status of Oracle Text indexes from several tables and views accessible from the portal schema.

To view a status report for Oracle Text indexes, run the script `textstat.sql` as the portal schema owner:

```
SQL> @textstat.sql
```

This script is located in the directory *ORACLE_HOME*`/portal/admin/plsql/wws`. Here is an example of the information that is generated by this script:

```
SQL> @textstat
Portal Text Indexes:

INDEX_NAME            STATUS   DOMIDX_STATUS DOMIDX_OPSTATUS IDX_STATUS
--------------------- -------- ------------- --------------- ------------
WWSBR_CORNER_CTX_INDX VALID    VALID         VALID           INDEXED
WWSBR_DOC_CTX_INDX    VALID    VALID         VALID           INDEXED
WWSBR_PERSP_CTX_INDX  VALID    VALID         VALID           INDEXED
WWSBR_THING_CTX_INDX  VALID    VALID         VALID           INDEXED
WWSBR_TOPIC_CTX_INDX  VALID    VALID         VALID           INDEXED
WWSBR_URL_CTX_INDX    VALID    VALID         VALID           INDEXED

Document and URL index preferences:
Document Index: true  - index will be used if valid
URL Index:      true  - index will be used if valid

Indexes with rows waiting to be indexed:

Index                     Rows to Index
------------------------  -------------
WWSBR_CORNER_CTX_INDX        2677

PL/SQL procedure successfully completed.

Scheduled Text Jobs:

LAST_DATE LAST_SEC  NEXT_DATE NEXT_SEC  B  FAILURES INTERVAL            WHAT
--------- --------- --------- ---------- - ------ -------------------------
25-AUG-05 04:57:32  26-AUG-05 04:57:32  N   0     SYSDATE + 24/24 wwsbr_stats.gather_stale;
25-AUG-05 04:57:32  26-AUG-05 04:57:32  N   0     SYSDATE + 1440/(24*60) wwv_
context.optimize(CTX_DDL.OPTLEVEL_FULL,1440,null);
25-AUG-05 06:59:30  25-AUG-05 07:59:30  N   0     SYSDATE + 60/(24*60)  wwv_context.sync;

Running Text Jobs:
no rows selected

Indexes sync on commit setting:
Item Index:        true    - Index will sync automatically when data commits
Page Index:        true    - Index will sync automatically when data commits
Document Index:    false   - Index needs to be syncronized manually
Category Index:    true    - Index will sync automatically when data commits
Perspective Index: true    - Index will sync automatically when data commits
URL Index:         false   - Index needs to be syncronized manually

SQL>
```

From this script you can view the following information:

- **Portal Text Index Status** - Shows whether all of the Oracle Text indexes exist and their current status. All working, valid indexes display `VALID` for the first three status columns and `INDEXED` for the final column as shown in this example. See also, Section 8.3.3.1, "Oracle Text Index Overview".
- **Document and URL Index Status** - Indicates whether the Document and URL indexes are enabled (true) or disabled (false). See also, Section 8.3.7, "Disabling Document and URL Indexing".
- **Number of Pending Rows Per Index** - Lists any indexes that are waiting to be indexed. An entry is listed for every index that has rows waiting to be indexed, or are pending. The number of pending rows is also shown. See also, Section 8.3.5.1, "Synchronizing Oracle Text Indexes".
- **Scheduled Oracle Text Job Details** - Lists any jobs that are scheduled for Oracle Text index maintenance. The report shows the last date and time that the job was run and the next date when the job is due to be run. The column labeled **B** shows whether the job is broken or not; if the job is marked **Y** it is broken and does not run. The **Interval** column indicates the next time that a job will run and finally, the **What** column indicates the procedure that will be run for each job. See also, Section 8.3.5.4, "Scheduling Index Synchronization".
- **Active Oracle Text Job Details** - Details any jobs that were running when the `textstat.sql` report ran.
- **Indexes Sync On Commit Setting** - On Oracle Database 10*g* or later, this section shows which indexes are configured to synchronize immediately after data commits to you portal and which ones need to be synchronized manually using `wwv_context.sync` (manually or using a job). See also, Section 8.3.5.2, "Synchronizing an Oracle Text Index On Commit".

  On earlier database versions, the following information is displayed:

```
Indexes sync on commit setting:
...Not available for this database version, available from 10g onwards.
```

8.3.9 Monitoring Oracle Text Indexing Operations

Oracle Text logs information to a file when indexes are created and populated. This enables you to monitor the progress of indexing operations, keep track of indexes, and troubleshoot any problems that may arise.

8.3.9.1 Using start_log to Monitor Index Operations

You can use the `ctx_output.start_log (filename)` command to log output from the indexing process. In the subsequent example, the log file is named `textindex.log`:

```
ctx_output.start_log('textindex.log');
ctx_output.add_event(ctx_output.event_index_print_rowid);
...
-- Create or syncronize the indexes
...
ctx_output.end_log;
```

You can determine the location of the log file using the `LOG_DIRECTORY` parameter in `ctx_adm.set_parameter`, for example, `/tmp`. Once the directory is set, all subsequent Oracle Text logs output to this directory:

```
ctxsys.ctx_adm.set_parameter('LOG_DIRECTORY', '/tmp');
```

8.3.9.2 Using logcrind.sql to Monitor Index Creation

You can use the script `logcrind.sql` (instead of `ctxcrind.sql`) to create the Oracle Text indexes with logging enabled. The script takes one parameter which is the name of the log file, for example:

```
SQL> @logcrind.sql textindex.log
```

This script sets the `LOG_DIRECTORY` to be the same as the database `udump` directory, as specified by the `user_dump_dest` initialization parameter.

The `add_event` call (used in the preceding example) is also used in the script `logcrind.sql` and this outputs the rowid of every row indexed to the log. This logging allows indexing operations to be tracked and also indicates whether the indexing of each row is successful or not.

Here is a sample from an Oracle Text indexing log:

```
13:53:27 05/06/03 begin logging
13:53:27 05/06/03 event
13:53:42 05/06/03 log
13:53:42 05/06/03 event
13:53:48 05/06/03 Creating Oracle index "MYPORTAL"."DR$WWSBR_CORNER_CTX_INDX$X"
13:53:48 05/06/03 Oracle index "MYPORTAL"."DR$WWSBR_CORNER_CTX_INDX$X" created
13:53:49 05/06/03 Creating Oracle index "MYPORTAL"."DR$WWSBR_DOC_CTX_INDX$X"
13:53:49 05/06/03 Oracle index "MYPORTAL"."DR$WWSBR_DOC_CTX_INDX$X" created
13:53:49 05/06/03 Creating Oracle index "MYPORTAL"."DR$WWSBR_PERSP_CTX_INDX$X"
13:53:49 05/06/03 Oracle index "MYPORTAL"."DR$WWSBR_PERSP_CTX_INDX$X" created
13:53:50 05/06/03 Creating Oracle index "MYPORTAL"."DR$WWSBR_THING_CTX_INDX$X"
13:53:50 05/06/03 Oracle index "MYPORTAL"."DR$WWSBR_THING_CTX_INDX$X" created
13:53:51 05/06/03 Creating Oracle index "MYPORTAL"."DR$WWSBR_TOPIC_CTX_INDX$X"
13:53:51 05/06/03 Oracle index "MYPORTAL"."DR$WWSBR_TOPIC_CTX_INDX$X" created
13:53:51 05/06/03 Creating Oracle index "MYPORTAL"."DR$WWSBR_URL_CTX_INDX$X"
13:53:51 05/06/03 Oracle index "MYPORTAL"."DR$WWSBR_URL_CTX_INDX$X" created
13:54:16 05/06/03 sync index: MYPORTAL.WWSBR_CORNER_CTX_INDX
13:54:17 05/06/03 Begin document indexing
13:54:17 05/06/03 INDEXING ROWID AAAUUcAAJAAAlhMAAA
13:54:17 05/06/03 INDEXING ROWID AAAUUcAAJAAAlhMAAI
..
13:54:18 05/06/03 INDEXING ROWID AAAUUcAAJAAAlhQAAk
13:54:18 05/06/03 Errors reading documents: 0
13:54:18 05/06/03 Index data for 159 documents to be written to database
13:54:18 05/06/03    memory use: 225971
13:54:18 05/06/03 Begin sorting the inverted list.
13:54:18 05/06/03 End sorting the inverted list.
13:54:18 05/06/03 Writing index data to database.
13:54:18 05/06/03    index data written to database.
13:54:18 05/06/03 End of document indexing. 159 documents indexed.
```

8.3.10 Viewing Indexing Errors

Any errors that occur when an index is created or synchronized are logged in the view `CTX_USER_INDEX_ERRORS`. You can see details for these errors, using the command:

```
SQL> desc ctx_user_index_errors;
 Name                    Null?    Type
 ----------------------- -------- ---------------
 ERR_INDEX_NAME          NOT NULL VARCHAR2(30)
 ERR_TIMESTAMP                    DATE
 ERR_TEXTKEY                      VARCHAR2(18)
 ERR_TEXT                         VARCHAR2(4000)
```

```
SQL>
```

This view gives the index name, the rowid (ERR_TEXTKEY column) corresponding to the row in the indexed table, and an error message that indicates the cause of the failure. Furthermore, the error log file indicates the rowid for the row in the table that is being indexed and a success or failure message.

Typically, you do not see errors for the item (WWSBR_THING_CTX_INDX), page (WWSBR_CORNER_CTX_INDX), category (WWSBR_TOPIC_CTX_INDX) or the perspective (WWSBR_PERSP_CTX_INDX) indexes as these index content that is produced by OracleAS Portal and this content is easy to index. It is more common to see indexing errors for document and URL content.

For the Document index, the content may have to be filtered to turn a binary document into plain text for indexing. There are a number of reasons this may fail. For example, the document format may not be supported by AUTO_FILTER, the Oracle Text filter. See also, Section 8.3.3.7, "Maximizing AUTO_FILTER Performance".

For the URL index, the URL content has to be fetched and this could fail for a number of reasons. For example, the URL may indicate a location that is not accessible as the OracleAS Portal server is behind a firewall and the proxy settings are not set correctly. Or, maybe the URL is incorrect, or perhaps the site that is being access is down.

In addition, URL content is filtered and this may produce errors. For example, all URL attributes are presumed to be plain text, so you'll see an error for any URL attribute that is not plain text.

8.3.11 Translating Indexing Errors to Objects in OracleAS Portal

The indexing errors shown in the view CTX_USER_INDEX_ERRORS or the Oracle Text indexing logs, show the rowid of the row in the table being indexed when the error occurred. You can use this information to determine which row is causing an indexing problem and you can also determine exactly which portal item or page this row corresponds to.

The following sections describe some queries that you may find useful if you need to determine the cause of an indexing issues:

- Item Indexing Errors
- Page Indexing Errors
- Category Index Errors
- Perspective Indexing Errors
- Document Index Errors
- URL Index Errors

8.3.11.1 Item Indexing Errors

The rowid gives the row in the items table that is causing problems. You can use a direct query to find out more information about that row. For example:

```
select i.name, i.title,          -- item title
       p.name page_name,         -- page name
       p.title page_title,       -- page display name
       pg.name page_group,       -- page group name
       sl.title page_group_title -- page group display name (default language)
  from wwv_things i,
       wwpob_page$ p,
       wwpob_item$ pi,
```

```
       wwsbr_sites$ pg,
       wwsbr_site_languages$ sl
 where i.masterthingid = pi.master_thing_id
   and i.siteid = pi.site_id
   and pi.page_id = p.id
   and sl.siteid = pg.id
   and sl.language = pg.defaultlanguage
   and pi.page_site_id = p.siteid
   and pg.id = i.siteid
   and i.rowid = 'AAAOwMAAJAAAWISAAF
```

8.3.11.2 Page Indexing Errors

The rowid gives the row in the pages table. You can use a direct query to find out more information about the page being indexed. For example:

```
select p.name page_name,
       p.title page_title,
       pg.name page_group,
       sl.title page_group_title
  from wwpob_page$ p,
       wwsbr_sites$ pg,
       wwsbr_site_languages$ sl
 where sl.siteid = pg.id
   and sl.language = pg.defaultlanguage
   and pg.id = p.siteid
   and p.rowid = 'AAAOv/AAJAAAaSSAAB'
```

8.3.11.3 Category Index Errors

You can use a direct query against the category table to determine faulty categories. You can also use a join to show the page group. This query shows the category name and display name, and the page group name and display name.

```
select c.title, c.name, pg.name, sl.title
  from wwv_topics c,
       wwsbr_sites$ pg,
       wwsbr_site_languages$ sl
 where sl.siteid = pg.id
   and sl.language = pg.defaultlanguage
   and pg.id = c.siteid
   and rowid='AAAOv/AAJAAAaSSAAB'
```

8.3.11.4 Perspective Indexing Errors

These are similar to categories. If you use a direct query against the perspective table it shows the faulty perspectives. You can also use a join to show the page group.

```
select p.title, p.name, pg.name, sl.title
  from wwv_perspectives p,
       wwsbr_sites$ pg,
       wwsbr_site_languages$ sl
 where sl.siteid = pg.id
   and sl.language = pg.defaultlanguage
   and pg.id = p.siteid
   and p.rowid = 'AAAOv/AAJAAAaSSAAB'
```

8.3.11.5 Document Index Errors

You are more likely to see errors with the Document index. In this case the index is on the table where the documents are actually stored. Therefore, you have to join back to the item table to determine the associated item.

The following query gives the document file name and item's Name and Display Name that a document query is associated with:

```
select d.filename, i.name, i.title  from wwv_things i,
       wwdoc_document$ d,
       wwv_docinfo di
 where
    d.name = di.name(+)
    and di.thingid = i.id(+)
    and di.masterthingid = i.masterthingid(+)
    and di.siteid = i.siteid(+)
    and d.rowid = 'AAAOYyAAJAAAWAaAAF'
```

Note that not all documents are necessarily associated with items, in which case you need to modify the query to join in a similar way to the page table.

8.3.11.6 URL Index Errors

Like the Document index, you have to join back to the item table to determine the associated item.

The following query shows the URL, and item Name and Display Name.

```
select u.url, u.absolute_url, i.name, i.title
  from wwv_things i,
       wwsbr_url$ u
 where u.object_id = i.id
   and u.object_siteid = i.siteid
   and u.object_type = 'ITEM'
   and u.rowid = 'AAAOYyAAKAAAWAaAAB'
```

Note that a URL may not be attached to an item, it may be attached to a page, in which case you need to modify the query to join in a similar way to the page table.

8.3.12 Common Indexing Errors

Some common indexing errors are described in the following sections:

- Common Document Indexing Errors
- Common URL Indexing Errors

8.3.12.1 Common Document Indexing Errors

Typically, document indexing errors are in the format:

```
DRG-11207: user filter command exited with status n
```

The actual exit status indicates the cause of the problem. For a description of common exit status values and their meanings, log on to Oracle Metalink, at `http://metalink.oracle.com` and read the article *Troubleshooting DRG-11207 errors*. This article has **DocId 210319.1**.

8.3.12.2 Common URL Indexing Errors

Here are some common URL indexing errors. The list is not exhaustive but it highlights some of the more common errors you may see:

`DRG-11604 URL store: access to %(1)s is denied`
Access to the document is denied to the indexing user agent. The crawler is not capable of authenticating or managing cookies returned by the site. Check that the URL can be accessed. If it is protected, it may not be possible to index the content.

`DRG-11609 URL store: unable to open local file specified by %(1)s`
`DRG-11610 URL store: unable to read local file specified by %(1)s`
These occur for file:// URLs where the file indicated cannot be opened or read. Remember that the file needs to be accessible from the computer on which the OracleAS Portal Repository database is running. Check that the file exists and that it is accessible from the database computer as the database user.

`DRG-11611 URL store: unknown protocol specified in %1)s`
The protocol specified in the URL is not one that the Oracle Text user agent recognizes. This can happen if no protocol is specified. A common cause of this problem is that a relative URL is specified but the Oracle Text Base URL option is not set to fully qualify the URL. Also, Oracle Text can only index http, file and ftp URLs. Look at the URL that has failed and make sure that it is in a supported fully qualified format, including a valid protocol. See also, Section 8.3.6, "Indexing and Searching URL Content"

`DRG-11612 URL store: unknown host specified in %(1)s`
The URL specified a host in the URL that cannot be resolved from the OracleAS Portal repository database server. It may be that a firewall lies between the OracleAS Portal repository server and the location specified by the URL. In this case it might be necessary to use a proxy server to access the URL. Check that the URL is correct and that the host is accessible from the OracleAS Portal database server. Also check that the OracleAS Portal proxy settings are correct and that the index is using the proxy settings. See also, Section 8.2.2.4, "Configuring Proxy Settings for Oracle Text".

`DRG-11613 URL store: connection refused to host specified by %(1)s`
This means that the host specified in the URL was resolved but the HTTP request was refused. Check that the URL is correct and that it is accessible.

`DRG-11614 URL store: communication with host specified in %(1)s timed out`
The request timed out. Check that the URL is correct and accessible.

`DRG-11616 URL store: too many redirections trying to access %(1)s`
When accessed, a URL can cause a redirect to another URL. This in turn can cause a redirect, and so on. If a large number of redirects occur, this error is displayed. This can occur if a redirection loop is found.

`DRG-11622 URL store: unknown HTTP error getting %(1)s`
An HTTP error that is not explicitly handled by Oracle Text has occurred. The HTTP error is reported in the error message.

8.3.13 Handling Indexing Hangs or Crashes

If for any reason a document or URL cannot be indexed, an error is logged. This situation should not prevent the indexing operation completing normally. However, any content that fails to be indexed is not searchable.

Sometimes an indexing operation can fail catastrophically in which case, the index operation is terminated before the indexes are properly populated. In most cases, such problems should be reported to Oracle Support Services. However, in some instances you may be able to work around the problem temporarily by excluding the content

that is causing a failure. See Section 8.3.13.2, "Preventing Indexes From Hanging and Crashing" for more information.

Rarely, an indexing operation causes a disastrous failure, that is, the server process performing the indexing is terminated. When this happens, this message is displayed in the client running the indexing operation:

```
ORA-03113 End of file on communication channel
```

> **Note:** If you are unsure whether an indexing operation completed successfully, repeat the operation from SQL Plus where *end of file* errors are clearly reported.

If the server process is terminated, the event should be recorded in the database logs. Use the database alert log to determine the location of any trace files that are written. The trace files may indicate errors such as ORA-0600 or ORA-7445. For example, this trace file shows errors that occurred whilst creating Oracle Text indexes using the script `logcrind.sql`:

```
ksedmp: internal or fatal error
ORA-7445: exception encountered: core dump [drsfdatam()+308] [SIGSEGV]
[Address not mapped to object] [0x0] [
] []
Current SQL statement for this session:
declare
l_dump_dest varchar2(512);
p_logfile varchar2(100) := 'sync_2012.log';
begin
dbms_output.enable(10000);
select value into l_dump_dest from v$parameter
where name = 'user_dump_dest';
ctxsys.ctx_adm.set_parameter('LOG_DIRECTORY',l_dump_dest);
ctx_output.start_log(p_logfile);
ctx_output.add_event(ctx_output.event_index_print_rowid);
dbms_output.put_line('Log file is: '||ctx_output.logfilename);
wwv_context.sync();
ctx_output.end_log;
end;
----- PL/SQL Call Stack -----
object line object
handle number name
8198f83c 244 package body CTXSYS.DRIDISP
8198f83c 377 package body CTXSYS.DRIDISP
8198f83c 334 package body CTXSYS.DRIDISP
8178acc8 403 package body CTXSYS.DRIDML
827124b0 2033 package body CTXSYS.DRIDDL
827124b0 2090 package body CTXSYS.DRIDDL
817ea0f0 1324 package body CTXSYS.CTX_DDL
8185a488 828 package body TOOLS.WWV_CONTEXT
82d83ed8 18 anonymous block
----- Call Stack Trace -----
```

8.3.13.1 Identifying Whether an Index Operation is Hanging

The easiest way to determine if an indexing operation is hanging is to run the indexing operation with Oracle Text logging enabled. See Section 8.3.9, "Monitoring Oracle Text Indexing Operations".

With logging enabled, the rowid of each row is recorded when it is indexed and you can see when an indexing operation hangs on the same row for a prolonged period. It may be normal for some rows to take a few minutes to process but if an operation takes much longer than expected, this may indicate a problem.

In general, the view CTX_USER_INDEX_ERRORS is not very useful if you are trying to find out why an indexing process is hanging or crashing. This is because information is only visible in this view after it is committed and a commit does not occur whilst an indexing operation is hanging. In fact, a commit may not occur at all if the operation crashes.

Operations such as URL indexing and document filtering can take quite a long time to process. Both of these operations are subject to timeout mechanisms to avoid lengthening this process even further:

- **URL indexing timeout -**The default timeout for fetching URL content is 30 seconds. If URL content is not retrieved within 30 seconds, the attempt is abandoned, a failure error is reported in the view CTX_USER_INDEX_ERRORS and the indexing process continues to the next row. In most cases, 30 seconds is sufficient time to fetch URL content. However, once the content is retrieved it must be indexed, so the total time can be slightly more than the URL timeout value.
- **Document filtering timeout -** The timeout for document filtering operations is not a *hard* timeout limit. The timeout setting, which by default is 120 seconds, is the time that is waited while no output is produced by the AUTO_FILTER. If the timeout is exceeded the current filtering operation is terminated, the content for the current document is not indexed, and the indexing process proceeds to the next document. If the AUTO_FILTER output file is still growing after 120 seconds, the filtering operation is allowed to continue.

These timeout mechanisms help to avoid problems with URL and document indexing, two areas where issues are likely to arise. However, you may still encounter situations where an indexing operation hangs indefinitely.

8.3.13.2 Preventing Indexes From Hanging and Crashing

If certain content is causing indexing operations to fail, you can exclude the content from the indexing process. First, you must identify the row that is causing the problem. This section describes how to do this and the additional steps required to exclude such content.

Step 1 Identify the rowid Causing Indexing Problems

You can do this using the Oracle Text logging facility, with `print rowid event` enabled. If you look at the generated log file you can determine the rowid (of the row being processed) when failure occurred. In most cases it is this rowid that is causing indexing problems.

However, in some cases the actual rowid being processed may not be written to the log file when the failure occurs. In this case you must determine the *next* rowid:

- If the entire table is being synchronized, for example, when an index is first created, the rowid is the next rowid from the table. To determine the rowid, select from the table without an order by clause.
- When only a few pending rows are being updated, look at the view `ctx_user_pending` to determine the next rowid.

When you have identified which row is causing your indexing problems, you should verify that it is the correct row. You do this by reproducing the failure while synchronizing that row only.

If the Oracle Text indexes do not exist, create the indexes (but do not populate them) using these command:

```
SQL> exec wwv_context.drop_prefs;
PL/SQL procedure successfully completed.
SQL> exec wwv_context.create_prefs;
PL/SQL procedure successfully completed.
SQL> declare
  2      l_indexes wwsbr_array;
  3  begin
  4      wwv_context.create_missing_indexes(l_indexes);
  5  end;
  6  /
PL/SQL procedure successfully completed.
SQL>
```

This creates all of the indexes, with no rows pending.

Step 2 Mark the Problem rowid As Pending

The next step is to mark the row suspected of causing indexing problems as pending. The column you need to update depends on which index you are updating. The names of these columns are indicated in the subsequent examples. You must replace the rowid given in these examples, with the rowid you wish to verify:

URL index (WWSBR_URL_CTX_INDX) The `absolute_url` column is populated by a trigger, so set it here to null:

```
update wwsbr_url$ set absolute_url=null where rowid = 'AAAOwQAAJAAAU0+AAL';
```

Document index (WWSBR_DOC_CTX_INDX) Update the `blob_content` column, but preserve the original `blob_content` value:

```
update wwdoc_document$ set blob_content = blob_content where rowid =
'AAAOYyAAJAAAWAaAAF'
```

Item index (WWSBR_THING_CTX_INDX) This index uses a user datastore created on the `ctxtxt` column. The value of this column is irrelevant and in OracleAS Portal is always 1.

```
update wwv_things set ctxtxt = '1' where rowid = 'AAAOwMAAJAAAU0eAAB'
```

Page index (WWSBR_FOLDER_CTX_INDX) Similar to the item index.

```
update wwpob_page$ set ctxtxt = 1 where rowid = 'AAAOwMAAJAAAWITAAA'
```

Category index (WWSBR_TOPIC_CTX_INDX) Similar to the item index.

```
update wwv_topics set ctxtxt = 1 where rowid = 'AAAOwMAAJAAAWITAAA'
```

Perspective index (WWSBR_PERSP_CTX_INDX) Similar to the item index.

```
update wwv_perspectives set ctxtxt = 1 where rowid = 'AAAOwMAAJAAAWITAAA'
```

If you have a site with several subscribers installed then you may need to switch subscriber before you can see the row that you are interested in. To change subscribers, use the following procedure to set the session context for a lightweight user:

```
wwctx_api.set_context
(
       p_user_name     IN varchar2,
       p_password      IN varchar2 default null,
       p_company       IN varchar2 default null
```

```
);
```

`wwctx_api` is a public PL/SQL API package. For more information, refer to the *OracleAS Portal PL/SQL API Reference* available on OTN at `http://www.oracle.com/technology/products/ias/portal`.

After the column update, the suspect row is placed in the pending queue.

Step 3 Synchronize the Index

Now you can synchronize the index and see if the same problem occurs, using the command:

```
SQL> exec wwv_context.sync();
```

This command synchronizes the suspect row only as it is the only row in the pending queue. The row can be updated again to repeat the test. See also, Appendix G.1.17, "sync".

Step 4 Exclude the Content Causing Problems

You can prevent the indexing operation from hanging or crashing in the future, by modifying, or even removing the row causing indexing problems. For example, if it is a document, you can edit the associated item in OracleAS Portal and remove the document.

> **Note:** Contact Oracle Support Services if your system hangs or crashes during indexing operations. If you can provide specific detail relating to the content causing the problem, it will help them to reproduce the problem more readily.

8.3.14 Troubleshooting Oracle Text Installation Issues

If you are experiencing issues with Oracle Text, use the `TEXTTEST` utility to check that Oracle Text functionality is installed and setup correctly. See Appendix H, "Using TEXTTEST to Check Oracle Text Installation" for details.

8.4 Oracle Ultra Search

This section introduces Oracle Ultra Search and the sample Ultra Search portlet. Specific topics in this section include:

- Oracle Ultra Search Overview
- Sample Oracle Ultra Search Portlet

8.4.1 Oracle Ultra Search Overview

Oracle Ultra Search is built on Oracle Database and Oracle Text technology and provides uniform search-and-locate capabilities over multiple repositories: Oracle Databases, other ODBC compliant databases, IMAP mail servers, HTML documents served up by a Web server, files on disk, and more.

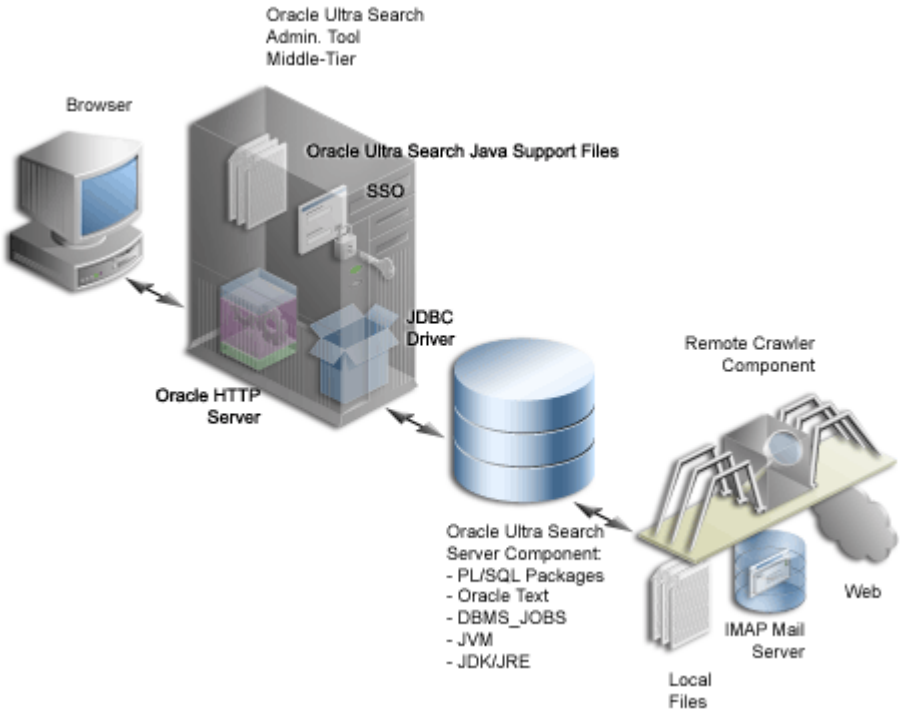
Oracle Ultra Search uses a *crawler* to collect documents. You can schedule the crawler to suit the Web sites that you want to search. The documents stay in their own repositories, and the crawled information is used to build an index that stays within your firewall in a designated Oracle Database. Oracle Ultra Search also provides APIs for building content management solutions.

In addition, Oracle Ultra Search offers the following:

- A complete text query language for text search inside the database
- Full integration with the Oracle Database server and the SQL query language
- Advanced features like concept searching and theme analysis
- Attribute mapping to facilitate attribute search across disparate repositories
- Indexing of all popular file formats (150+)
- Full globalization, including support for Chinese, Japanese and Korean (CJK), and Unicode

Figure 8–11 shows an overview of the Oracle Ultra Search architecture:

Figure 8–11 Oracle Ultra Search Architecture



You will find additional information on OTN, http://www.oracle.com/technology/, in:

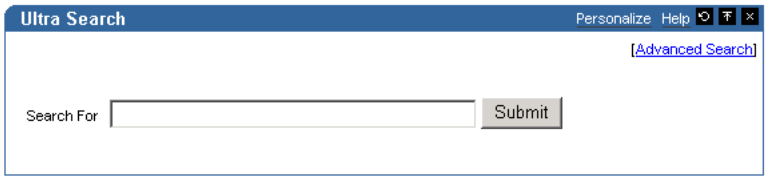
- *Oracle Ultra Search User's Guide*
- Oracle Ultra Search papers and presentations.

Oracle Ultra Search is integrated with OracleAS Portal so that you can add powerful multi repository search facilities to portal pages. It also has the capability to crawl OracleAS Portal's own repository and search *public* content.

8.4.1.1 About the Oracle Ultra Search Sample Query Applications

Oracle Ultra Search includes fully functional sample query applications to query and display search results. The query applications are written as J2EE-compliant Web applications. The sample query applications also include the Ultra Search portlet, shown in Figure 8–12.

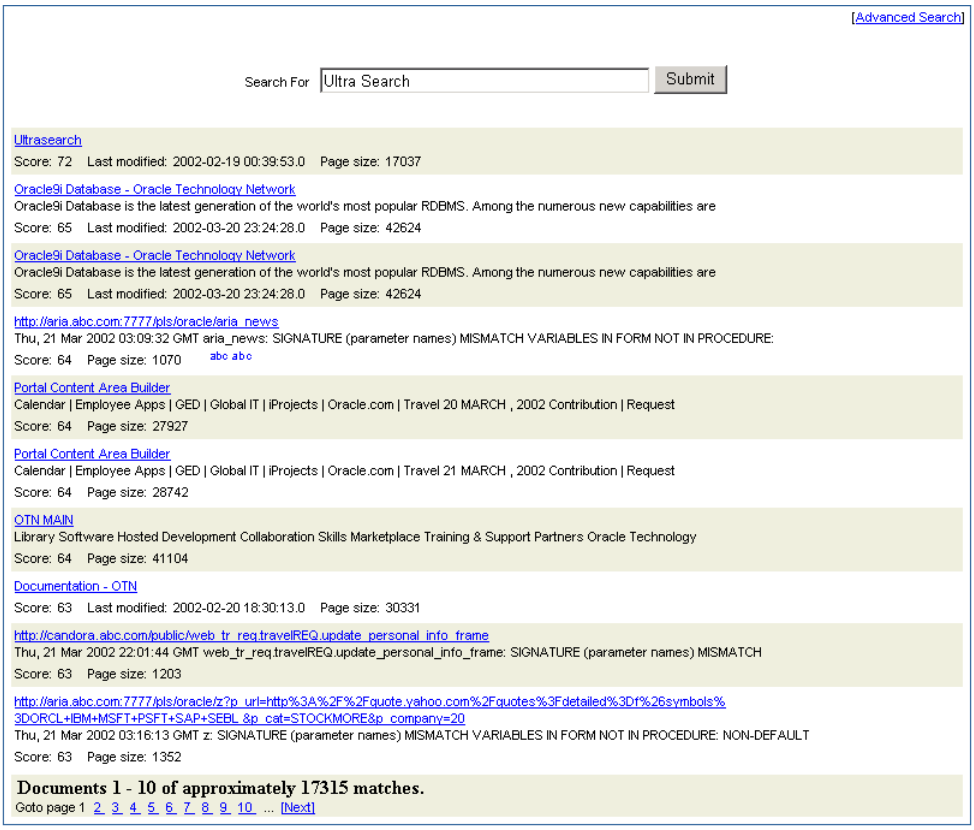
Figure 8–12 Oracle Ultra Search Portlet



The Oracle Ultra Search portlet demonstrates how to write a search portlet for use in OracleAS Portal. When the user issues a search query a list of results matching the user's search criteria are returned, as shown in Figure 8–13.

See Section 8.2.4, "Configuring Oracle Ultra Search Options in OracleAS Portal" for information about how to use the Oracle Ultra Search portlet in OracleAS Portal.

Figure 8–13 Example of Query Results in the Oracle Ultra Search Portlet



If you do not want to use the Oracle Ultra Search sample query applications, you can build your own query application by directly invoking the Oracle Ultra Search Java query API. Because the API is coded in Java, you can invoke the API methods from any Java-based application, such as from a Java servlet or a JavaServer page (as in the case of the provided sample query applications). To display e-mails that have been crawled and indexed, you can also directly invoke the Oracle Ultra Search Java Mail API methods.

See Also:

- *Oracle Ultra Search User's Guide*
- README file located at *ORACLE_HOME*/ultrasearch/sample/sample_readme.htm

8.4.1.2 About the Oracle Ultra Search Administration Tool

The Oracle Ultra Search administration tool is a J2EE-compliant Web application that lets you manage Oracle Ultra Search instances. From the Oracle Ultra Search administration tool you can:

- Create Oracle Ultra Search instances
- Manage administrative users
- Define data sources and assign them to data groups
- Configure and schedule the Oracle Ultra Search crawler
- Set query options
- Translate search attributes and LOV and data group display names to different languages

The Oracle Ultra Search administration tool and the Oracle Ultra Search sample query applications are part of the Oracle Ultra Search middle-tier components module. However, as the Oracle Ultra Search administration tool is independent from the Oracle Ultra Search sample query applications, they can be hosted on different computers to enhance security or scalability.

You can access the Oracle Ultra Search administration tool through OracleAS Portal. In the **Services** portlet, navigate to the **Ultra Search Administration** page. See Section 8.2.4.1, "Accessing the Oracle Ultra Search Administration Tool" for details.

See Also: *Oracle Ultra Search User's Guide*

8.4.1.3 About Oracle Ultra Search Configuration

The *Oracle Ultra Search User's Guide* provides detailed configuration instructions for Oracle Ultra Search.

8.4.2 Sample Oracle Ultra Search Portlet

Oracle Ultra Search provides a search portlet that can be embedded in OracleAS Portal pages. It is implemented as a JavaServer Page (JSP) application and called the Oracle Ultra Search Portlet Sample. The Oracle Ultra Search Portlet Sample is a Web application that complies with the OracleAS Portal portlet interface and this means it can be placed on portal pages.

You will find additional information about the OracleAS Portal portlet interface and the Oracle Application Server Portal Developer Kit, on OTN, http://www.oracle.com/technology/products/ias/portal/pdk.html.

The Oracle Ultra Search Portlet Sample implements a provider that contains exactly one portlet. The provider name is *Ultra Search Provider* and it belongs to the *Oracle Application Server Providers* provider group. The portlet in the Ultra Search provider is also called *Ultra Search*.

Note that Web providers are not registered with OracleAS Portal as part of the Oracle Application Server installation, as the provider must be up and running for

registration to take place. This is not possible because the very last step performed during the installation is the starting of OC4J.

See Section 8.2.4.3, "Registering the Ultra Search Provider with OracleAS Portal" for information about registering the Ultra Search provider.

8.4.2.1 Public Data Searching

The Oracle Ultra Search portlet enables you to add Oracle Ultra Search functionality to portal pages. However, remember that Oracle Ultra Search does not support any security model for search end-users. This means that all data crawled and indexed by Oracle Ultra Search is accessible to all users of a particular Oracle Ultra Search instance. There is no way to specify that a particular portal user has access to a subset of search results returned by Oracle Ultra Search.

8.4.2.2 Sample Portlet Files

The portlet sample files are located in the following file:

ORACLE_HOME/ultrasearch/sample.ear

When the application server first deploys `sample.ear`, the content of this file is expanded into the following directory:

ORACLE_HOME/ultrasearch/sample/query

You can view the source code using your preferred text editor. You can also read the file `ORACLE_HOME/ultrasearch/sample/query/portlet/README.html` for a complete list and description of all the files used by the sample portlet, and a full description of how it works.

8.4.2.3 Restrictions

The list of values in the Oracle Ultra Search portlet does not work when the Oracle Ultra Search provider is running on a different host than the OracleAS Portal middle tier. This is due to a security bug inherent in JavaScript.

9

Tuning Performance in OracleAS Portal

This chapter discusses how you can tune the performance of your OracleAS Portal on the configuration, after you have set up the basic configuration of your portal system.

This chapter contains the following list of options for tuning the performance of OracleAS Portal:

- Setting the Number of Server Processes
- Setting the Number of Idle Processes
- Setting the Number of PPE Fetchers
- Tuning the Oracle HTTP Server
- Generating Performance Reports
- Tuning File System Cache to Improve Caching Performance
- Tuning Oracle Net Services

See Also:

- *Oracle Application Server Performance Guide*
- The Performance page on Portal Center:

  `http://www.oracle.com/technology/products/ias/portal/performance_10g1014.html`

9.1 Setting the Number of Server Processes

Oracle HTTP Server processes Web requests by distributing them to HTTP processes. Oracle HTTP Server can serve all types of requests originating in users' browsers, such as those for static files, Java servlets, or PL/SQL procedures.

`MaxClients` is an Oracle HTTP Server configuration directive that controls the maximum number of Web requests that Oracle HTTP Server can handle at any given time. When the `MaxClients` value is exceeded, Oracle HTTP Server refuses to handle any new requests until it handles the current load and the HTTP processes are freed. In fact, client browsers may be *locked out* if the number of allowable sessions has been exceeded by other browsers.

One way to think of the `MaxClients` directive is that it's a regulator that permits the right flow of concurrent Web requests to your server. Set it too low, and your Web portal performance may suffer. Even though you may have the server and database resources to handle more traffic with quicker response intervals, Web requests cannot get through because you have not set enough processes in `MaxClients`.

Setting `MaxClients` too high unnecessarily consumes resources, because each HTTP process server consumes resources, such as CPU time, memory, and I/O. It may also result in poorer rather than better performance. Oracle HTTP Server can handle all sorts of requests, including those for PL/SQL procedures. When Oracle HTTP Server receives such a request, it hands it off to Portal Services to communicate with the portal database. For each server process that executes a portal database request, there will be a need to cache a database connection. The value you set for `MaxClients`, therefore, sets the upper limit of database connections that Portal Services can open.

Say you set `MaxClients` to the maximum number, 1024. At any given time, Oracle HTTP Server is ready to handle 1024 simultaneous Web requests, including some that require database connections. Even if your server is large enough to deal with this, the database it is connected to may not be. If the ratio of requests for PL/SQL procedures versus other types of requests suddenly becomes very high, you risk overloading your database.

Note: On Windows, consider tuning the Oracle HTTP Server parameter `ThreadsPerChild`.

The key to good performance is determining the number of Web requests the servers in your configuration can process, and how much traffic your database can handle. So if your portal configuration includes multiple middle-tier servers connected to a single database, the number of possible Web requests you can handle is probably going to be limited more by database capacity than the middle tiers.

See Also:

- *Oracle HTTP Server Administrator's Guide*
- "Configuring the MaxClients Setting" in Section 9.4, "Tuning the Oracle HTTP Server"

9.2 Setting the Number of Idle Processes

`MinSpareServers` is a UNIX-specific Oracle HTTP Server directive that sets the minimum number of idle sessions. An idle session is one that is not currently handling a Web request. If the number of idle sessions is fewer than the number specified in `MinSpareServers`, new processes are created at a maximum rate of 1 in every second.

You should consider tuning this parameter only on very busy sites. The default setting is 5. Setting this parameter to a large number is almost always a bad idea. A rule of thumb is to set `MinSpareServers` at a little over the average number of Web requests your portal typically handles. Ideally, you can set it so user requests are filled all the time by open ports without having to open a new one, but this is possible if you have the database resources to support a lot of ports.

Unlike UNIX, Windows is a thread-based operating system where one process is started and then additional child processes are threaded as required. For Windows computers, the directive is called `MaxThreadsPerChild`. This is the number of concurrent requests the server will allow. Set this value according to the responsiveness of the server and the amount of system resources you want to allow the server to consume. `MaxThreadsPerChild` on Windows is equivalent to `MaxClients` on UNIX.

See Also: *Oracle HTTP Server Administrator's Guide*

9.3 Setting the Number of PPE Fetchers

A request for a portal page originates in the form of a URL sent from a user's browser to the HTTP server. If the request is for a portal page, it is forwarded to the Parallel Page Engine (PPE). The PPE then asks each provider that owns a portlet on the page to execute the portlet and return content to the portal page.

There are two options available to enable you to increase the concurrency of the PPE:

Option 1: Create a New OC4J Instance to Create Another Set of PPE Threads

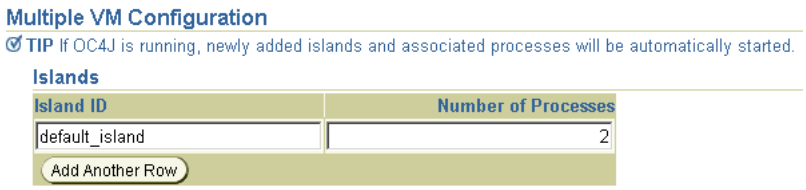
Complete these steps to change the number of *OC4J_Portal* processes:

1. Access the Oracle Enterprise Manager 10*g* Application Server Control Console.

   For details, see Section 7.2.1, "Accessing the Application Server Control Console".

2. Click the link for the application server middle tier where OracleAS Portal is installed.
3. Click the **OC4J_Portal** link.
4. Click the **Administration** link.
5. Click the **Server Properties** link.
6. In the Under the **Multiple VM Configuration** section, change the **Number of Processes** for the `default_island` as shown in Figure 9–1.

Figure 9–1 Multiple VM Configuration Section



7. Click **Apply**.
8. Navigate back to the **OC4J_Portal** home page.
9. Click **Restart**, to restart the OC4J_Portal instance.

Alternatively, you can edit the file `opmn.xml` manually, though the use of Application Server Control Console is the recommended approach.

The parameter to create multiple Oracle Containers for J2EE instances is called `numProcs` and is configured in the file *ORACLE_HOME*`/opmn/conf/opmn.xml`

The changed file would look something like this:

```
<oc4j instanceName="OC4J_Portal" gid="OC4J_Portal" numProcs="2">
  <config-file path="E:\Ora902\j2ee\OC4J_Portal\config\server.xml"/>
  <java-option value="-server -Xincgc -Xnoclassgc -Xmx100m "/>
  <oc4j-option value="-properties"/>
  <port ajp="3001-3100" rmi="3101-3200" jms="3201-3300"/>
  <environment>
    <prop name="PATH" value="E:/Ora902/bin"/>
    <prop name="DISPLAY" value="localhost:0"/>
  </environment>
</oc4j>
```

For the configuration changes to take effect perform the following steps:

1. Run the following command:

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

2. Restart the Oracle Application Server middle tier as follows:

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

Option 2: Increase the Value of Default Number of Threads

The PPE uses a pool of *fetchers* to forward requests to providers and wait for data to be returned. Once it is finished with the request, the fetcher is available to handle another new request.

The parameter to tune the number of PPE threads is called `poolSize` and is configured in the file *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF/web.xml`.

The default setting is `25`. For most Web portals, you should never have to change pool size. But keep in mind that if pool size is too low, the user notices that pages take too long to draw at peak periods. If pool size is set too high, a possible resource drain may occur because too many concurrent URL requests can overwhelm the PPE.

The changed file would look something like this:

```
<web-app>
  <servlet>
    <servlet-name>page</servlet-name>
      <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
    <init-param>
      <param-name>logpath</param-name>
      <param-value>./</param-value>
    </init-param>
    ...
    <init-param>
      <param-name>poolSize</param-name>
      <param-value>50</param-value>
    </init-param>
    ...
  </servlet>
...
```

For the configuration changes to take effect:

1. Run the following command:

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

2. Restart the Oracle Application Server middle tier.

> **Note:** Even for a site with high traffic, the `PoolSize` parameter should not be set to beyond the range of 50-125. If there is a need to set the value higher than this, then consider adding more OC4J instances. Refer to "Option 1: Create a New OC4J Instance to Create Another Set of PPE Threads" for the procedure to do this.

9.4 Tuning the Oracle HTTP Server

However you choose to configure the Oracle HTTP Server listener, you can optimize performance by setting an approximate number of simultaneous requests that can be handled by the Oracle HTTP Server listener.

The total number of database sessions needed to run OracleAS Portal is a factor of the total number of concurrent requests that simultaneously need to access the portal repository. The total number of concurrent requests of any type that can be serviced by a given instance is itself a factor of the Oracle HTTP Server configuration parameter called `MaxClients`.

Each connection to the portal repository results in one network connection and two sessions (the two sessions use the same physical connection). The first session is for *portal* and the second is for *portal_public*.

Therefore, if `MaxClients` is set to 150, but the maximum number of concurrent requests that ever need to simultaneously access the portal repository is 50, then you need to ensure that the database is configured to allow for (50*2) = 100 sessions.

It is theoretically possible, though extremely remote, that all 150 requests have to simultaneously access the portal repository, in which case the number of database sessions required would be (150*2) = 300.

If the number of concurrent requests that need to simultaneously access the portal repository is high, and the allowed number of sessions has been exceeded, then clients may be "locked out". However, setting a very high value for the number of sessions will unnecessarily consume resources.

Notes:

1. Typical requests that are concurrently being serviced at any point in time consist of a wide variety of request types (for example, static images, portal page requests, and other Oracle HTTP Server requests), and the real number of concurrent requests that simultaneously need to access the portal repository is quite small.
2. As OracleAS Portal leverages OracleAS Web Cache and the Portal File Cache to cache content, many times OracleAS Portal does not need to contact the portal repository at all, thereby reducing the database session requirements.
3. For portal page requests, the `poolSize` parameter in the PPE acts like a throttle, which reduces the possibility of flooding the system with concurrent requests for portal pages.
4. This section only talks about sessions needed by the Portal Services running inside OracleAS Portal. Other entities which connect to the same repository must also be accounted for.

Configuring the MaxClients Setting

Because login frequency is generally lower than OracleAS Portal access frequency, it makes sense to configure the OracleAS Single Sign-On on a separate Oracle HTTP Server listener. The objective is to tune down the `MaxClients` setting to a value that is reasonable, without affecting the needs of the portal system.

See Also: *Oracle Application Server Performance Guide*

Perform the following steps to configure the `MaxClients` setting:

1. For the OracleAS Single Sign-On's listener, once you've determined the approximate value to set for the `MaxClients` parameter, edit this accordingly in the configuration file, `httpd.conf`, which is located in:

   ```
   ORACLE_HOME/Apache/Apache/conf/
   ```

   Tune down the `MaxClients` setting to control the number of requests that Oracle HTTP Server services on the Oracle HTTP Server listener. This controls the maximum number of sessions that can be established.

2. For the OracleAS Portal listener, you can separately tune the `MaxClients` parameter according to the needs of the OracleAS Single Sign-On and the needs of OracleAS Portal, without creating a conflict. This parameter directly corresponds to the number of sessions established and to the maximum workload that the Oracle HTTP Server listener can handle on the portal listener.

   The following example shows the `MaxClients` section in the `httpd.conf` file:

   ```
   # Limit on total number of servers running, that is, limit on the number
   # of clients who can simultaneously connect --- if this limit is ever
   # reached, clients are LOCKED OUT, so it should NOT BE SET TOO LOW.
   # It is intended mainly as a brake to keep a runaway server from taking
   # the system with it as it spirals down...
   #
   MaxClients 150
   ```

Notes:

- If you tune the OracleAS Single Sign-On and the OracleAS Portal separately, each will have a separate listener. The OracleAS Portal will control the resources (sessions) on the portal database and the OracleAS Single Sign-On will control the resources on the OracleAS Single Sign-On database.
- The number of sessions and connections that the database permits is limited by the value set in the Oracle Database 10*g*'s `init.ora` file. Refer to the Oracle Database 10*g* documentation library for more information.

9.5 Generating Performance Reports

This release includes a set of SQL scripts that can generate performance reports for OracleAS Portal. Other than using these scripts, there is no way to obtain performance-reporting information. These scripts allow a portal administrator to load OracleAS Portal log files into a database table and create reports based on that information. The scripts are located in the following directory:

ORACLE_HOME`/portal/admin/plsql/perf`

The file `README.html` in the `scripts` subdirectory explains how the scripts can be used to monitor OracleAS Portal performance.

The statistics collected indicate, among other things, how long overall requests take to complete, how much of that time was spent in the user's procedure, which user made the request, whether a database connection was obtained from the connection pool, and what type of caching was used. The performance scripts also enable you to extract information similar to that, which was available in earlier releases of OracleAS Portal. Some of the performance reports that you can generate include:

- Unique logins for each day, or hour
- Page views for each day, or hour
- Top ten pages and portlets and their response time
- Response times
- Peak login time each day
- Logins for each day
- Portlets execution time
- Slowest portlet
- Total hits for each day
- Most and least popular portlets
- Unique users logged in each day
- Page hits for each day
- Portlet hits for each day
- Request breakdown by IP address and hostname

9.6 Tuning File System Cache to Improve Caching Performance

Tuning the File system cache can increase caching performance. Two ways of tuning the file system cache are:

- Configuring File System Cache to Reside on a Faster File System.
- Moving Session Cache Directory to More Performant File System.

> **Note:** Starting with this release, OracleAS Portal implements in-memory caching of the session cache. This functionality reduces the possibility of contention for reading frequently used session cache objects, thereby reducing the need to move the session cache content to a more performant file system.

More information on how to do this can be found in the section describing how to optimize PL/SQL performance, in the *Oracle Application Server mod_plsql User's Guide*.

9.7 Tuning Oracle Net Services

Portal Services leverage Oracle Net Services to connect to the OracleAS Portal schema in the OracleAS Metadata Repository. By tuning Oracle Net Services, you can improve database access performance

Refer to the paper, "*Tuning Oracle Net Services to optimize mod_plsql Database access times*" on the Oracle Technology Network (OTN) at `http://www.oracle.com/technology/`. The tuning steps mentioned in this document are still applicable to OracleAS Portal.

10

Exporting and Importing Content

OracleAS Portal provides a set of export and import utilities that enable you to transfer content between portals. This chapter provides a summary of recommendations and best practices for using the export and import utilities. This chapter contains the following main sections:

- "Before You Start OracleAS Portal Export or Import"
- "Export and Import in OracleAS Portal"
- "Behavior of Objects After Migration"
- "Recommended Best Practices When Exporting and Importing"

10.1 Before You Start OracleAS Portal Export or Import

This section describes the tasks you need to perform and information about how OracleAS Portal Export and Import works with other components of Oracle Application Server. This section contains the following topics:

- How Does OracleAS Portal Export and Import Work?
- Additional Information

How Does OracleAS Portal Export and Import Work?

The OracleAS Portal Export and Import process consists of the following steps:

1. Create **transport sets** and extract the content of the transport sets to transport tables. Transport sets contain the portal objects that you want to export to your target portal environment. This information is displayed in a **manifest**. The manifest is a list of objects in a transport set, used to provide a granular level of control over the export.
2. Move the transport sets from one system (source) to another (target) using the OracleAS Portal Export and Import command-line scripts, which create a dump file of the transport set.
3. Transfer the Export and Import command-line script and the dump file to the target system using FTP or another file transfer utility.
4. Invoke the Export and Import command-line script to import the dump file to the transport tables on your target system.
5. Import the objects from the transport tables to the target portal repository using the Transport Set Manager portlet.

Additional Information

- Refer to the section on moving product-specific metadata for OracleAS Portal from a test metadata repository to a production metadata repository in the *Oracle Application Server Administrator's Guide*.
- Refer to the section about controlling the exporting and importing of portlet personalizations in the *Oracle Application Server Portal Developer's Guide*.

10.2 Export and Import in OracleAS Portal

This section describes the prerequisites to exporting and importing in OracleAS Portal, use cases, and the OracleAS Portal Export and Import processes. This section contains the following topics:

- What Do I Need to Check Before I Begin?
- Examples of Using Export and Import
- OracleAS Portal Export and Import - Recommended Method
- OracleAS Portal Export and Import - Alternate Method

10.2.1 What Do I Need to Check Before I Begin?

Before beginning the export and import process, ensure you have the following information:

- System Requirements
- Additional Considerations
- Privileges for Exporting and Importing Content
- OracleAS Portal instance information:
  - Portal schema name.
  - Portal schema password.
  - Portal connect string information.
  - Portal user name.
  - Portal user password.
  - Company name (used only for hosted portal installations), in most cases, leave it blank.

Note: The OracleAS Portal schema password is a random password created when the application is installed.

10.2.1.1 System Requirements

Before exporting and importing content, ensure that your system meets the minimum system requirements, as described in this section.

Notes:

- Export and import functions only within the same release of OracleAS Portal and the same patch release, for example, release 9.0.4.0 to release 9.0.4.0 or release 10.1.2 to release 10.1.2. You cannot export and import between two different releases, such as release 9.0.4 to release 10.1.2 or release 9.0.4.0 to release 9.0.4.1.
- For successful migration of objects, the version of the portal repository should be the same in the target and the source. Any difference in the versions of the middle tiers does not impact migration.

- **Using Different Releases of Export**. Whenever you move data between different releases of Oracle Database, the following rules apply:
  - The Oracle Database `imp` utility and the database to which data is being imported (the target database) must be either the same release or a later release.
  - The release of the Oracle Database `exp` utility must be same as the earliest release of the source or target database.

Notes:

- Oracle Database `exp` and `imp` are the export and import utilities used to dump and restore data in an Oracle-specific format for backup and transfer of user data.
- If you have problems with database release mismatches, then contact Oracle Support Services.

The choice to use the database Oracle home or the middle-tier Oracle home depends on the release of the database used for the source and target portal installations. By default, the 10.1.2 release of the middle tier uses a 10.1.0.2 release Oracle home.

Based on the recommendations given earlier, the following conditions apply when a 10.1.2 release of a portal and 10.1.2 release of a middle tier is involved:

- Always use the middle-tier Oracle home to export content. Version 9.0.1.5 is the earliest version of the database supported for a 10.1.2 release of a portal installation.
- Always use the target database Oracle home to import content. The release of the import utility and the target database must be the same.

Note: If you have configured a 9.0.4 release of a portal (9.0.4 or 9.0.4.1) to use a 10.1.2 release of a middle tier, then the rules described in this section must be followed.

For example, to create an export file that will be imported into a later release of a database, use a release of the Oracle Database `exp` utility that is the same as the source database. To create an export file that will be imported into an earlier

release of a database, use a release of the Oracle Database `exp` utility that is the same as the release of the target database.

> **Note:** Oracle recommends you use the same release of the database for the source and target portal installations.

- **Oracle export and import and character sets.** The Oracle Database `exp` utility always exports user data, including Unicode data, in the character sets of the export server. The character sets are specified when the database is created.

  The Oracle Database `imp` utility automatically converts the data to the character sets of the import server.

  Some 8-bit characters can be lost (that is, converted to 7-bit equivalents) when you import an 8-bit character set export file. This occurs if the client system has a native 7-bit character set or if the NLS_LANG operating system environment variable is set to a 7-bit character set. Most often, you notice that accented characters lose their accent marks.

  Both the Oracle Database `exp` and `imp` utilities alert you of any required character set conversion before exporting or importing the data.

  > **Note:** When the character set width differs between the export client and the export server, the data may be truncated if the conversion causes the data to expand. If truncation occurs, then the export displays a warning message.

- **Understand your source and target portal instances.**
  - **Do you have command-line access to appropriate directories on the source and target computers?** You must have command-line access to run the shell or command utilities generated by the export import process. The command-line utilities, in turn, access the Oracle Database `exp` and `imp` utilities, and the OracleAS Portal instance.
  - **Is your database configured to allow background jobs to run?** Each export or import process sets up a background process. Therefore, verify that the `job_queue_processes` database parameter is set appropriately.

    To check the value of the `job_queue_processes` parameter, perform the following query from SQL*Plus:

    ```
    %select name, value from v$parameter where name='job_queue_processes'
    ```

    The value for `job_queue_processes` should be at least `2` to allow the background jobs to run.

    An alternative way of checking the `job_queue_processes` parameter is to examine the `init.ora` file in your database's *`ORACLE_HOME`*.
- **When do you export and import data?** Perform the export and import process after regular business hours, and disable access to OracleAS Portal during the process. One way to disable access to the portal temporarily is to configure your listener for a different port number for the duration of the export and then revert to the original port when the export process is complete.

> **Note:** If the Oracle Database `exp` and `imp` utilities finish with errors or warnings, then you should not import that transport set. The errors or warnings that are recorded in the Oracle Database `exp` and `imp` log files (typically named *`<script_file_name>_<long identifier>`*`_exp.log` and *`<script_file_name>_<long identifier>`*`_imp.log`) should be corrected first.

- **How much time does the export or import process take?** The exact amount of time to export or import content in OracleAS Portal cannot be determined. Many dependencies affect the time it takes to export and import content. The following are dependencies that can affect the processing time.

  Dependencies that affect the **Export** process are as follows:

  - Objects being exported have number of dependencies spanning across page groups.
  - References or dependencies between objects.
  - Extraction is taking a long time to start, which depends on the assigned database job in the queue.
  - The extraction process is taking a long time due to a large number of documents being extracted, especially BLOB columns.
  - Insufficient memory in the TEMP tablespace for sort operations.
  - Schema validation taking a long time, due to a large number of objects that need to be validated.

  Dependencies that affect the **Import** process are as follows:

  - Preliminary check for large page groups, which also depends on the number of internal and external dependencies that need to be checked.
  - Import process taking a long time to start, which depends on the assigned database job in the queue.
  - Insufficient memory in the TEMP tablespace for sort operations.
  - Post-import schema validation taking a long time due to a large number of objects being validated.
  - Difference between the source and target languages is reasonably high.

    > **Tip:** Before importing large transport sets, you could increase the values of relevant database cache parameters based on your requirement. This will reduce the time taken to import large transport sets reasonably.

10.2.1.2 Additional Considerations

This section provides a list of some additional considerations you must make before you export and import data in OracleAS Portal.

- When exporting or importing large data sets, ensure that there is sufficient space in the TEMP tablespace. This ensures that the export or import process does not fail due to insufficient memory.
- For exporting large page groups, use the `opeasst.csh` script. See Section 10.2.3.1.3, "Exporting Large Page Groups" for more information.

- For importing large page groups, use the import script with the `-automatic_merge` option. See Section 10.2.3.2.1, "Running Your Script on the Target System" for more information.
- If you have installed any Business Intelligence and Forms components and use related portlets in OracleAS Portal on the source portal instance, then you must ensure that the same components are installed on the target portal instance before you can export and import data between the portal instances. If the same Business Intelligence and Forms components are not found on the target portal instance, then, during import, the portlets related to those components will be removed from the pages in which they appear.

> **Caution:** Do not manually update system tables to resolve any issues you might have in the source or target portal instances. Doing so will cause the export and import process to fail. If you have any problems with source or target instances, then contact Oracle Support Services.

10.2.1.3 Privileges for Exporting and Importing Content

This section describes the privileges required to successfully export and import content. The privileges described subsequently apply to the export and import of Oracle Instant Portal content also.

Privileges for Exporting Content

To allow for secured control over the export of shared objects (objects in the Shared page group), there are two privileges defined at the infrastructure level.

- **Any Transport Set - Manage** enables you to export and import portal objects, including shared objects. This privilege is granted to the DBA group by default during the portal installation process.
- **Any Transport Set - Execute** enables you to export and import portal objects, excluding shared objects. This privilege is granted to the PORTAL_ADMINISTRATORS group by default during portal installation process.

Table 10–1 provides a description of export user privileges.

Table 10–1 Export User Privileges

User Privileges	Export Objects That Are Not Shared?	Export Objects That Are Shared?
Any Transport Set - Manage	Yes	Yes
Any Transport Set - Execute	Yes	No
Any Transport Set - None	No	No

Privileges for Importing Content

In addition to the **Any Transport Set - Manage** privilege, you must also have the **Manage** privilege on objects of a given type to successfully import content.

For example, a page group containing Web providers requires you to have **Manage All** privileges on All Providers and All Page Groups to import that page group. Table 10–2 provides a description of each object type and the required privilege level.

Note: The ORCLADMIN and OracleAS Portal users are granted the **Manage All** privilege on all page groups at the time of installation or upgrade. Members of the DBA group are also granted the **Manage All** privilege on all page groups by default.

Table 10–2 Import User Privileges

Object Type	Privileges
All Page Groups	**Manage All** and **All Providers Manage** are required to import page groups and shared objects.
All Providers	**Manage** is required to import page groups, Portal DB Providers, Web providers, WSRP producers, and other database providers.
All Portal DB Providers	**Manage** is required to import Portal DB Provider objects.
All Shared Components	**Manage** is required to import shared components if the Portal DB Provider objects reference the shared components.

Note: If you import a page based on a style that belongs to the shared objects group and do not have the necessary privileges to import shared objects, then the style of the page is reset to **Main Style** by default.

10.2.2 Examples of Using Export and Import

OracleAS Portal supports the ability to copy or update page groups and portal content between your source and target destination portal instances. This section gives examples of the most common uses of the OracleAS Portal Export and Import processes.

10.2.2.1 Case 1: Exporting and Importing Between Development and Production Instances

This case shows the steps to copy or update portal page groups and portlets between a development instance and a production instance of OracleAS Portal.

Note: User personalizations are not exported; therefore, any personalizations of a page or portlet on the source are not exported or imported.

Scenario 1: Exporting pages and content to a target portal system. The first export to your target system must migrate the entire page group. The following steps provide an overview of the process:

1. Develop page groups, applications, and content on the source system.
2. Identify pages, applications, and content to export, then create transport sets accordingly and export to the target system.
3. Import the transport sets on the target system, into your portal repository.

Scenario 2: Updating content on your target instance. OracleAS Portal supports updating items and region-level content on your target system only under the following circumstance:

Export and import of all changes from the source to the target instance. All page structure, content, and user preferences on your target system are replaced with the content from your source system. The first export to your target system migrates the entire page group from the source portal to the target portal instance.

See Section 10.4, "Recommended Best Practices When Exporting and Importing" for more information about the recommended practices for exporting and importing content.

Note: Editing of imported content is not supported from 10*g* Release 2 (10.1.2) onwards.

For example, if you have a page (named `Page1`) within a page group (named `PG1`) on the source that was migrated to the target, then you must not edit `Page1` on the target.

Similarly, if you have a page (`Page1`) within a page group (`PG1`) on the source, then you must not create a page with the name `Page1` within page group (`PG1`) on the target.

10.2.2.2 Case 2: Deploying Identical Content Across Multiple Portal Instances

Oracle Database `exp` and `imp` utilities can be used to deploy identical content across multiple OracleAS Portal instances. In this case, the OracleAS Portal objects (portlets, page groups, and so on) can be created in one instance, and propagated to multiple instances using the Oracle Database `exp` and `imp` utilities. For more information, refer to the information on staging a test environment from a production environment, in the *Oracle Application Server Administrator's Guide*.

10.2.2.3 Case 3: Consolidating Content from Multiple Sources

When you use OracleAS Portal Export and Import to migrate content from multiple portal instances to a single target portal instance, you must consider the following points:

- Do not create objects with the same names on different source portal instances from where you plan to import. This helps avoid namespace collisions between shared objects. For example, assume that you create a shared template (`shared_tempate1`) in source instances (`source1` and `source2`) used by page groups (`pgrp1` and `pgrp2`) in `source1` and `source2` respectively. Now, if you try to consolidate the two page groups from `source1` and `source2` into one target instance, then this will result in errors as both page groups use different shared templates with the same name (`shared_template1`).
- Do not create page groups with the same name. For example, do not create a page group (`pgrp1`) in source instances `source1` and `source2` if you need to consolidate these two page groups in into a single target instance. This warning is also valid for names of database provider objects, shared components, Web providers, and database providers.

10.2.3 OracleAS Portal Export and Import - Recommended Method

This section describes the recommended method to export and import content in OracleAS Portal. It contains the following topics:

- How Does OracleAS Portal Export Work?
- How Does OracleAS Portal Import Work?

- How Do I Manage My Transport Sets?

10.2.3.1 How Does OracleAS Portal Export Work?

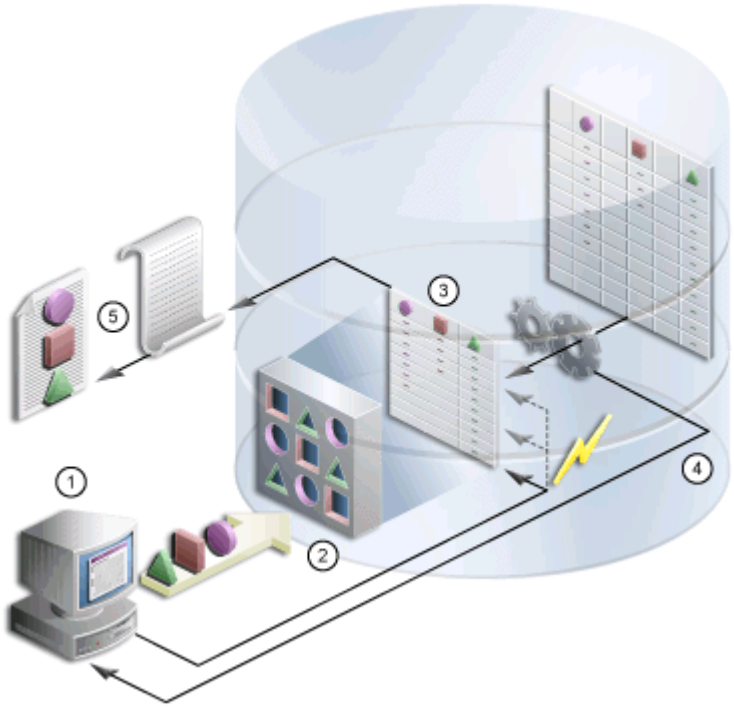
This section describes the export process and the steps required to successfully transfer content from the source portal system, including:

- Creating Transport Sets
- Exporting Data
- Exporting Large Page Groups

10.2.3.1.1 Creating Transport Sets Once the system requirements are verified, your goal is to create a transport set. Figure 10–1 shows the process.

> **Note:** Limit any possible conflict issues by making one person responsible for maintaining a transport set.

Figure 10–1 Export Process



1. From the Navigator or Bulk Actions (enables you to add multiple pages at once to the export transport set), select the objects to be exported. The Transport Set Manager is automatically displayed.
2. Select a name and select the export options for the transport set. In the Transport Set Manager, click **Export Now** to initiate the export.
3. The procedure extracts the data and populates the transport tables.
4. Generate a migration script and log information from the Transport Set Manager.
5. Run the script to generate a dump file.

The Export/Import Dependency Manager ensures that all the object dependencies in the transport set are correctly extracted. Specifically, the Dependency Manager classifies each object as explicitly selected, referenced, external or child, based on how the object is related to the objects being explicitly exported. The information is displayed in the manifest, as shown in Figure 10–2. The following list shows the classification of objects:

- **Explicitly Selected Objects**. Objects, that were explicitly selected, from the Navigator or Bulk Actions for export.
- **Referenced Objects**. Objects that are directly or indirectly referenced by the explicitly selected objects, but are always within the same page group as an explicit object. For example, a style used by a page is a referenced object when it belongs to the same page group.
- **External Objects**. External objects ensure that the explicitly selected objects perform on the target portal. For example, external providers and database schemas could be considered external objects. Generally, shared objects and components are external objects unless explicitly selected.
- **Child Objects**. Objects that are part of a hierarchy. For example, subpages, subcategories and subperspectives are child objects of a page, category and perspective.

Notes:

- When a referenced object contains child objects, the child objects are imported in Reuse mode. You should therefore explicitly select the referenced object and include it in the transport set. This will enable you to set the import mode to **Replace on Import**. Before importing the page group in Reuse mode, note the page group properties. After importing the page group manually, update any changes to reflect the old properties.
- A child object is picked up for migration only for an explicit object. If a parent page, category, or perspective appears in the referenced section, then the child objects are not picked. See Table 10–10, " Import Behavior of Child Objects" for more information.
- Containers of explicit objects and referenced objects are migrated as external dependencies.

Working with Import Modes

The manifest provides a granular level of control over the import mode. The manifest is the list of objects in a transport set. There are two modes available during import:

- **Replace on Import**. If the object exists on the target, then it is replaced. If it does not exist, then it is created. If this mode is not selected and the object exists, then the object on the target portal is retained as is. However, if the object does not exist on the target, then it is created.
- **Reuse on Import**. If the object does not exist on the target, then it is created. If it already exists, then it remains as is.

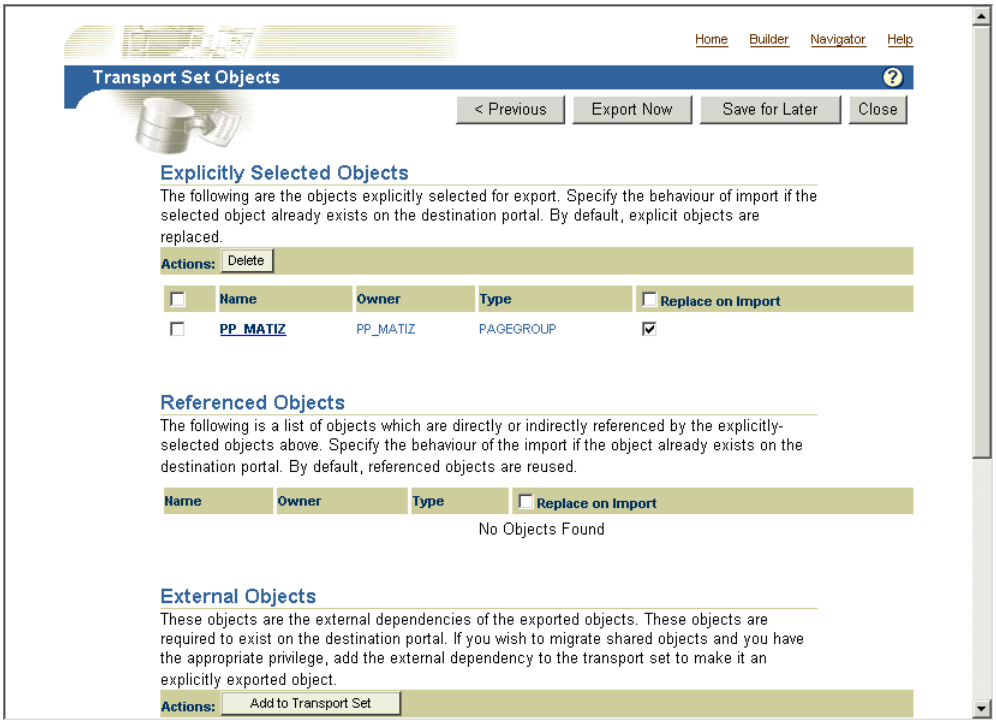
Table 10–3 describes the object classification and the default modes.

Table 10–3 Default Modes

Object Classification	Default Import Mode
Explicitly selected objects	Replace on Import
Referenced objects	Reuse
Child objects	Replace on Import
External objects	Reuse

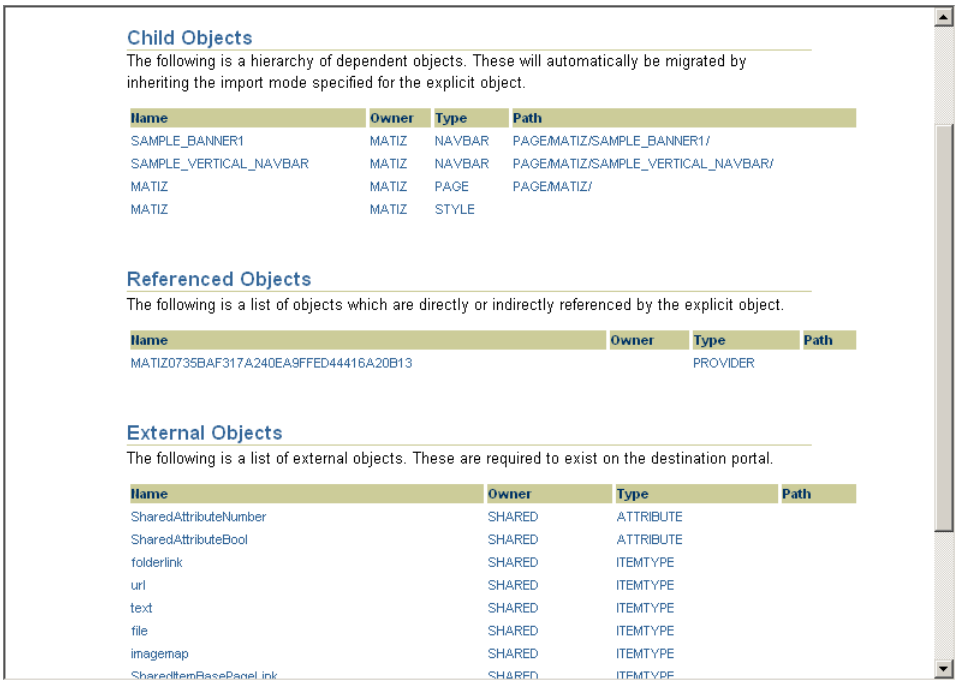
Figure 10–2 is an example of a transport set manifest.

Figure 10–2 Transport Set Manifest



Clicking the name of an object, for example an explicitly selected object, displays a detailed screen of child, referenced, and external objects. Figure 10–3 is an example of a detailed manifest screen.

Figure 10–3 Detailed Manifest Screen



Child Objects

The following is a hierarchy of dependent objects. These will automatically be migrated by inheriting the import mode specified for the explicit object.

Name	Owner	Type	Path
SAMPLE_BANNER1	MATIZ	NAVBAR	PAGE/MATIZ/SAMPLE_BANNER1/
SAMPLE_VERTICAL_NAVBAR	MATIZ	NAVBAR	PAGE/MATIZ/SAMPLE_VERTICAL_NAVBAR/
MATIZ	MATIZ	PAGE	PAGE/MATIZ/
MATIZ	MATIZ	STYLE	

Referenced Objects

The following is a list of objects which are directly or indirectly referenced by the explicit object.

Name	Owner	Type	Path
MATIZ0735BAF317A240EA9FFED44416A20B13		PROVIDER	

External Objects

The following is a list of external objects. These are required to exist on the destination portal.

Name	Owner	Type	Path
SharedAttributeNumber	SHARED	ATTRIBUTE	
SharedAttributeBool	SHARED	ATTRIBUTE	
folderlink	SHARED	ITEMTYPE	
url	SHARED	ITEMTYPE	
text	SHARED	ITEMTYPE	
file	SHARED	ITEMTYPE	
imagemap	SHARED	ITEMTYPE	
SharedItemBasePageLink	SHARED	ITEMTYPE	

Note: Editable seeded item types are extracted. It is recommended that you do not edit seeded types. If you want to extract them, then create custom types in the Shared Objects page group based on the existing seeded types. The Dependency Manager includes these in the manifest.

10.2.3.1.2 Exporting Data

Review Section 10.3, "Behavior of Objects After Migration" before exporting and importing your portal content from a source to a target instance.

Note: Portlet repository information (security, organization, and so on) related to the portlet is not migrated during the export and import process.

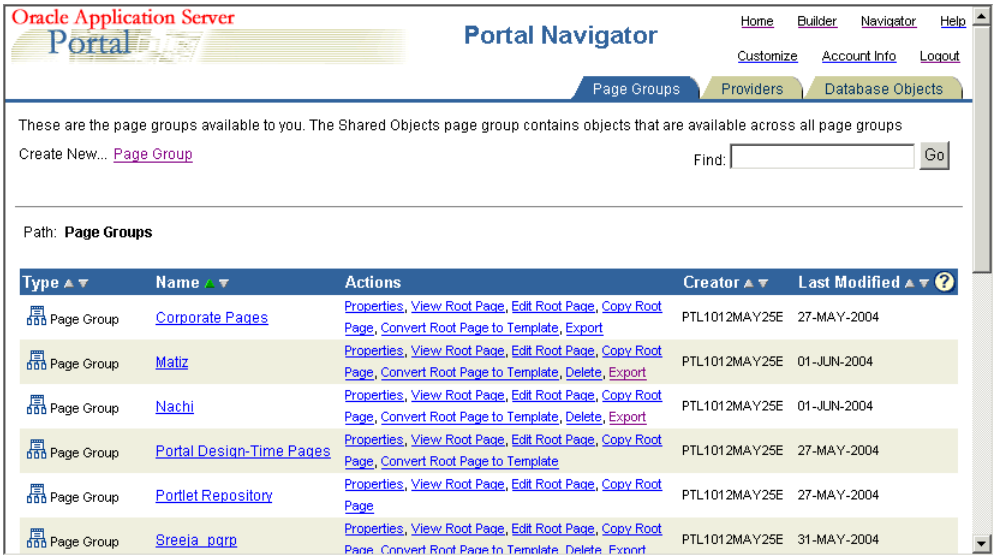
To create a transport set for export:

1. Select the objects for export. You can do this from the **Navigator**, or search results **> Bulk Actions** for page groups. See Figure 10–4.

Note: Be sure to export portlets (Portal Forms, Portal Reports, Charts, Dynamic Pages) before exporting portal pages and page groups that reference them.

Figure 10–4 is an example of the Portal Navigator.

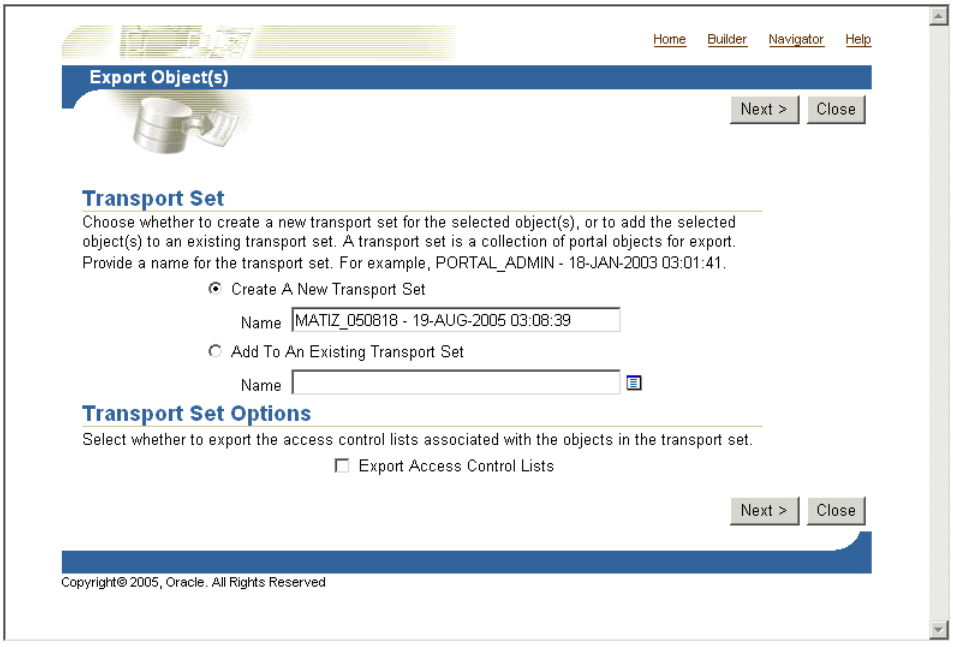
Figure 10–4 Portal Navigator



2. Click the **Export** link to display the Transport Set Manager. Make the transport set name as descriptive as possible, and avoid using any special characters at the start of the name. For example, *My Company Transport Set 18-JAN-2003*.

   Figure 10–5 is an example of the Transport Set Manager.

Figure 10–5 Transport Set Manager



3. Select **Export Access Control Lists** if you want to include access control lists (ACLs) associated with the objects in the transport set. If you select this option, the following happens:

   - Users and groups associated with the objects are migrated.
   - Privileges attached to the objects are migrated.

- Parameters and events associated with the users are migrated.

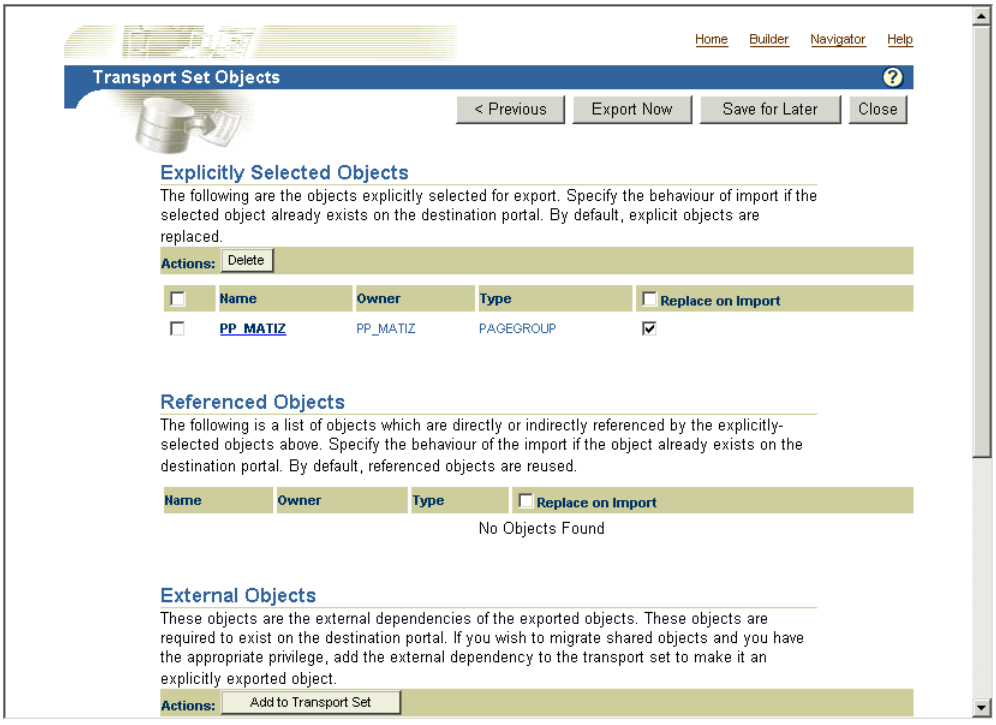
4. Select the import modes, delete any explicitly selected objects, and add (make explicit) any external objects. Making an external object explicit enables you to add a new object to a transport set instead of going back to the Portal Navigator and adding it. External objects are not exported or imported by default until they are added as explicitly selected objects. See Figure 10–6.

5. Select either **Export Now** if you are finished, or **Save for Later** if you want to add more objects. See Section 10.2.3.3, "How Do I Manage My Transport Sets?" for more information about how to edit and browse the transport sets on the system.

Note: When you select some of the transport set options and select **Save for Later**, the next time you add an object to the transport set, all of the previously selected options are reset. Therefore, you need to select the options each time you add an object until you finalize the transport set.

Figure 10–6 shows the transport set objects.

Figure 10–6 Transport Set Manager Objects

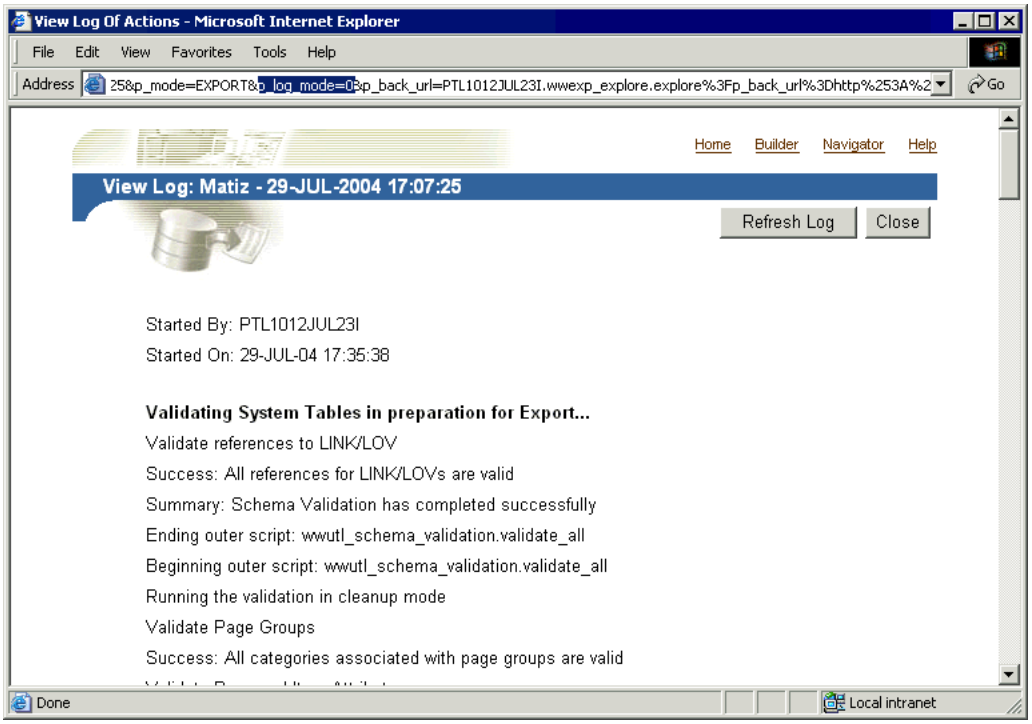


6. To finalize the transport set, click **Export Now**. The objects marked for export are copied to the transport tables for migration. These operations happen in the background.

7. Check the log in your transport set manager for any errors by clicking the **View Log Of Actions** link.

   Figure 10–7 is an example of the **View Log** page.

Figure 10–7 Transport Set Export Log



Note: To view a detailed log of the export process, including debug messages, perform the following steps:

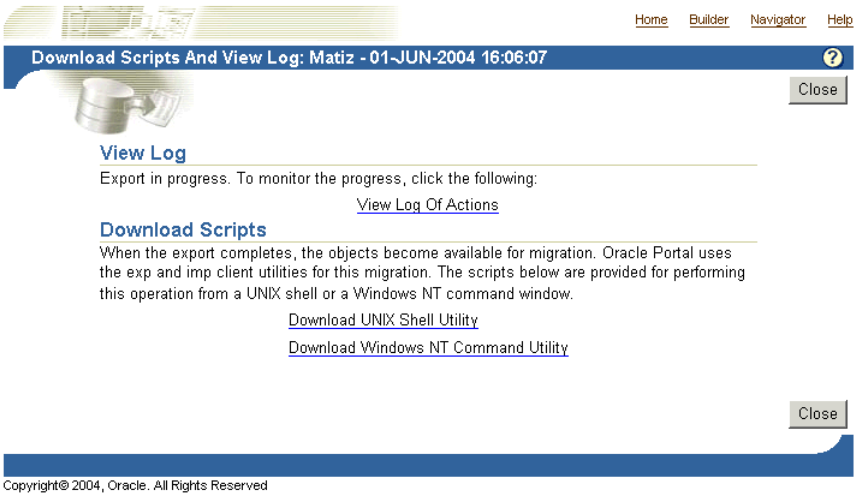
1. In the Transport Set Export Log page (shown in Figure 10–7), go to the Address or Location bar in your browser, and search for the string `p_log_mode=0` in the URL (shown highlighted in the screenshot).
2. Change `p_log_mode=0` to `p_log_mode=1`.
3. Press **Enter**.

---

8. Select an appropriate export script based on your operating system.

   Figure 10–8 is an example of the **Download Scripts And View Log** page.

Figure 10–8 Portal Migration Scripts



For Netscape users:

1. Click the selected script, then click **Save Target As**.
2. Change the name and remember to include the correct file extension, `.csh` for UNIX or `.cmd` for NT. For example, `MyScript.csh`.
3. Save the file to the directory on your file system where you want to run the export script. Usually, this directory is where your export portal resides.

> **Note:** UNIX users should save the file to a local directory and transfer the script to the middle-tier computer where the Oracle Database `imp` utility resides to create the dump file. Ensure that you do not edit the script.

For Internet Explorer users:

1. Right-click the selected script, then click **Save Target As**.
2. Change the name and remember to include the correct file extension, `.csh` for UNIX or `.cmd` for NT. For example, `MyScript.csh`.
3. Save the file to the directory on your file system where you want to run the export script. Usually, this directory is where your export portal resides.

> **Note:** This location must have access to the database. On some systems, the downloaded UNIX script requires you to set the Execute permissions correctly before running it. Ensure that you do not edit the export script.

Running Your Script to Create an Export Dump File

The next steps in the export process are to create a transport set dump file using the script you created in the previous section, and then transfer your export data to your target system.

To create a dump file:

1. Run a script with the parameters shown in the following example. The example assumes that the name of the script is `MyScript.csh`. The parameters in bold are applicable only for export, and they are mandatory.

```
%MyScript.csh
Usage: MyScript.csh <-mode export_or_import> <-s portal_schema>
<-p portal_password> <-pu portal_username> <-pp portal_userpassword>
<-company company_name> <-c connect_string> <-d dump_file_names>
<-automatic_merge>
```

Notes:

- Remember to set the infrastructure Oracle home (*ORACLE_HOME*) when running the export script.
- The value for the `company_name` parameter is the company name you see in the login page when working in a hosted portal. When working in a portal that is not hosted, the value for the parameter should be `none`. If you are running the script in interactive mode, then do not pass a value. Ensure that you do not edit the export script.

Table 10–4 provides a description of the parameters you can use in this process.

Table 10–4 Parameter Descriptions

Parameters	Description
`-mode`	Mode for invoking the Export Import Command Line Utility EXPORT mode: Exports content to dump files using the Oracle Database `exp` utility. IMPORT mode: Imports content from dump files using the Oracle Database `imp` utility.
`-s portal_schema`	Oracle Database account for the portal
`-p portal_password`	Oracle Database password for the portal
`-pu portal_username`	Lightweight user name for logging in to the portal
`-pp portal_userpassword`	Lightweight user password for logging in to the portal
`-company company_name`	Company name (for example, ORACLE)
`-c connect_string`	TNS connection information to the remote database
`-d dump_file_names`	Names of files for Oracle Export or Import utilities to write to or read from. If multiple file names are specified, then they must be separated by commas. For example: `FILE1.DMP,FILE2.DMP` **Note:** If multiple file names are not specified, then the Export or Import utilities will automatically prompt for another file name during the export and import process, if required.
`-automatic_merge`	Automatically imports contents of the dump file

2. Transfer your export data. To do this:
   a. Run the script using `-mode export` as the option.

```
%MyScript.csh -mode export
```

This prompts you for information such as the schema name (source), password, dump file names, and so on. It also creates a dump file upon completion.

b. Finally, using FTP, transfer your dump file and the Export and Import script to the computer where your target OracleAS Portal schema resides.

10.2.3.1.3 Exporting Large Page Groups

You can use the `opeasst.csh` (Oracle Portal Export Assistant) script to export large page groups, which can time out in the browser while calculating the page group dependencies. These timeout issues are due to the Dependency Manager and the preliminary check routines that are run as foreground processes. The actual data extraction and the data merge are performed in the background.

The script can be found in the `/portal/admin/plsql/wwu` directory. The following is an example of the script:

```
%opeasst.csh
Usage: opeasst.csh <-s portal_schema> <-p portal_password> <-c connect_string>
<-ts transportset_name> <-pgrps pgrp_names> <[-export_acls]>
```

Table 10–5 provides a description of the parameters used in this process.

Table 10–5 OPEASST.CSH Parameter Descriptions

Parameters	Description
`-s portal_schema`	Oracle Database account for the portal.
`-p portal_password`	Oracle Database password for the portal.
`-c connect_string`	TNS connection information for the source database.
`-ts transportset_name`	Name of the transport set to be created.
`-pgrps pgrp_names`	Comma-delimited list of Page groups for export. **Note:** Exporting seeded page groups using the script is not allowed.
`-export_acls`	Export object-level privileges.

Perform the export from the command line, and then perform the following tasks:

1. Check the log in your transport set manager for any errors by clicking the **Status** link. See Section 10.2.3.3, "How Do I Manage My Transport Sets?" for more information about how to edit and browse the transport sets on the system.
2. When the export is complete browse your transport sets and select the appropriate script for your operating system. See Section 10.2.3.1.2, "Exporting Data" for details.
3. Run the script using `-mode export` as the option.

   ```
   %MyScript.csh -mode export
   ```

   This prompts you for information such as the schema name (source), password, dump file names, and so on. It also creates a dump file upon completion.
4. Using FTP, transfer your dump file and the export and import script to the computer where your target OracleAS Portal schema resides.

5. To import your objects, the contents of the transport set dump file must first be imported to the transport set tables on the target system. See Section 10.2.3.2.2, "Importing Data" for details.

The following features and limitations currently exist:

- The script supports only exporting page groups.
- Multiple page groups can be exported at once using comma-delimited values.
- Dependency Manager logs are available after export. These logs help identify data inconsistencies in the source, for example missing dependencies.
- Export Access Control Lists feature is supported.
- There is no import mode option available, that is, **Replace on Import** or **Reuse**.
- Exporting database providers is not supported.
- If the Dependency Manager results in some external objects for the page group being exported, then all the external objects are automatically made explicit by the script without any user intervention. Those objects that can be made explicit are recursively added to become part of the transport set until there are no remaining external objects in the transport set.
- The script name cannot be changed.

Notes:

- Remember to set the infrastructure Oracle home (*ORACLE_HOME*) when connecting to the database to run the `opeasst.csh` script.
- To run shell script tools on the Windows operating system, you need one of the following UNIX emulation utilities:
  - — Cygwin 1.3.2.2-1 or later. Visit `http://sources.redhat.com`.
  - — MKS Toolkit 6.1. Visit `http://www.datafocus.com/`.

10.2.3.2 How Does OracleAS Portal Import Work?

This section describes the import process and the steps required to successfully transfer content to the target portal system, including:

- Running Your Script on the Target System
- Importing Data

10.2.3.2.1 Running Your Script on the Target System To import your objects, the contents of the transport set dump file must first be imported to the transport set tables on the target system. This is done by calling the same script (used in the export) with the `-mode` parameter set to `import`. The parameters in bold are applicable only for import and are mandatory.

```
%MyScript.csh
Usage: MyScript.csh <-mode export_or_import> <-s portal_schema>
<-p portal_password> <-pu portal_username> <-pp portal_userpassword>
<-company company_name> <-c connect_string> <-d dump_file_names>
<-automatic_merge>
```

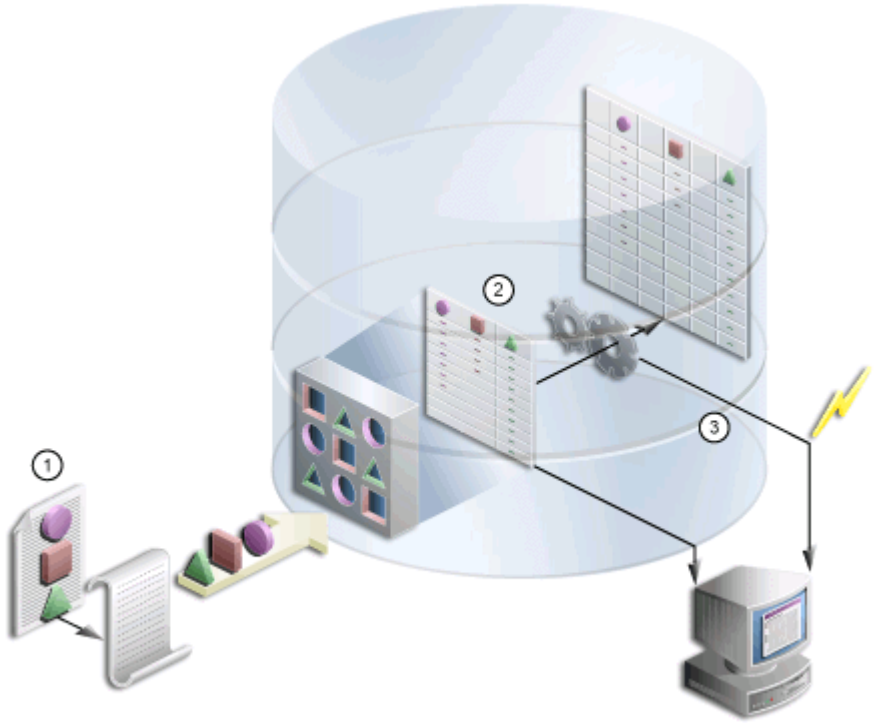
To perform the entire import from the command line, which initiates a background process, you must include the portal user name and password parameters. This is required to validate your role on the target portal instance.

Notes:

- Remember to set the infrastructure Oracle home (*ORACLE_HOME*) when running the import script.
- Before running the script using the `-automatic_merge` option, you must ensure that all external objects listed in the manifest exist in the target instance. External objects include database schema, tables, external applications, and so on. This information can be obtained by checking the external objects in the source instance.
- The value for the `company_name` parameter is the company name you see in the login page when working in a hosted portal. When working in a portal that is not hosted, the value for the parameter should be `none`. If you are running the script in interactive mode, then do not pass a value.

The contents of the dump files are imported, and the transport set is made available from the user interface for merging on the target portal system. Figure 10–9 shows how the import process works.

Figure 10–9 Import Process



1. You import the contents of the transport set dump file to the transport set tables utilizing the same script used in the export.
2. A background job is submitted to initiate the import, and log information is generated.

3. Once the import is complete, you can access the transport set from the user interface.

Notes: To preserve data integrity, avoid:

- Importing an object, changing its name, and then reimporting it.
- Importing an object, moving it to shared objects, and then reimporting it.
- Importing an object, and then moving it from one hierarchy to another.

10.2.3.2.2 Importing Data To import an object, the contents of the transport set must first be imported to the target system. When you select a transport set for import, a preliminary check process determines if the objects already exist on the target.

To import your content:

1. Locate the **Export/Import Transport Set** portlet, installed by default on the **Administer** tab.

Note: When you import a transport set and click the **Browse Transport Sets** link, you will see the newly imported transport set with the Export Complete status and links to the export scripts.

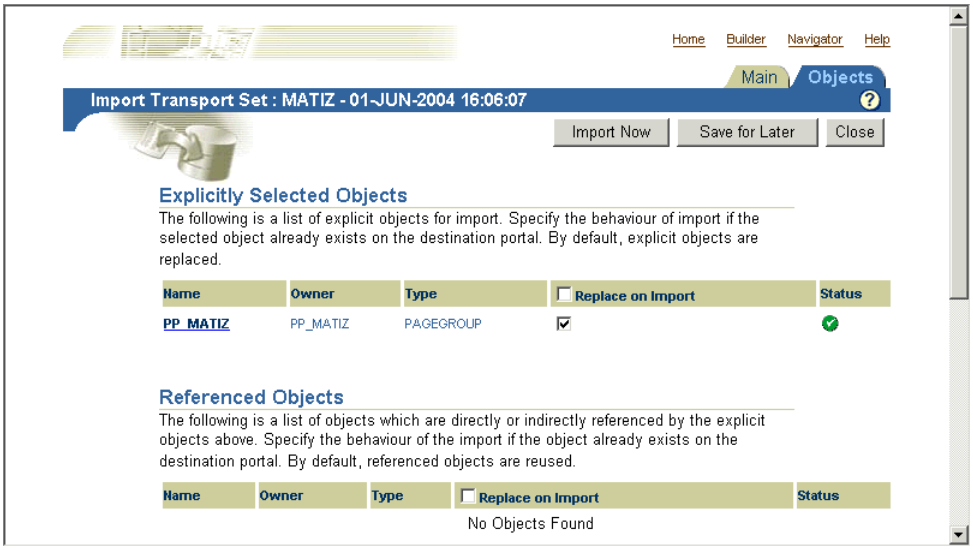
Selecting a transport set on the target for **Reuse** resets the transport set. This makes the transport unusable because it was not exported from the target instance and therefore no objects exist that match the objects in the transport set.

---

2. Select the imported transport set and click **Import.** The **Objects** page of the Import Manager is displayed.

   Figure 10–10 shows the **Objects** page that displays the list of objects included for import.

Figure 10–10 Transport Set Manager Import Objects



3. If you select **Replace on Import**, then the object is replaced if it is found in the target portal.

> **Note:** **Replace on Import** mode is the default mode for explicitly selected objects; **Reuse** is the default mode for referenced objects. The import modes are not applicable to the external objects until they are made explicitly selected objects.

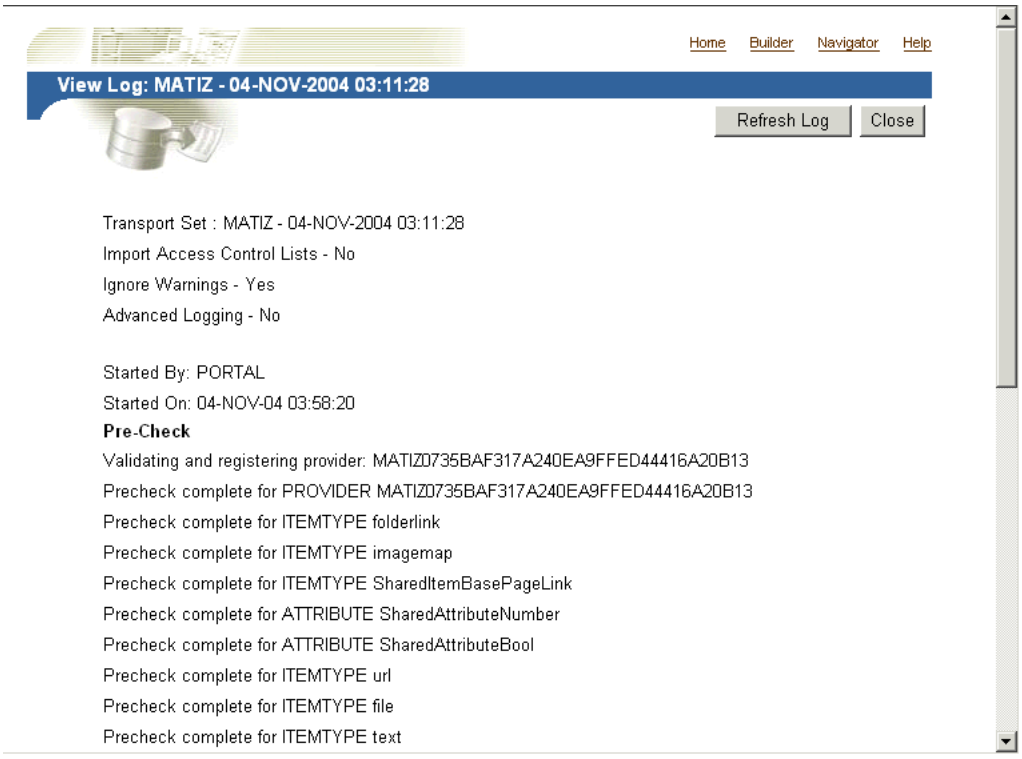
4. To view the log output, click the **Status** icon. Table 10–6 provides a description of each status type.

Table 10–6 Status Descriptions

Status	Description
	Pass
	Fail
	Pass with warnings

Figure 10–11 shows a sample **View Log** page.

Figure 10–11 Transport Set Manager Import Log

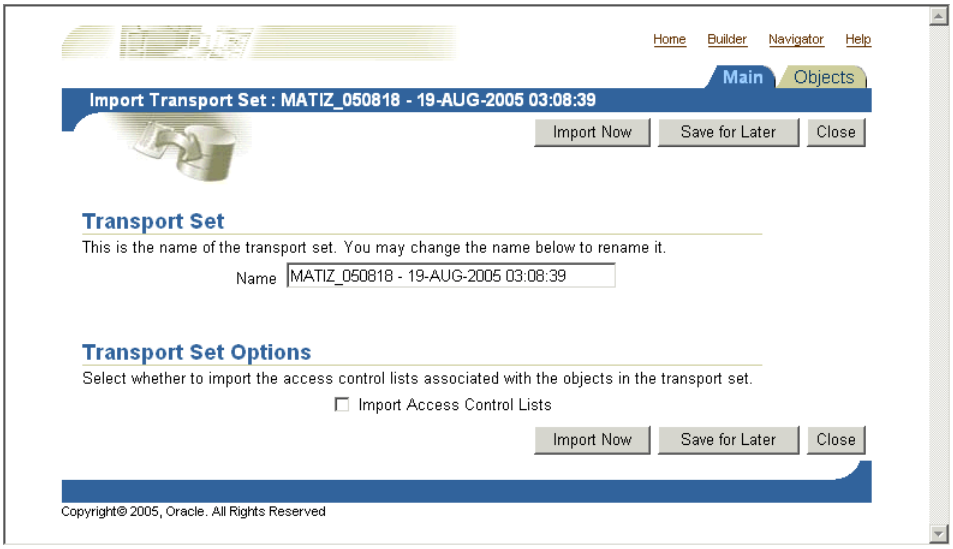


Note: To view a detailed log of the import process, including debug messages, perform the following steps:

1. In the **Transport Set Import Log** page (shown in Figure 10–11), go to the Address or Location bar in your browser, and search for the string `p_log_mode=0` in the URL.
2. Change `p_log_mode=0` to `p_log_mode=1`.
3. Press **Enter**.

5. Click **Close** to return to the **Objects** page.
6. Click the **Main** tab.

Figure 10–12 Import Transport Set Page



7. Select the **Import Access Control Lists** option under Transport Set Options if you want to include the access control lists (ACLs) associated with the objects in the transport set.

Note: The **Import Access Control Lists** option cannot be selected if you did not select it during the export process.

If you select this option, the following happens:

- User and group profiles get created only if they do not exist on the target.
- User and group profiles do not get updated upon subsequent imports. Default groups of users are not imported.
- If a user exists on the target, then the user's default group is populated from Oracle Internet Directory.

8. Select either **Import Now** if you are finished, or **Save for Later**.

When you select **Import Now**, the exported objects are imported in the background. Clicking **Save for Later** saves changes to the transport set for later resolution and import.

9. Check the log for errors.

   To ensure that all the content has been imported correctly:

   – In the Navigator, verify that the content in each portal page group that you imported was imported correctly. Specifically, for each portal page, verify that the appropriate portlets appear in each region of your portal page. When these portlets (navigation pages, pages exposed as portlets, database provider components, or Web portlets) occur as external dependencies and they do not exist on the target, then the portlet entry is deleted from the page.

   Note: During the import, a two-step preliminary check process is performed. Clicking **View Log** shows both the first stage of the process and the preliminary check as complete. This is done before the import and before populating the portal tables with data.

   Clicking **Refresh Log** will show both the second stage of the process and the preliminary check with different timestamps.

Warnings and Failures During Import

Objects that are being imported can be classified into two types:

- Warning types - Objects that, on failure, cascade warnings to explicitly selected objects.
- Failure types - Objects that, on failure, cascade failures to explicitly selected objects.

A warning type will raise warnings and allow the explicitly selected objects to be imported. A failure type object is not imported.

If an explicitly selected object has two dependencies, a warning type and a failure type, and if both the dependencies fail the preliminary check process, then the failure type will be dominating, and the explicitly selected object will fail.

A warning type affects explicitly selected objects more than any other kind of object. Referenced and external objects raise failures and warnings for the explicitly selected objects based on their type. Table 10–7 describes the warning or failure behavior for each object.

Table 10–7 Warning and Failure Types

Object	Type	Expected Behavior
Attribute	Failure	The explicitly selected object will fail if the dependent attribute fails.
Item type	Failure	The explicitly selected object will fail if the dependent item type fails.
Page type	Failure	The explicitly selected object will fail if the dependent page type fails.
Style	Warning	The style will revert to the main style of the page group to which it belongs.
Category	Warning	The category is set to none.
Perspective	Warning	The perspective associated with an item or page is removed.
Portal Templates for pages	Failure	The explicitly selected object will fail if the dependent template fails.

Table 10–7 (Cont.) Warning and Failure Types

Object	Type	Expected Behavior
Portal Templates for items	Warning	The Portal Template for item associated with an item or page is removed.
HTML Template	Warning	The HTML Template associated with an item or page is removed.
Page	Warning	There are three possible outcomes when a page is a dependent of another object: ■ **Page exposed as a portlet**. The portlet entry is removed from the region that contained the page portlet. ■ **Page link pointing to a page**. The page link is removed from the region, because the page to which the link is pointing to has failed. ■ **Page Parameters and Events dependency**. The link that was pointing to the page that failed is reset to point to the same page in which the Page Parameters and Events link is located.
Navigation page	Warning	The navigation page portlet is removed from the page. You can associate the page with another navigation page after the import.
Color, font, JavaScript, application template, image	Warning	Set to the default at run time.
Database provider component	Warning	The portlet entry where the component is placed is deleted from the page.

When the container objects in the following list appear as external dependencies, because their child objects were selected for export and they do not exist on the target, the explicitly selected objects (child objects of the container objects) will fail.

- Page group
- Portal DB Provider
- Category
- Perspective
- Page

Cascade Warning Behavior

Warnings or failures detected in objects during the preliminary check behave as shown in Table 10–8.

Table 10–8 Cascade Warning Behavior

Object	Warning Status	Failure Status
Contained object	Status is cascaded to the container object.	Status is cascaded to the container object.
Hierarchical object	■ Status is cascaded to all parent objects. ■ Status is **not** cascaded to child objects.	■ Status is cascaded to all child objects. ■ Status is cascaded to all parent objects.

Table 10–8 (Cont.) Cascade Warning Behavior

Object	Warning Status	Failure Status
Referred object	Status is **not** cascaded to all referenced objects.	Status is cascaded to all referenced objects.

Portlet Cleanup

Imported portlets are synchronized with the target portlet repository during the import process. If a portlet instance fails during the resolution phase of the import process, then it is deleted from the source page.

For example, a page can have a portlet, which could be one of the following:

- Navigation page
- Page exposed as a portlet
- Portal DB Provider component
- Web portlet

When these portlets appear as external dependencies in Reuse mode and do not exist on the target page, the portlet instance is deleted from the page. If these dependencies were made explicit and their import failed, then the portlet instances would still be deleted.

To summarize, if the imported portlet does not exist in the portlet repository on the target, then it gets deleted from the source page.

Note: The portlet cleanup operation deletes portlet dependencies such as Page Parameters and Events, URL, text, and so on. The page structure remains unchanged after removing the portlet instance from a source page.

If the navigation page (external dependency) does not exist on the target page, then a page using that navigation page passes with warnings, and the navigation page portlet gets deleted from the source page.

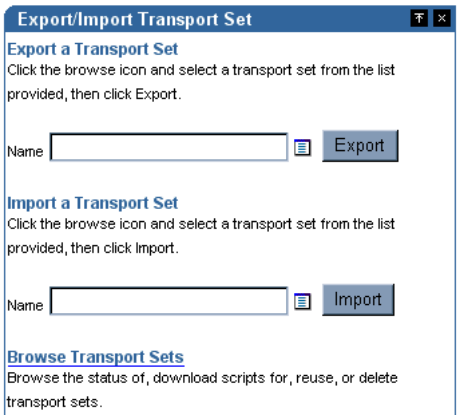
10.2.3.3 How Do I Manage My Transport Sets?

The **Export/Import Transport Set** portlet is installed by default on the **Administer** tab and enables you to export, import, edit, and browse the transport sets on the system. This section discusses the following:

- Editing Transport Sets
- Browsing Transport Sets

Figure 10–13 shows the **Export/Import Transport Set** portlet.

Figure 10–13 Export/Import Transport Set Portlet



10.2.3.3.1 Editing Transport Sets You can view and edit the list of objects selected for a transport set. Once you create a new transport set and select the **Save for Later** option:

1. Navigate to the **Export/Import Transport Set** portlet.
2. Select the transport set from the export list of values.
3. Edit the preferences.

10.2.3.3.2 Browsing Transport Sets You can view all of the transport sets that are on the system and their current status. You can also view the log of actions, view the referenced objects, and download the export and import scripts. Additionally, you can delete transport sets from the system, and you can reuse a transport set by selecting the transport set and clicking **Reuse**.

When you select transport sets and click **Delete**, you are prompted for confirmation. Clicking **OK** does not affect transport sets that are in the *Export*, *Export In Progress*, *Precheck In Progress*, *Migration In Progress*, *Import*, or *Import In Progress* statuses.

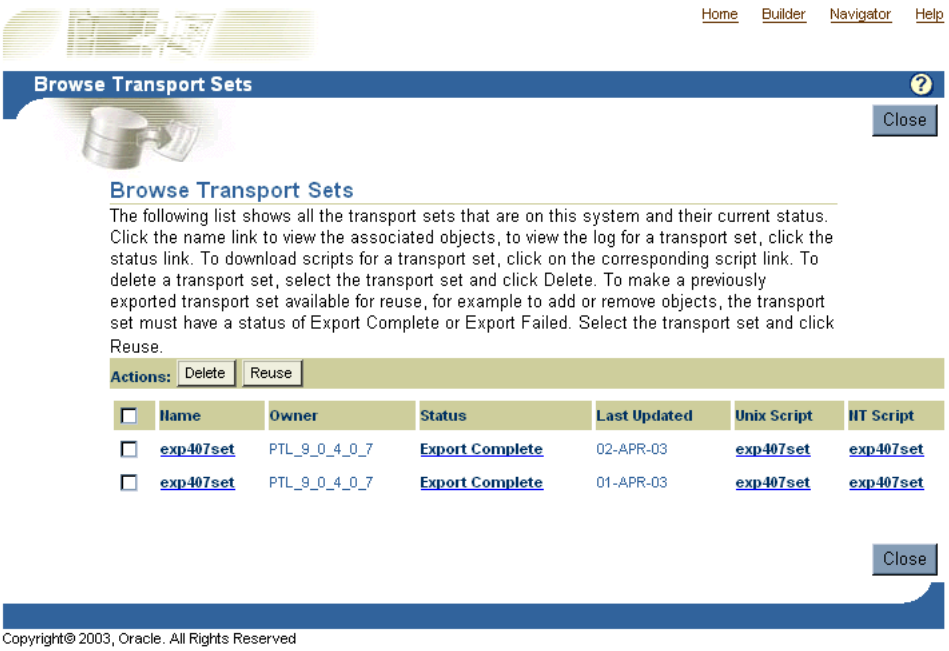
When you select transport sets and click **Reuse**, you are prompted for confirmation. Clicking **OK** does not affect transport sets that are in the *Export*, *Export In Progress*, *Migration In Progress*, *Ready For Import*, *Import*, *Import In Progress* statuses.

Notes:

- The **Reuse** option is valid only for transport sets in the source portal with a status of **Export Complete** or **Export Failed**.
- You can import objects with multiple hierarchies in the same transport set.

Figure 10–14 shows a sample **Browse Transport Sets** screen.

Figure 10–14 Browse Transport Sets



10.2.4 OracleAS Portal Export and Import - Alternate Method

You can export and import content when both the source and target OracleAS Portal instances exist in a customer database installation, and not in a product metadata repository. For details, refer to the information on staging a test environment from a production environment, in the *Oracle Application Server Administrator's Guide*.

10.3 Behavior of Objects After Migration

The following considerations should be made before migrating portal content from a source instance to a target instance using OracleAS Portal Export and Import. This section discusses the behavior of OracleAS Portal objects after migration.

Import of Translations

Import of translations in Overwrite mode will not be a strict overwrite, and will act as if the translations are being merged. Any unwanted translations on the target, which do not exist on the source, are not removed when the page group is imported in Overwrite mode. You can remove the unwanted translations after the import. However, new translations brought from the source will be imported. This behavior is true for translations of all relevant objects in the subsequent tables.

This section contains the following subsections:

10.3.1 Behavior of OracleAS Portal Objects

This section discusses the behavior of the following portal objects after migration:

- Page Groups
- Attributes
- Approvals
- Items
- Pages
- Regions
- Portal Templates
- HTML Templates
- Categories
- Perspectives
- Navigation Pages
- Styles
- Item Types
- Page Types

10.3.1.1 Page Groups

On the first export and import, if a page group does not exist, then it is created on your target system. Any settings at the page group level are replicated on the target system. On the second import, depending on the mode selected:

Replace on Import mode. The page group properties from the source replace those on the target. All objects within the page group are created or updated depending on whether or not they existed.

Reuse mode. When page groups already exist on the target, the properties are reused and not updated. New objects within the page group are created; existing objects are reused.

Notes:

- New pages are currently not created when page groups are imported using Reuse mode.
- The order of visible objects (in the **Configure** tab) may differ between the source and target portal. The result is that the drop-down lists (when selecting an item, category, and so on) will look different in the target portal. You can manually reorder the visible objects in the target.
- All configurable settings of a page group are reused and overwritten appropriately in the **Configure** tab (found when you click **Properties** for a page group).
- If a page group is imported with a different name, then a new page group is created on the target.
- Migration of the **Shared Objects** page group excludes pages that cannot be edited or exported, for example, the A to Z root pages.

10.3.1.2 Attributes

On the first export and import, the attributes are created on the target system**.** The second import, depending on the mode selected for your target:

Replace on Import mode. The properties of the attribute are updated.

Reuse mode. When the attribute already exists on the target, it is reused and not updated.

Notes:

- Attributes that are marked as external cannot be created on the target, even with **Any Transport Set - Manage** privilege.
- Attributes on the source and the target can only be considered the same when they have the same name, are the same type and have the same unique internal identifier. If the two attributes have the same unique internal identifier but different names, then they can be only imported in Replace on Import mode. If the name and the type are the same, but the unique internal identifier is different, then the attribute import will fail and cascade to any other related objects.

10.3.1.3 Approvals

To view the approvers, access control lists must be exported and imported along with the page group or page that has an approval defined on it.

Replace on Import mode. The approval process can be established for a page or page group. If a page group or a page is marked for either insert or update, then the approval object will be processed in Replace on Import mode. All the information in the target will be deleted and re-created.

Reuse mode. No action is performed.

10.3.1.4 Items

Item information comes as a part of page export. They follow the import mode of the page.

Replace on Import mode. When a page is imported in Replace on Import mode, items in page regions from the source are copied to the target. Any items found only on the target are removed, items that exist on both the source and target are updated, and items that exist only on the source are created.

Reuse Mode. No items are imported from the source. The page from the source is only used as a reference, and will determine the import mode of items.

Notes:

- The schema associated with a PL/SQL item, page, or attribute, is extracted only if it is not a Public schema or a Creator schema. Such a schema is marked as an external object. The schema needs to be present on the target database to avoid a preliminary check failure. However, you can proceed with the import. The logs will show appropriate messages indicating that it will result in run time errors that can be corrected by bringing in the schema later and reassociating it.
- The list of object items will show differently between source and target unless you migrate those referenced objects (pages, categories, and perspectives) within the same transport set as the list of objects. Note that the Dependency Manager will not mark the objects referenced in the list of objects for export. For this reason, you need to explicitly mark those referenced objects for export, or ensure that they are already in the transport set.

- If portlet instance items are moved from one region to another between subsequent imports of the same page, then any personalizations made by users on those portlet instances are removed.
- Items for pages based on a template are synchronized, in Overwrite mode.
- All explicitly checked-out items in an active state are made checked-in after import.

10.3.1.5 Pages

Exports the page and the page type, template, and style it references along with content (item and portlets).

Replace on Import mode. The properties of the page are replaced. See Section 10.3.1.6, "Regions" for region import behavior. See Section 10.3.1.4, "Items" for item behavior.

Reuse mode. The original page on the target is reused. Child objects are not created on the target (if they do not already exist).

Refer to Table 10–9, " Import Behavior of Regions in Overwrite Mode", for information on import behavior when a page is imported in Overwrite mode.

Notes:

- The current release does not support locking and unlocking content using WebDAV. Content contributors can lock a file, which in turn will check out the item. On import, no owned locks will be displayed.
- When a page exposed as a portlet appears in the external objects list, make sure to include the page in the transport set.

10.3.1.6 Regions

Region information comes as part of page export. They follow the import mode of the page.

Replace on Import mode. When a page is imported in Replace on Import mode, page regions from the source are copied to the target. Any regions found only on the target are removed, including all content in those regions.

Reuse Mode. No regions or items are imported from the source. The page from the source is only used as a reference, it will determine the import mode of regions.

Note: This release of OracleAS Portal implements synchronization of target regions with the source. See Table 10–9, " Import Behavior of Regions in Overwrite Mode" for more information.

Synchronization of Regions

This release of OracleAS Portal implements synchronization of target regions with the source. The import behavior when a page is imported in Overwrite mode is described in Table 10–9.

Table 10–9 Import Behavior of Regions in Overwrite Mode

Case	Source	Target	import Behavior
Synchronization of target regions with the source	Region_A Region_B Region_D	Region_A Region_C Region_D Region_E	■ The attributes of Region_A and Region_D are updated with the properties from the source. ■ Region_B is not found on the target and will be created. ■ Region_C and Region_E, which exist only on the target, are deleted.
Region delete from target	-	-	When a region is deleted from the target, all the items and portlets, including user personalizations, are deleted from the target.
Root region mismatch for a page **Note:** A page can only have one root region.	Root region – Region_X	Root region – Region_Y	The entire root Region_Y hierarchy is deleted from the target and re-created with the Region_X hierarchy from the source.
Region type mismatch **Note:** Available region types are item, portlet, tab, and subpage.	Region_X – Type A	Region_X – Type B	When there is a region type mismatch, all the items and portlets under that region (including user personalizations) are removed from the target and re-created with the items from the source region.
Region type match	Region_X – Type A	Region_X – Type A	The target items are synchronized with the source items for that region.

Table 10–9 (Cont.) Import Behavior of Regions in Overwrite Mode

Case	Source	Target	import Behavior
Synchronization of target items with source **Note:** This happens whenever the source and target region type matches.	Item_A Item_B Item_D	Item_A (base user) Item_A (personalized for User A) Item_C (base user) Item_D (base user) Item_E (personalized for User B)	▪ Item_A (base user) is overwritten. ▪ Item_A (User A personalization) is preserved on the target. ▪ Item_B is created on the target. ▪ Item_C (base user) is deleted from the source. ▪ Item_D (base user) is overwritten. ▪ Item_E (User B personalization) is preserved on the target. **Note:** Although Item_E does not exist in the source, it is not deleted from the target because it is a user personalization on the target. Only base user item records are part of the structure of a page, and are shown when a page is edited.

10.3.1.7 Portal Templates

Exports the template, the style it references, and any content on the template. The layout and content of pages that depend upon the template are synchronized with the revised template on the target.

Replace on Import mode. The template properties are replaced on import.

Reuse mode. Template information is reused on the target and is not updated from the settings on the source system.

Notes:

- Do not export or import the Category Pages Template or Perspective Pages Template found in the shared objects or page group. They are present only if a category or perspective is created in that page group.
- A template can force all pages based on the template to use the template's style, or it can allow pages based on it to have their own styles. When importing a template whose style has changed, the changes are only propagated to the pages based on the template, if the template forces the pages to use the template's style.
- Templates that were modified after the last import cannot be reused. If you try to reuse a modified template, then the template will fail the preliminary check stage along with the pages in the transport set that are based on the template. Appropriate messages are logged in the preliminary check logs indicating that you have to mark the template in Overwrite mode to proceed with the import.
- When a page or an item that uses Portal Templates for Items is migrated, the Portal Templates for Items are brought in as dependencies in the target.

Caution:

Region and Item movements done on a template-based page are lost if the template is imported in Overwrite mode. This is also true for:

- Items and portlets that are hidden or deleted on the template-based page.
- Tabs that are moved or deleted on template pages.

This is because templates always take precedence over pages based on them. Only changes that are specific to the page are retained.

As a workaround, you can perform the following high-level steps:

1. Migrate the updated template in Overwrite mode.
2. Migrate the pages that contain modified template items and portlets, in Overwrite mode. In this case, import the template in Reuse mode for the changes to be preserved.

This workaround is valid only for pages migrated in a transport set, and ensures that the modifications made to the items and portlets in the template-based pages are preserved. This procedure needs to be performed every time the template is imported in Overwrite mode.

10.3.1.8 HTML Templates

On the first export and import, the HTML Templates are created on the target system. On the second import, depending on the mode selected for your target:

Replace on Import mode. The properties of the HTML Template are updated.

Reuse mode. If the HTML Template already exists on the target, then it is reused and not updated.

10.3.1.9 Categories

Exports the category and its subcategories.

Reuse mode. The original category on the target is reused. Child objects are not created on the target (if they do not already exist).

Notes:

- The category page (the page that appears when a category is clicked) and the category template are not exported. They are created each time on import. The category is always reused; therefore, you make changes once on the target, and the changes will not be lost during subsequent imports. This applies to the category, the category page, and the category template.
- There is no Replace on Import mode. The Replace on Import option will not apply; the category will always be reused.

10.3.1.10 Perspectives

Exports the perspective and its subperspectives.

Reuse mode. The original perspective on the target is reused. Child objects are not created on the target (if they do not already exist).

Notes:

- There is no Replace on Import mode. The Replace on Import option will not apply; the perspective will always be reused.
- The perspective page (the page that appears when a perspective is clicked) and the perspective template are not exported. They are created each time on import. The

perspective is always reused; therefore, you make changes once on the target, and the changes will not be lost during subsequent imports. This applies to the perspective, the perspective page, and the perspective template.

10.3.1.11 Navigation Pages

Exports the navigation page, the style it references, and any links on the navigation page.

Replace on Import mode. The properties of the navigation page are replaced.

Reuse mode. The original navigation page on the target is reused.

10.3.1.12 Styles

Exports the style.

Replace on Import mode. The properties of the style are replaced.

Reuse mode. The style on the target is reused.

Notes:

- Styles on the source and the target are considered the same when they have the same name and the same unique internal identifier. If the two styles have the same unique internal identifier, but different names, then they can be only imported in Replace on Import mode.
- Attributes associated with styles are not imported. A local style is associated with all the local attributes of the page group to which the style belongs, and all the shared attributes. A shared style is associated with all the shared attributes.

10.3.1.13 Item Types

Exports the item type and the attributes it references.

Editable seeded item types present in all portal instances are extracted.

Notes:

- If you have to modify a seeded item type, then Oracle recommends you make a copy of the seeded item type, and then modify the properties of the copy.
- Item types on the source and the target are considered the same when they have the same name, are the same type, and have the same unique internal identifier. If the item types on the source and the target have same unique internal identifier, but different names, then they can only be imported in Replace on Import mode.
- Currently, when the attributes associated with the custom types (item type, page type) are modified or the functions associated with the custom type are modified between imports, the changes are not always correctly migrated. You should delete and re-create the custom type on the target. This results in all the items and pages (based on the custom type) being deleted.
- If an item link in a page points to an item on another page, then during export of the page containing the item link, the page containing the linked object is brought in as a dependent page.

10.3.1.14 Page Types

Exports the page type and the attributes it references.

Notes:

- Page Types on the source and the target can only be considered the same when they have the same name, are the same type and have the same unique internal identifier. If the page types on the source and the target have same unique internal identifier but different names then they can only be imported in Replace on Import mode.
- Currently, when the attributes associated with the custom types (item type, page type) are modified or the functions associated with the custom type are modified between imports, the changes are not always correctly migrated. You should delete and re-create the custom type on the target. This will result in all the items/pages (based on the custom type) being deleted.

10.3.2 Import Behavior of Child Objects

This section describes the functioning of child objects after migration. Table 10–10 describes the behavior in detail.

Table 10–10 Import Behavior of Child Objects

Name of Object	Objects	Import Behavior
Contained objects, which contribute to the structure of the object	▪ Regions ▪ Items ▪ Tabs and subtabs on a page	▪ Contained objects are created or overwritten when the contained object is created or overwritten. ▪ When container objects are reused for the target, none of the contained objects will be created from the transport set, even if they do not exist on the target.
Contained objects that do not contribute to the structure of the object, but act as placeholders within a container.	▪ Attribute, Style, Category, Perspective, Item Type, Page Type, Page, and so on, in a page group. ▪ Form, Report, Chart, Dynamic Page, and so on, in a Portal DB Provider.	▪ Contained objects are created when the container object exists on the target, or are created from the transport set. ▪ When container objects are reused for the target, only new contained objects will be created from the transport set. All the existing objects will be left untouched on the target.
Child objects	▪ Subpage ▪ Subcategory and subperspective	▪ Child objects are created when the parent object exists on the target, or are created from the transport set.

10.3.3 Behavior of DB Provider Objects

This section describes the behavior of the following DB Provider objects after migration:

- Seeded DB Providers
- Portal DB Providers
- Portal DB Provider Components
- Shared Components
- Registered Database Providers

10.3.3.1 Seeded DB Providers

- When a page with Develop-in-Place portlets is imported, the components related to those portlets are automatically created in the database schema of the target portal's Develop-in-Place provider.

  The name of the Develop-in-Place database provider is **PTL_TOOLS_APP**. The underlying database schema for PTL_TOOLS_APP is **<PortalSchema>_APP**.

- For other seeded database providers, the relevant components brought in from the source portal are automatically created in the database schema of the target portal's database provider, if the provider already exists on the target portal.

  You have to make the seeded database provider a part of the transport set, if it does not already exist on the target portal. Otherwise, you do not need to move it.

Notes:

- The Develop-in-Place provider cannot be exported or imported on a standalone basis, that is, the Develop-in-Place portlets have to exist on a page.
- The Develop-in-Place provider, unlike other database providers, does not show up as an external object in the UI manifest.
- When migrating any database provider, if the Develop-in-Place components or components from other database providers are getting their data from database objects in a schema other than the underlying schema for the database provider, then that database schema should also be exported and imported into the target portal in advance using the `exp` and `imp` utilities.

10.3.3.2 Portal DB Providers

On the first export and import, if a Portal DB Provider does not exist, then it is created on the target system.

- Portal DB Provider properties will be created on the target.
- Provider registration will be done for the newly created Portal DB Provider.

On the second import, depending on the mode selected for the target:

Replace on Import mode. The Portal DB Provider properties from the source replace those on the target. All components within the Portal DB Provider are created or updated depending on whether or not they exist.

Reuse mode. When a Portal DB Provider already exists on the target, the properties are reused and not updated. New components within the Portal DB Provider are created, and existing components are reused.

Note: If you are migrating a Portal DB Provider, then you need to perform the following tasks before importing the Portal DB Provider:

1. Ensure that the schema that is used by the Portal DB Provider being exported, exists in the target database instance and that the CONNECT and RESOURCE roles have been granted to it.
2. Run the `provsyns.sql` script (located in the *`MID_TIER_ORACLE_HOME`*`/portal/admin/plsql/wwc` directory) on the target. Using SQL*Plus, log in as the Portal schema owner and run the script from the SQL prompt, as follows:

   ```
   SQL> @provsyns.sql <db_provider_schema_name>
   ```

The `provsyns.sql` script can be executed multiple times for a Portal DB Provider schema.

10.3.3.3 Portal DB Provider Components

The following are the Portal DB Provider components:

- Menu
- Forms
- Reports
- Charts
- Calendars
- List of Values
- Link
- Hierarchies
- Dynamic Pages
- XML/URL Components
- Data Components

On the first export and import, the components are created on the target system.

- The first version of the component will be created under the nominated Portal DB Provider, and this will be the production version.
- A package will be created with the same name as the component under the schema associated with the Portal DB Provider.

On the second import, depending on the mode selected for the target:

Replace on Import mode. A new version of the component is created on top of any existing versions, and this will be the production version. Existing versions on the target, if any, will be archived. The package will be regenerated with the information obtained from the production version.

Reuse mode. If the component does not exist on the target, then it will be created.

Notes:

- List of Values and Link components do not have versions or a package associated with them. Therefore, these components are deleted and re-created on the target, in Overwrite mode.

- Because the List of Values and Link components cannot render on their own, or they are not in portlet form, there will not be any personalizations attached to these components.
- The List of Values (LOV) appears as an external object, which you can choose to make explicit. If an LOV does not exist on the target, then the import will proceed, and the logs will indicate that the LOV associated with the attribute was reset, and you could bring in the LOV and reassociate it later.

10.3.3.4 Shared Components

The following are the shared components:

- Color
- Font
- Image
- JavaScript
- UI Templates (Structured, Unstructured)

On the first export and import, if a shared component does not exist, then it is created on the target system.

On the second import, depending on the mode selected for the target:

Replace on Import mode. The shared components are deleted and re-created with the source information.

Reuse mode. When a shared component already exists on the target, the properties are reused and not updated. New shared components are created, and existing components are reused.

Note: System colors, fonts, and templates are reused on the target, and they are never exported and imported.

10.3.3.5 Registered Database Providers

The schema associated with a registered database provider is marked as an external object in the manifest. Note that on import:

- If the provider and the schema do not exist on the target, then the schema fails the preliminary check, which causes the provider to fail, in turn causing the explicit object to fail.
- If the provider exists and the schema differs on the source and the target, then the provider is assigned a warning status, and the logs will display that a difference in schemas exists.

Note: You must ensure that all the objects are valid after you migrate the schema from the source to the target, to avoid database registration errors.

10.3.4 Behavior of Portal DB Provider Reports Object Types

The Report Security Access Objects are always exported or imported as part of the Portal DB Provider export and import.

Notes:

- The granular export and import of Report Security Access Components are not supported.

- The Report Security Access Components behave in the same manner as DB Provider components in versioning.
- A package is created or regenerated for the Report Definition File (RDF) access component, similar to DB Provider Components.

10.3.5 Behavior of Web Providers

This section describes the following Web providers:

- OmniPortlet
- Web Clipping Providers, WSRP Producers, and Other Web Providers

Enabling and Disabling Export and Import of Web Providers

To enable or disable the migration of OmniPortlet and Web Clipping providers, edit the following variable in the *MID_TIER_ORACLE_HOME*\j2ee\OC4J_Portal\applications\portal\portal\WEB-INF\web.xml file:

```
<env-entry>
<env-entry-name>oracle/portal/provider/global/transportEnabled</env-entry-name>
   <env-entry-value>true</env-entry-value>
   <env-entry-type>java.lang.String</env-entry-type>
</env-entry>
```

Set the value to `false` to disable export and import of OmniPortlet and Web Clipping providers.

10.3.5.1 OmniPortlet

OmniPortlet providers, including their default personalizations and related information, referenced by your transport set will be exported and imported with the pages automatically.

Connection information (for example database, user name, password, URL, HTTP authentication user name and password, and so on) associated with an OmniPortlet instance is migrated automatically by default.

If you want to disable the exporting and importing of connection information because of security reasons, then edit the *MID_TIER_ORACLE_HOME*/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet/provider.xml file and set the `exportConnectionInfo` parameter to `false`. For example:

```
<provider class="oracle.webdb.reformlet.ReformletProvider">
   <exportConnectionInfo>false</exportConnectionInfo>
   ...
</provider>
```

If the connection information is not migrated, then the imported OmniPortlet uses the connection information of the same name on the target, if it exists. You can also enter the connection information of the imported OmniPortlet instance from the **Edit Defaults** page or the **Personalize** page.

If the connection information to be imported has the same name as an existing connection information of a provider in the target, then the source provider's connection information will not be imported unless the Overwrite mode is specified. Messages will be written to the transport log if the import of connection information failed.

Reuse mode. OmniPortlet providers are always reused.

Notes:

- If the provider registration generates an error due to insufficient privileges, then the provider object fails the preliminary check stage. This is then cascaded to the explicitly selected objects. A provider failing always fails the explicitly selected objects.
- Edit Default customizations are migrated. User personalizations are preserved on target, if present.

Important:

- If `localePersonalizationLevel` is different between the source OmniPortlet provider and target OmniPortlet provider, then some imported personalizations may become inaccessible in the imported pages. For example, if the current locale is Japanese, and if `localePersonalizationLevel` is set to `locale` on the source OmniPortlet provider and to `none` on the target OmniPortlet provider, then the Japanese personalizations will become inaccessible after importing.

  You can set `localePersonalizationLevel` in the `provider.xml` file located in the directory, *`MID_TIER_ORACLE_HOME`*`/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet`.

  For detailed information about `localePersonalizationLevel`, see the release note at *`MID_TIER_ORACLE_HOME`*`/portal/pdkjava/v2/pdkjava.v2.release.notes.html`.

- If OmniPortlet portlets are configured to use an SSL URL for fetching data, then you must copy these files manually, as SSL URL certificates are not exported and imported by default. Perform the following steps to manually copy the certificate files to the target instance:

  1. Append the SSL URL certificates to the certificate file used by the OmniPortlet provider (default is *`MID_TIER_ORACLE_HOME`*`/portal/conf/ca-bundle.crt`).
  2. Update the `<trustedCertificateLocation>` tag in OmniPortlet `provider.xml` file located in *`ORACLE_HOME`*`/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet/`.
  3. Restart OC4J.

10.3.5.2 Web Clipping Providers, WSRP Producers, and Other Web Providers

Web Clipping providers, WSRP producers, and other Web providers referenced by your transport set must either exist already on your target system or be able to be registered successfully during the import on your target system.

Reuse mode. Web Clipping providers, WSRP producers, and other Web providers are always reused.

Important: If Web Clipping portlets are configured to use SSL URLs for fetching data, then you must copy these files manually, as SSL URL certificates are not exported and imported by default. Perform the following steps to manually copy the certificate files to the target instance:

1. Append the SSL URL certificates to the certificate file used by the Web Clipping provider (default is *`MID_TIER_ORACLE_HOME`*`/portal/conf/ca-bundle.crt`).

2. Update the `<trustedCertificateLocation>` tag in the OmniPortlet `provider.xml` file located in *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portalTools/webClipping/WEB-INF/providers/webClipping/`.
3. Restart OC4J.

Notes:

- If the provider registration generates an error due to insufficient privileges, then the provider object fails the preliminary check stage. This is then cascaded to the explicitly selected objects. A provider failing always fails the explicitly selected objects.
- When WSRP portlets are imported, the portlet personalizations are not imported.

10.4 Recommended Best Practices When Exporting and Importing

The following is a summary of important recommendations and best practices developed for migrating portal content from a development or test environment to a production instance using OracleAS Portal Export and Import.

10.4.1 Naming Convention for Replicated Tabs

In earlier releases, the replicated tabs on the target had a different name from that on the source when the tabs were replicated on the pages based on the template. As a result, when you brought in the page at a later time, the tabs on the source did not match the ones on the target, and extra tabs were created on the target.

In this release of OracleAS Portal, having a predictable naming convention for replicated template tabs helps to avoid duplication of tabs. Because a page name has to be unique only in a hierarchy, the replicated tabs assume the same name as the template tab. However, you must ensure that you do not rename the replicated tab.

10.4.2 Migrating Page Groups and Components

Page groups and their associated components may be moved from development to production using the Export and Import utilities described in this document. In addition to page groups as a whole, individual components within page groups such as subpages, categories, perspectives, and page styles can be moved individually to the target system, only if the entire page group has been imported to the target system earlier.

- **Considerations and best practices to keep in mind**:

  - The first export to your target system migrates the entire page group from the source portal to the target portal instance. Subsequent transport sets can then export an individual page or other page group component on the target portal installation.

    Note: The preliminary check process will fail for an object if the page group does not exist on the target. Whenever a page group object is exported, the page group that owns the object is included as an external dependency. You can choose to make the page group explicit if you do not know if the page group exists on the target, and therefore avoid any potential preliminary check failures.

    The same applies to other objects included in a hierarchy. Categories, perspectives, and pages when exported display the parent category, perspective ,or page as an external dependency in addition to the page group to which they belong. All database provider components display the provider as an external dependent when they are exported by themselves.

    The default settings of a page group, for example, the default template, style, navigation page, and so on are also extracted by the Dependency Manager and classified as either reference or external (that is, local or shared).

  - All new or existing content on a page is replaced when a page with the same name is reimported to the target.
  - You can only move objects within a page group to the same page group of the same name on the target portal.
  - A page is migrated along with any subpages.
  - After an initial import operation to your target system, if you change the name of the page group on the target system, then subsequent import attempts to that page group will fail.
  - Categories, item types, perspectives, and page types that are configured in the source are not automatically configured in the target. You must explicitly configure these objects unless you are doing a page group export.

- **Page URL behavior:** Always use page link item types or path-based URLs when creating links to portal pages. Do not use *raw* portal page URLs.

  By default, portal page URLs generated by OracleAS Portal contain installation-specific ID numbers that change when the object is exported. This causes broken links when pages are imported into a different site.

  The following is an example of a URL generated for a page. If the page is imported on another site, then this page ID will change.

  ```
  http://my.portal.com/servlet/page?_pageid=47,49&_dad=portalr2&_schema=portal
  ```

  If you are using such URLs as manually entered links, then Oracle recommends you use path-based URLs or Page Link item types.

  The same page has the following path-based URL:

  ```
  http://my.portal.com/portal/pls/portal/url/PAGE/HRPAGEGROUP/HRHOME/HRBENEFITS
  ```

To find the path-based URL for a page, look at the page property sheet. A link to the property sheet can be displayed by adding a Property Sheet Smart Link item to the page.

You can also use a Page Link item type to create a link to a page. The Page Link item type dynamically generates the correct link at run time.

- **Page portlets**: When you replace a page, the content and the structure are replaced on the target.

Note:

- This release does not support importing and exporting the OracleAS Portal Survey components or the Favorites portlet. Any new Favorites or Groups added in the source will not show up in the transport set, nor will they be migrated to the target.
- This release supports exporting and importing generic page portlets. A generic page portlet can now be configured to point to any page. The page that the page portlet points to is marked by the Dependency Manager as *referenced/external* depending on whether or not it belongs to the same page group. On import, this information is resolved and stored in the preference store. On import, if the page does not exist on the target, then the portlet is reset.
- This release supports exporting and importing Web providers and their default personalizations. See the section on controlling the export and import of portlet personalizations in the *Oracle Application Server Portal Developer's Guide*.

To preserve the content in a page (items, portlets) on the target, but import a style layout, or for rendering changes from the source, you must expose your content through the **Federated Portal Adapter** portlet. The key is to separate your content from your page structure into two separate page groups. One for content only, exposed through the Federated Portal Adapter, and the other is your display page group. Users can use this to access, view, and customize their portal. Follow these steps:

1. On the source system, create a page group that only contains pages that have one region that you will later expose to other pages. This region is to be populated with either portlets or items. Name this page group Content Page Group.
2. Export this content page group to the target system.
3. On the target system, register the content page group through the Federated Portal Adapter. Expose these pages as portlets through the Federated Portal Adapter provider on the target system.
4. On the source system, register the same provider (using the same name as the Federated Portal Adapter provider).
5. On the source system, create another page group called Display Page. In this page group, create pages with regions that expose the portlets from the Federated Portal Adapter provider. You can also include tabs and other portlet regions in this page group if required.
6. Export the Display Page group to the target system.

7. From the target system, update, delete, modify, and add new items to the regions and pages in the content page group exposed through the Federated Portal Adapter provider.
8. On the source system, make changes to the page structure (tabs, new regions, and so on) to the Display Page page group.
9. Export the latest Display Page page group to the target system.
10. Verify that the Content Page Group contains the new changes that you made in Step 7, on the target environment.
11. Verify that the target system contains the latest changes to the pages in the Display Page page group that you recently changed.

Note: When a page containing a portlet from an adapter rendered provider (the loop-back case) is imported and the provider is automatically registered on the new portal, it will have the old URL, referencing the old portal.

When a loop-back provider is required in the new portal, you will have to create one or update the default provider.

- **Page and Portlet Personalizations and Edit Defaults Migration.** You can preserve the user customizations on a page or portlet on the target system while replacing or reusing the edit properties of that page or portlet.

Note: Personalizations for Web portlets are not currently preserved. Migration of Edit Defaults is supported for OmniPortlet and Web Clipping providers. If other providers implement this feature, their Edit Defaults will also be migrated. See the *Oracle Application Server Portal Developer's Guide* for information about how to implement this support.

Base objects that no longer exist on the page in the source portal will be removed from the target page after subsequent imports. This ensures that all personalizations for base portlet regions are also removed. Base objects are regions, portlets, items, and tabs that are imported as part of the core definition of the page, defining its structure and content.

Portlets that already exist on a page behave in the following way when the page is imported in Replace on Import mode:

- Edit Defaults will be migrated.
- User Personalizations will be preserved.

Properties of the page behave in the following way when the page is imported in Replace on Import mode:

- Edit Properties will be replaced.
- User Personalizations will be preserved, subject to the user customizations being valid.

Note: You can personalize, add, hide or show, delete, and move portlets and tabs. The page must have at least one portlet region and one tab (tab related customizations) in that region. The customized objects inherit the properties of the page. When a region is deleted, for example, a second import removes the region or tab from the page, then customized objects will also be deleted.

When you import the page with an increase in the number of portlets on a page, the source takes precedence even if you have customized the page in the target and deleted a portlet. The next time you import the same page, the deleted portlet is considered to be a new portlet to be added to the structure on the target. This also applies to tabs.

The order of appearance of these portlets (personalizations) and the portlets that form the content of the page are determined by the source and mode of import.

- **Replace on Import mode**. The portlets from the source are arranged in the order found in the source followed by the portlets in the target (personalizations).
- **Reuse Mode**. The personalizations are preserved, and there will be no changes to the target page.

10.4.3 Migrating Portal DB Providers and Components

Portal DB Providers and their associated components can be moved from a development environment to a production environment using the Export and Import utilities described in this chapter. In addition to Portal DB Providers as a whole, individual components within Portal DB Provider such as forms, reports, charts, and calendars can be moved individually to a target system. This is possible only if the entire Portal DB Provider was imported to the target system earlier.

Some considerations and best practices for migrating Portal DB Provider components are:

- Avoid using the portal schema for storing Portal DB Provider components, or the database objects that the components reference.
  - In the source environment, create a separate schema (referred to as the *portlets schema*) for the Portal DB Provider components. This is the schema that is referenced in the registration information when the Portal DB Provider is created.

    For more information, see the section "Creating a Schema in OracleAS Portal" in the *Oracle Application Server Portal Developer's Guide*.
  - In the source environment, create a separate schema (referred to as the *database objects schema*) for the database objects that the components reference. If the database objects already exist in a particular schema, then ensure that this schema is not referenced when creating the Portal DB Provider. This is the schema that holds database objects such as Tables, Views, or Procedures that are used in the creation of Portlet DB Provider components. For example, when you build a form based on a table, view, or a procedure, the table, view, or procedure is stored in the database objects schema.
  - Before importing the Portal DB Provider and its components, ensure that the database objects schema referenced by the components is available in the target environment. The database objects schema must have the same name as

in the source environment. Ensure that the database objects and database objects schema have the same grants and privileges as in the source environment. Also ensure that the status of all database objects is valid. The database objects schema can be exported or imported using the database's export or import utilities.

  - Before importing the Portal DB Provider and its components, create an empty portlets schema in the target environment with the same name as in the source environment.

- Ensure that the Portal DB Provider does not have any components that are in Edit or Archive mode. All components being exported should have only one valid production version to ensure that the target environment contains valid components after an import.
- If a page group contains portlets from a Portal DB Provider, then the provider has to be explicitly included in the transport set you are exporting. As an alternative, you can also export or import the provider earlier.
- The schema associated with registered Portal DB Providers is extracted as external object on the manifest.

Note: While importing a database objects schema, you must ensure that the ACLs (roles and privileges) associated with the schema already exist on the target system. This ensures that the generation of components, or the registration of database providers, does not fail during the import.

10.4.4 Migrating Search Components

There a number of options for adding search components to your pages. You can add a Basic Search to match search criteria entered into the Search field, an Advanced Search, and a Custom Search to create an automatically executed search.

10.4.4.1 Basic and Advanced Search Portlets

Basic Search portlets and Advanced Search portlets can be exported and imported. After import, the portlets should appear as they did in the source portal including the user preferences (if the user preferences were being imported).

10.4.4.2 Custom Search Portlets

Custom Search portlets can have many customizations which refer to other objects in the portal, such as page groups to search, attributes to search on, image on submission form, style for results, page for the results, attributes for the results, default values for category, perspective and item type attributes. These can be referred to as dependencies. When a custom search portlet is exported and imported, its dependencies are not automatically exported and imported. Therefore, it is possible that a custom search portlet is customized in the source but the dependencies do not exist in the target.

Also, a custom search portlet in the source may have been customized and then the dependency is removed from the portal and the custom search portlet's customizations are not updated. In this case, when the custom search portlet is used for a search, the missing reference is ignored. When the custom search portlet is customized again and the customizations saved the missing reference is removed.

On export, all the custom search portlets that were selected for export are checked and any missing references are removed. The customizations are then included in the transport set.

On import, a preliminary check determines if any dependencies are missing in the target after import. Messages are written to the log. For each custom search portlet that has missing dependencies, the log will show the reference path of the custom search portlet and the missing dependencies and what will happen on import.

The page on which the custom search portlet resides will be flagged with a warning. On the actual import, the custom search portlet customizations are modified to have the correct IDs of all the same dependencies in the target, and the customizations are copied into the target.

Note: Search results saved using the Saved Searches portlet are not imported or exported. You should submit the same search in the new target and save the latest set of search results.

10.4.5 Migrating Content Between Upgraded OracleAS Portal Instances

Export and import is not supported between two portals that are upgraded from releases earlier than 9.0.2. For example, assume that you have a source development portal instance and a target production portal instance, both of release 3.0.9. You then upgrade both the instances independently to release 9.0.4, and then to release 10.1.2. Exporting and importing content between these two upgraded 10.1.2 development and production instances is not supported.

During an upgrade from a pre-9.0.2 release of OracleAS Portal, objects (styles, attributes, item types, and page types) are given a new Global Unique Identifier (GUID). If the GUIDs do not match between objects in two OracleAS Portal instances, then the preliminary check for these objects will fail. If, for example, you have a source development instance and a target production instance, then you must resynchronize the OracleAS Portal instances to avoid preliminary check failures. To do this, perform the following steps:

1. Create an empty portal instance that will become the new source development instance.
2. Export the contents of the target production portal instance.
3. Import the contents into the new source development portal instance.

You have now exported and imported the contents from your target production portal instance to the new source development portal instance.

References to seeded page group objects, such as Top Level Pages and Design-Time Pages, will not resolve to the correct GUIDs across two instances. Remove these references from the objects you are exporting. Alternatively, you can create new objects that copy the functionality of the seeded page group objects.

Caution: Any new components in the development instance are lost during the re-creation of the development portal instance. Migrate all the new components from the development instance to the production instance before you upgrade the production instance. If you have partially developed components, then you must re-create these after the new development portal instance is created.

See Also: Section 10.2.4, "OracleAS Portal Export and Import - Alternate Method"

10.4.6 Exporting and Importing in a Hosted Environment

OracleAS Portal Export and Import supports the creation of classified content that can be used for replicating content and structure for new subscriptions. It does this by letting the portal instance set the subscription information during the import of the transport set contents into system tables. This means that in a hosted environment, you can export from any subscription, and you can import into any other subscription. This import is not limited to just one subscription; you can import the contents of the same transport set into multiple subscriptions, as follows:

1. Run the command-line utility in import mode.
2. Log in to a subscription.
3. Import the contents of the transport set into the subscription.

Example 10–1 shows a scenario where OracleAS Portal Export and Import can be used to import the contents of a transport set into multiple subscriptions.

Example 10–1 Importing Content into Multiple Subscriptions

1. Create a default seed subscription where the objects will be created and managed.

   In this subscription, you create a classified content and structure, which could consist of page groups, pages, other page group objects, Portal DB Providers and their components (exposed as portlets in the pages), portlets from Web providers, and so on.
2. Export the content and structure to a transport set, which becomes the seed transport set.
3. Export the contents of the transport set to a dump file.
4. Create a new subscription with the same structure and content defined earlier, by performing the following steps:
   a. Create the new subscription.
   b. Import the contents of the dump file into the portal instance.
   c. Log in to the new subscription.
   d. From the **Transport Set** portlet, select the transport set and import it.
   e. Verify that the new subscription now contains the required structure and content.
5. Repeat the previous step for each new subscription that you want to be based on the structure and content created in step 1.

This procedure can be used to create multiple taxonomic categories by creating transport sets for each category, and following the preceding procedure to populate new subscriptions.

Note: In a hosted environment with multiple subscribers, you cannot secure transport sets to a specific subscription in OracleAS Portal. If you created a transport set for export and import, then any other user who logs in to OracleAS Portal will be able to view the contents of the transport set that you created, in all subscriptions in that portal.

10.4.7 Importing Data with Oracle Text Index Synchronization Turned Off

While importing large data sets into a target OracleAS Portal instance, it is sometimes observed that the import process takes a longer time than normal, if synchronization of Oracle Text indexes is enabled. The import process is faster if you disable the synchronization of text indexes for the period of the import. To disable the synchronization of Oracle Text indexes, perform the following steps:

1. Before you start the import process, run the following command in the target OracleAS Portal instance as the portal schema owner (PORTAL):

   ```
   @textjsub.sql STOP
   ```

   Refer to Section 8.3.5.4, "Scheduling Index Synchronization" for details on scheduling, starting, and stopping text index synchronization.

2. Ensure that the `wwv_context.sync` job does not exist on the `dba_jobs` table.
3. Import your data set. See Section 10.2.3.2.2, "Importing Data" for more details.
4. Run the `textjsub.sql` script as the portal schema owner (PORTAL):

   ```
   @textjsub.sql START
   ```

5. Optionally, run the command to synchronize Oracle Text indexes. Refer to Section 8.3.5.1, "Synchronizing Oracle Text Indexes" for the procedure to do this.

10.4.8 Migrating Users and Groups

Oracle recommends the following procedure for exporting and importing:

- Develop your portal objects (page groups, content, portlets, and so on) on your source development system.
- To simplify the task of exporting and importing, assign users, groups, and privileges only on your production system.
- Use Export and Import to migrate your portal objects to your target production system.
- Apply users and privileges to imported portal objects as needed.

Users and groups are defined in Oracle Internet Directory. When you choose to include access control lists and User and Group preferences during OracleAS Portal Export, the user and group profiles held in the portal schema are included in the transport set. However, this does not migrate the user and group definitions that are held in Oracle Internet Directory.

For the user and group profiles to be properly imported on the target portal, the user and groups that they refer to must exist in the target portal's associated Oracle Internet Directory.

If you are building your portal content on a test or development server, with the intention to then move that content to a production server, you have the option of assigning your security privileges on the test server and then migrating them, along with the content, to your production server.

In this scenario, assign the privileges to groups, so there is no need to ensure the consistency of the user population between the test and production infrastructures.

If you want to precisely model your user population on both the production and test servers, the best approach is to use Oracle Directory Integration and Provisioning capabilities to synchronize the data from the production directory server to the test server. Synchronizing the data from production to test also provides you the option of

adding test users and groups to the test Oracle Internet Directory server without affecting the production server.

> **Note:** See the *Oracle Internet Directory Administrator's Guide* for more information on setting up directory synchronization. Note that it is advisable to automatically synchronize the data from production to test, but not the other way around.
>
> The *Oracle Internet Directory Administrator's Guide* can also be referred for additional information on migrating users and groups.

With the production groups also present on the test server, you can model and test all your access privileges on the test server and then safely migrate the portal access control lists with your exported objects onto the production system.

If you are introducing new groups and access privileges for those groups on the test system, then before you move the portal content and access control lists to production, make sure you migrate the group definitions to production first. You can actually create the groups on production first, and let the synchronization process reflect the new group on the test system before applying the test access control entries, if you need to actually create the group on the test instance first, you can create the group on production with the same means you used to generate the group on test. If this was done manually, and you want to avoid repeating the manual step on production, you can issue an LDAP query on the test instance to generate an LDIF file, which you can then load onto the production instance. For example:

```
%ldapsearch -h testoid.domain.com -p 389 -D cn=orcladmin -w password123 -b
'cn=portal.iasdb.domain.com,cn=groups,dc=domain,dc=com' -s sub -L 'cn=groupname' > newgroup.ldif
```

> **Note:** Before loading the LDIF file containing the group information into the production Oracle Internet Directory instance, you may need to edit the file to correct the portal instance name to match the name for that portal instance on the production Oracle Internet Directory instance. This name will typically be different between the test and the production instances and the name is part of the group DN, so it will have to be modified before loading the file.

In this example, `cn=portal.iasdb.dbserver.domain.dcom, cn=groups, dc=us, dc=oracle, dc=com` is the location under which the portal groups are located. Refer to Chapter 6, "Securing OracleAS Portal" for more information on the organization of the entries in the Directory Information Tree in Oracle Internet Directory. This creates a file called `newgroup.ldif` containing the group definition. You can then load the file on the production Oracle Internet Directory instance by using `ldapadd`:

```
%ldapadd -h prodoid.domain.com -p 389 -D cn=orcladmin -w password123 -v -f
newgroup.ldif
```

You may only want to deploy default privileges granted to some of the seeded portal groups, or no privileges at all. If no privileges are deployed, then the user performing the import will own the objects. The user can then further grant privileges on the target system as necessary for the specific deployment.

There is no need to synchronize seeded groups or users, assuming that, if privileges are granted to seeded groups in Portal, and those seeded groups are still present on the target system, then the privileges will be correctly associated with those seeded groups.

When migrating group profiles from the source to the target, the import will remap the DNs of the groups to the local group base on the target system if the exported profile was one for a local group on the source. A local group is one that is under the portal group container (the group install base). For groups that were not under the group install base, the DN will remain unchanged.

Note: The `ssoexp` and `ssoimp` scripts found in the `wwu` directory are obsolete for Oracle Application Server 9.0.x and not compatible with the 9.0.x login server. These should not be used.

11

Using the Federated Portal Adapter

This chapter provides information about the Federated Portal Adapter, previously known as the "PL/SQL HTTP Adapter". It describes how it can be used to share portlets with other OracleAS Portal instances.

This chapter contains the following sections:

11.1 About the Federated Portal Adapter

In this section, we will describe the following:

11.1.1 Overview

The Federated Portal Adapter is a component of OracleAS Portal that allows OracleAS Portal instances to share their database portlets through the Web portlet interface. It is a tool that uses SOAP and HTTP to distribute database providers across database servers. The Federated Portal Adapter allows database providers to be accessed as though they were Web providers.

In earlier releases of OracleAS Portal, all database providers accessed from a portal instance had to be on the same physical database server that contained the portal instance.

In Oracle Portal release 3.0.9, it was possible to distribute database portlets across database servers. To do this the user had to register each portal 'node' with each other which created a database link between the 'nodes'. These portal nodes would not function beyond a firewall. Furthermore the registration of the portal nodes was symmetric, which made the registration of multiple nodes hard to manage

Oracle Portal already had the concept of Web providers where the communication between the portal and the provider is done with the open protocols HTTP and SOAP. The PDK-Java services allow users to easily develop providers in Java that receive SOAP messages and respond accordingly.

The Federated Portal Adapter is a module written in the portal instance (in both Java & PL/SQL) that receives the SOAP messages for a Web provider, parses the SOAP and then dispatches the messages to a database provider as PL/SQL procedure calls. In effect, the Federated Portal Adapter makes a database provider behave exactly the same way as a Web provider. This allows users to distribute their database providers across database servers. All remote providers can now be treated as Web providers, hiding their implementation from the user and effectively replacing the distributed portal installations.

11.1.2 Differences Between Database Providers and Web Providers

The biggest difference between database providers and Web providers is that typically database providers use a portal session within the code, so that as part of the Federated Portal Adapter a portal session is created on the remote portal instance. The SOAP messages were extended to contain enough information to create a session on the remote portal instance, which means that the user in the remote portal must be the same user as in the local portal. For example, if 'UserA' is running in 'PortalA' and is using a provider on 'PortalB' through the Federated Portal Adapter then a session will be created in 'PortalB' for 'UserA'. Typically this means that 'PortalA' and 'PortalB' would share the same Oracle Application Server Single Sign-On, as partner applications. However an alternative arrangement could be that they have separate OracleAS Single Sign-Ons but the OracleAS Single Sign-Ons share the same name server. An example could be two OracleAS Single Sign-Ons sharing the same Oracle Internet Directory instance.

11.1.3 Use of the Federated Portal Adapter

The use of the Federated Portal Adapter can be divided into three categories:

Table 11–1 Use of the Federated Portal Adapter

Category	Description
OracleAS Portal Database Providers	Portal DB Providers created within OracleAS Portal will have the necessary code to be run through the Federated Portal Adapter. This means that applications created containing forms, charts, reports, and so on, can be shown on any other portal instance.
Pages	Pages exposed as portlets can also be run through the Federated Portal Adapter. Regions within pages can contain portlets or items. Using the Federated Portal Adapter these can now be accessed from any portal instance.
User Created Providers	Users may wish to create their own PL/SQL providers. You will be able to expose these providers through the Federated Portal Adapter as long as they are coded in accordance with the guidelines given in this chapter.

11.1.4 Security Issues

The Federated Portal Adapter creates a portal session in the remote portal based on the information passed in an initSession SOAP message. This introduces a security issue because it may be possible to replicate these SOAP messages and create sessions for any user on a portal and then access the portal as that user. To avoid this, an

encryption key is shared between the two portals and part of the SOAP message is encrypted using that key. The requested private portal session can only be created if the previously shared key can decrypt it. Otherwise a PUBLIC session is created. The request to display a portlet is made with a Show message that is protected by the encrypted cookie which is created by the initSession SOAP message. The use of an encryption key means that the Federated Portal Adapter can safely trust the incoming SOAP message and create portal sessions in the portal instance without opening the portal to hackers.

> **See Also:** Section 11.2.2, "Federated Portal Adapter User Authentication Using HMAC"

If it is known that the portal instance will only be accessed through the Federated Portal Adapter from other portal instances, then security can be enhanced by configuring the listener to restrict access from computers other than the known portal instances. This is done by using the 'Allow' directive in the `httpd.conf` file.

11.1.5 Federated Portal Adapter Related Portlet Modifications

It should be noted that database providers written before Oracle Application Server will not work when accessed through the Federated Portal Adapter if one of the following conditions is true:

- The portlet contains relative links.
- The portlet is personalizable.

All links within a portlet should be absolute links, that is, `http://<host>:<port>/images/foo.gif` rather than relative, `/images/foo.gif` when using the Federated Portal Adapter. This is because the request is processed by the Parallel Page Engine on the local portal instance. Relative links will therefore be interpreted as relative to the local portal and not to the portal containing the portlet.

Personalization is an issue because the processing of personalization is different between database and Web providers. For Web providers the personalization form is submitted to the Parallel Page Engine of the local portal, which in turn calls the portlet again and the personalizations are saved and the page is redirected appropriately. Because database providers accessed through the Federated Portal Adapter are effectively Web providers, this method of personalization should be undertaken for these providers. A public API is provided (WWPRO_API_ADAPTER) to do this.

Portal Database Portlet Providers developed in previous releases of OracleAS Portal will be upgraded automatically to work with the Federated Portal Adapter. Pages exposed as providers can also be accessed through the Federated Portal Adapter.

11.2 Setting Up the Environment to Use the Federated Portal Adapter

To use the Federated Portal Adapter there are a few administrative steps that must be undertaken. These steps are:

- Checking the PlsqlSessionCookieName Value
- Federated Portal Adapter User Authentication Using HMAC
- Setting the Cookie Domain
- Sharing an OracleAS Single Sign-On and an Oracle Internet Directory Server

11.2.1 Checking the PlsqlSessionCookieName Value

DADs must have a unique `PlsqlSessionCookieName` value for all the portals accessed by the Federated Portal Adapter.

For example,

- `portal1` can have the schema name `portal`, the DAD name `portal` and the `PlsqlSessionCookieName` value `portal1`.
- `portal2` can have the schema name `portal`, the DAD name `portal`, but must have a different `PlsqlSessionCookieName` value, like `portal2`.

> **Note:** In previous releases of OracleAS Portal, the DAD name had to be the same as the schema name, and the DAD name was always the same as the name of the session cookie created. This is no longer the case. You can now specify the name of the cookie created when portal is accessed by the DAD, and the schema name does not have to be the same as the DAD name.

Oracle Enterprise Manager 10*g* can be used to update the Session Cookie Name. To do this:

1. Navigate to the Oracle Enterprise Manager 10*g* Application Server Control Console.

   For details, see Section 7.2.1, "Accessing the Application Server Control Console".
2. Navigate to the Application Server instance where you would like to add the DAD.
3. Select **HTTP Server** from the System Components table.
4. Click **Administration**.
5. Click **PL/SQL Properties**.
6. To edit an existing DAD, click the DAD name in the **DADs** section.
7. Click **Document, Alias and Session** in the navigation area on the left.
8. Enter a new value for **Session Cookie Name** in the page, and click **OK**.
9. Restart the Oracle HTTP Server and OC4J_Portal.

You can also manually change the `PlsqlSessionCookieName` value in the `dads.conf` file. This file is located under:

ORACLE_HOME`/Apache/modplsql/conf/dads.conf`

A typical entry in this file looks like this:

```
<Location /pls/portal>
    SetHandler pls_handler
    Order allow,deny
    Allow from All
    AllowOverride None
    PlsqlDatabaseUsername portal
    PlsqlDatabasePassword SomePassword
    PlsqlDatabaseConnectString myhost.domain.com:1521:mySID
    PlsqlDefaultPage portal.home
    PlsqlAuthenticationMode SingleSignOn
    PlsqlSessionCookieName portal
    PlsqlMaxRequestsPerSession 500
```

```
    PlsqlDocumentTablename portal.wwdoc_document
    PlsqlDocumentPath docs
    PlsqlDocumentProcedure portal.wwdoc_process.process_download
    PlsqlPathAlias url
    PlsqlPathAliasProcedure portal.wwpth_api_alias.process_download
    PlsqlFetchBufferSize 128
</Location>
```

To edit a DAD entry, change the value of `PlsqlSessionCookieName` to, for example, `portal2`. After saving the file, update the Oracle HTTP Server configuration and restart the middle tier as follows:

```
MID_TIER_ORACLE_HOME/dcm/bin/dcmctl updateconfig -ct ohs
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_Portal
```

> **See Also:** Section 4.5.3, "Configuring a Portal DAD" for instructions on how to configure a DAD by using Application Server Control Console

> **Note:** It is recommended that you edit the `dads.conf` file using Application Server Control Console.
>
> If you manually edit the `dads.conf` file, then you must add the necessary `mod_rewrite` and `mod_oc4j` directives to the `httpd.conf` and `mod_oc4j.conf` files respectively. To do this, perform the steps mentioned in Section E.2, "DAD Configuration File (dads.conf)" using the Application Server Control Console.

11.2.2 Federated Portal Adapter User Authentication Using HMAC

Federated Portal Adapter functionality will support the registering of remote Database providers between geographically dispersed portals. Database providers are registered as if they were Web providers residing at a special URL on the remote portal.

> **Note:** If you are only rendering public content in the remote portlets, you can ignore this section.

In order that more than just public content can be rendered in the remote portlets we require that in some way we can guarantee that user A on one portal is the same as user A on another portal. This will typically be achieved by using a shared Oracle Application Server Single Sign-On using the *partner application* feature, but may also be achieved with a shared name server (for example, Oracle Internet Directory), synchronized name servers or a manual process.

If this environment can be achieved, then using the *Hash Message Authentication Code* (HMAC) authentication mechanism, private sessions can be initiated on a remote portal to render private content of remote portlets.

Setting the HMAC Keys

If the administrator of portal A wishes to permit users of portal B to create private sessions on portal A, a private 'key' will have to be stored on each portal. This key is used to encode and decode portions of each SOAP request sent between them. If a key is missing or they are different on each portal, only PUBLIC sessions will be created.

A key must be at least 10 characters long, and one administrator should inform the other administrator of its value in a suitably secure way.

SQL scripts are provided to perform the task of maintaining the key store - all are found in the *ORACLE_HOME*/portal/admin/plsql/wwc directory.

Table 11–2 SQL Scripts for Maintaining the Key Store

Script	Description
proadsss.sql	Sets the key at the sending end (portal instance on which the page with the remote portlets is created).
proadssr.sql	Sets the key at the receiving end (portal instance on which the portlets are created).
proadsds.sql	Removes the key at the sending end (portal instance on which the page with the remote portlets is created).
proadsdr.sql	Removes the key at the receiving end (portal instance on which the portlets are created).

In each case, *sending* and *receiving* refer to the SOAP message.

Example 11–1 Setting the HMAC Keys:

In the example mentioned earlier, portal B is the sender (sending SOAP and show requests) and portal A is the receiver of those requests. The portal administrator of portal B must connect to SQL*Plus as the portal owner and run:

```
SQL> @proadsss
Enter provider portal PL/SQL Adapter URL:
http://<portalA_hostname>:<port>/adapter/<portalA_DAD>
Enter shared key:<shared key>
exit;
```

The portal administrator of portal A must connect to SQL*Plus as the portal owner and run:

```
SQL> @proadssr
Enter provider portal PL/SQL Adapter URL:
http://<portalB_hostname>:<port>/adapter/<portalB_DAD>
Enter shared key:<shared key>
exit;
```

If sharing of providers is required both ways, then this will need to be repeated the other way round, possibly with different shared keys. It should also be noted that a portal can expose its providers to several other portal instances (for example, 'Portal A' exposes providers to 'Portal B' and 'Portal C') and separate keys can be set up between each of the portal instances.

11.2.3 Setting the Cookie Domain

Normally cookie domains are restricted to a single computer. This can be widened by running a script on each portal, and then selecting the **Web provider in same cookie domain as the portal** option on provider registration. Once this is done, '*deep link*' functionality can be achieved. This means that when you click a link in a portlet rendered by the Federated Portal Adapter, the browser renders the referred page (typically from the remote portal). The session context that has already been established is also maintained.

Cookies received by a browser, or other HTTP client, are sent to servers if the domain of the cookie matches the server's host name. So cookies with the domain '`.co.uk`' and '`mycompany.co.uk`' will be sent with a request to '`http://mycompany.co.uk/portal/pls/etc/etc`'. By default the scope of cookies created by portal is restricted to the host name of the middle-tier computer.

Because communication to the portlets is done in the middle tier by the Parallel Page Engine (PPE) and not the browser, the session cookie for the remote portal will, by default, not be sent to the remote portal when links are followed within the portlet.

This can be solved by widening the scope of the cookies created by portal and making sure that the cookies received by the PPE are sent back to the browser Widening the scope of the cookies created by portal is achieved by running the SQL script `ctxckupd.sql` in the *ORACLE_HOME*`/portal/admin/plsql/wwc` directory.

For example, there are two portals:

- `http://myhost1.mycompany.co.uk:3000/portal/pls/portalA`
- `http://myhost2.mycompany.co.uk:4000/portal/pls/portalB`

and a provider is registered from 'Portal B' on 'Portal A'.

When showing a page on 'Portal A' that contained a portlet from 'Portal B' by default a portal session cookie for 'Portal B' whose domain is '`myhost2.mycompany.co.uk:4000`' would be created, and sent to the PPE. If the '*Web provider in same cookie domain as the portal*' property is checked on the provider registration page then this cookie will be sent back to the browser, but the domain of the cookie will then be '`myhost1.mycompany.co.uk:3000`' because that is where it is being sent from, because the PPE is at '`myhost1.mycompany.co.uk:3000`').

If a link is followed from within the portlet the cookie is not sent with the request, because the domain of the cookie does not match with that of the host of the request.

To solve this, connect to SQL*Plus as the portal owner of each portal and run *ORACLE_HOME*`/portal/admin/plsql/wwc/ctxckupd.sql` and broaden the scope of the domain's cookies created by OracleAS Portal so each portal is in the same domain. Once this is done, the scope of the cookie domains created by any of the portals will be broad enough to be sent back to the browser. Links within the portlet will then work correctly.

See Also: Section C.5, "Configuring the Portal Session Cookie"

11.2.4 Sharing an OracleAS Single Sign-On and an Oracle Internet Directory Server

The benefits of single sign-on can be maximized, by utilizing a common Identity Management server. portal session information is passed to the remote portal, which uses the Federated Portal Adapter to create a session. It is recommended that all portals on which you want to create private sessions, share the same Oracle Internet Directory server and the same OracleAS Single Sign-On.

For example, if a user 'JSMITH' displays a page on one portal and a portlet on that page is being sourced from the Federated Portal Adapter on a remote portal, then a session is created on the remote portal for user 'JSMITH'. If the two portals do not share OracleAS Single Sign-On then 'JSMITH' may be the user name for 'John Smith' on one portal and 'Jane Smith' on the other. To avoid this sort of problem, ensure that all the portals participating in the Federated Portal are configured to use a single Oracle Identity Management. They should all use the same OracleAS Single Sign-On for authentication. However, if the portlets being shown are 'public' then there is no need to share the OracleAS Single Sign-On and a public portal session will be created at the remote portal instance.

If you currently have two portals using distinct OracleAS Single Sign-On servers, you may first need to consolidate the OracleAS Single Sign-On servers. To do this, refer to the information on consolidating multiple servers in the *Oracle Application Server Single Sign-On Administrator's Guide*.

Consolidating the servers means that you will be decommissioning one of the servers and identifying the other as the common server for both portals to use. Then you'll need to configure the portal that was configured to use the decommissioned OracleAS Single Sign-On to the consolidated one. To do this, you have to update the Portal Dependency Settings File (`iasconfig.xml`) and run the Portal Dependency Settings Tool (`ptlconfig`), as shown in Example 11–2.

Example 11–2 Sharing an OracleAS Single Sign-On and an Oracle Internet Directory Server

You have two portals, `portal1` and `portal2`. You decide to decommission the SSO server for `portal2` and configure `portal2` to use the SSO server for `portal1`. To do this, perform the following steps:

1. Update the `iasconfig.xml` file and change the OIDDependency element for `portal2` to point to the same OIDComponent that `portal1` refers to, as shown in the following sample `iasconfig.xml` file:

```
<IASConfig XSDVersion="1.0">
   <IASInstance Name="iAS-1" Host="abc.company.com">
      <WebCacheComponent ListenPort="3002" InvalidationPort="3003"
InvalidationUsername="invalidator" InvalidationPassword="welcome1"
SSLEnabled="false"/>
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
   </IASInstance>

   <IASInstance Name="iAS-2" Host="xyz.company.com">
      <OIDComponent AdminPassword="welcome1" AdminDN="cn=orcladmin"
SSLEnabled="false" LDAPPort="3002"/>
   </IASInstance>

   <PortalInstance DADLocation="/pls/portal1" SchemaUsername="portal1"
SchemaPassword="welcome1" ConnectString="db1">
      <WebCacheDependency ContainerType="IASInstance" Name="iAS-1"/>
      <OIDDependency ContainerType="IASInstance" Name="iAS-2"/>
      <EMDependency ContainerType="IASInstance" Name="iAS-1"/>
   </PortalInstance>

   <PortalInstance DADLocation="/pls/portal2" SchemaUsername="portal2"
SchemaPassword="welcome1" ConnectString="db2">
      <WebCacheDependency ContainerType="IASInstance" Name="iAS-1"/>
      <OIDDependency ContainerType="IASInstance" Name="iAS-2"/>
      <EMDependency ContainerType="IASInstance" Name="iAS-1"/>
   </PortalInstance>
</IASConfig>
```

   You can remove the OIDComponent element that `portal2` was earlier using, if it is not being referenced by any other PortalInstance element in the file.

2. Run the `ptlconfig` tool, as shown in the following example:

```
ptlconfig -dad portal2 -oid -sso
```

> **Note:** Refer to Appendix A, "Using the Portal Dependency Settings Tool and File" for more information.

11.3 Registering a Provider Using the Federated Portal Adapter

Registering a provider through the Federated Portal Adapter is like registering any Web provider. You must perform the following steps:

1. On the first page of the **Register Provider** screen enter the **Name**, **Display Name**, **Timeout**, and **Timeout Message** as you would normally. Make sure the **Implementation Style** is set to **Web**. Although the provider is actually written in PL/SQL, all communication to it is as a Web provider and not a database provider so it is important to set the **Implementation Style** to **Web**.

2. On the second page enter the URL of the adapter service. The syntax for the URL should be:

   `http://host:port/adapter/dad/schema`

   If the DAD and the schema are the same you can just use:

   `http://host:port/adapter/dad`

   where the host, port, DAD and schema locate the remote portal instance. You can verify that this is the correct URL by pasting it into a browser.

   If the URL is correct you should get to a page with the message "Congratulations - you got to the adapter test page"

3. Select the **Web provider in same cookie domain as the portal** option. This will ensure that cookies generated from the provider will be sent back to the browser. Note that it may be necessary to broaden the scope of the cookies created by portal as described earlier.

4. Enter the 'Service Id'. This should be in the form '`urn:<provider name>`'. Where `provider name` is the name of the provider on the remote portal instance, this is case sensitive and will be upper case. This is the information that the Federated Portal Adapter uses to locate the specific provider at the remote portal.

   Note that for page groups exposed as providers, the name of the provider will be something like 'MYPAGE970D272EBE9D2D0FE034080020F7DA4B' it is important that you specify this 'Name' rather than the 'Display Name'. The name and display name can be accessed from the **Remote Providers** portlet, available in the **Portlets** subtab under the **Administer** tab in OracleAS Portal. Clicking the **Browse Providers** icon displays the names of all the providers.

5. In the **User/Session Information** section, select the **User** radio button and set the **Login Frequency** to be **Once Per User Session**. These settings make sure that information is sent with the request to allow a portal session to be created on the remote portal instance.

Note: When you create or register a new provider, a page is created in the Portlet Repository under **Portlet Staging Area** to display portlets for that provider. This page is not visible to all logged in users. It is only visible to the user who published the provider, and the portal administrator. The publisher or portal administrator can change the provider page properties to grant privileges to appropriate users and groups, as required.

11.4 Writing Custom Portlets Using the Federated Portal Adapter

There are two main areas of code that need special attention when writing database providers that are accessed through the Federated Portal Adapter. They are:

- Relative Links
- Personalization

11.4.1 Relative Links

Any links within portlets that are accessed through the Federated Portal Adapter should absolute rather than relative. If links are relative then they will not work because they will be relative to the local middle tier rather than the remote middle tier. For example, links should be of the form '`http://myhost.mycompany/etc/etc`' rather than '`/etc/etc`'.

11.4.2 Personalization

The way personalizations work when accessing portlets through the Federated Portal Adapter is now very similar to the method used by PDK-Java portlets. There are two main areas of the portlet code that need to be changed to make personalization work through the Federated Portal Adapter:

- The show call of the portlet needs additional logic to show the portlet in *edit_defaults* mode, or, if the parameter '*p_mode*' is null, in *personalize* mode. If the '*p_mode*' is 'OK', 'APPLY' or 'RESET', then the personalizations should be saved as appropriate.
- The <FORM> HTML tags generated for the personalize page should be created using the procedure *wwpro_api_adapter.open_form*. This will ensure that the action for the form is correct, and that the correct parameters are passed upon page submission. The sequence of events when submitting the personalization form is:

  1. The page submits to the 'local' PPE. There are several standard parameters that need to be sent with this submission (for example, *_providerid*, *_dad*, *p_action*, and so on) and the parameters that are being personalized. The procedure *wwpro_api_adapter.open_form* is supplied to make the generation of this submission as simple as possible.
  2. The PPE then shows the personalization page again. However the '*p_action*' parameter will now be set so that during the *show_portlet* call of the portlet it will be one of the following settings:

     '**OK**' - In this case the personalizations should be saved and then there should be a redirect to the page containing the portlets.

     '**APPLY**' - In this case the personalizations should be saved and the personalization page is shown.

     '**RESET**' - In this case the default values for parameters are queried and the personalization page is shown.

The *database services provider* is a sample provider in the Oracle Application Server Portal Developer Kit (PDK) that works with the Federated Portal Adapter. For more information, see the Portal Developer Kit on the Oracle Technology Network (OTN), `http://www.oracle.com/technology/products/ias/portal/pdk.html`.

11.5 Troubleshooting Federated Portal Adapter

There are some known restrictions showing page portlets with the Federated Portal Adapter.

- The **Show Details** mode does not work, that is, the portlet name cannot be displayed as a link that shows additional information about the portlet.
- If the page portlet contains tabs, then clicking a tab is a 'deep link' and the rendered page takes over the whole page, that is, it is not shown within the original page as a portlet.
- The rendering of navigation pages, which includes the page banner, does not work properly when pages are displayed through the Federated Portal Adapter. For example, the **Personalize** link in a *regular page portlet* displays personalization options for the container page, but this is not the case in a *remote page portlet*. Also, page portlets shown through the Federated Portal Adapter do not display the banner of the container page, whereas the banner is displayed in the case of *regular page portlets*.
- If the page portlet has a navigation page portlet that has a sub page region in it, the sub page region will not be displayed on the page portlet when it gets rendered through the Federated Portal Adapter. For a non-remote page portlet, the region shows the sub pages of the container page holding the portlet.
- When you export and import Federated Portal Adapter portlets, the portlets are not shown on the target instance if you have not performed the following tasks on the target portal instance:

  1. Configure the target portal instance to use the PL/SQL adapter running on the source portal instance.
  2. Grant the **View** permission to the target portal instance for the Federated Portal Adapter Web Provider page in the source portal instance.

Part IV

Appendixes

Part four contains the following appendixes:

A

Using the Portal Dependency Settings Tool and File

OracleAS Portal is dependent on other Oracle Application Server components, such as OracleAS Web Cache and Oracle Internet Directory and it can be configured to work with load balancing routers and reverse proxy servers. The *Portal Dependency Settings File* (`iasconfig.xml`) file in OracleAS Portal stores configuration data about dependent components in a central place. You can use `iasconfig.xml` to check and edit settings used by an OracleAS Portal instance. If you make changes in the `iasconfig.xml` file, you must use the *Portal Dependency Settings Tool* (`ptlconfig`) to update in the OracleAS Portal schema in the Oracle Application Server Metadata Repository.

> **Note:** The `ptlasst` command line utility has been removed in this release. All of the functionality that was provided by `ptlasst` in the previous release is now available through the Portal Dependency Settings tool and file `ptlconfig` and `iasconfig.xml`.

This appendix discusses the Portal Dependency Settings file and the Portal Dependency Settings tool, in the following sections:

A.1 Portal Dependency Settings Tool

There are two ways to update the OracleAS Metadata Repository with any configuration changes:

- Using the Oracle Enterprise Manager 10*g* Application Server Control Console. See Chapter 7, "Monitoring and Administering OracleAS Portal" for more information. If you make configuration changes using the Application Server Control Console, the `iasconfig.xml` and the OracleAS Portal schema in the OracleAS Metadata Repository will be updated for you automatically.
- Editing the `iasconfig.xml` file and running the script `ptlconfig`.

The `ptlconfig` script can:

- Update the portal schema in the OracleAS Metadata Repository for a *specific* portal instance defined in the Portal Dependency Settings file.
- Encrypt all plain text passwords in the Portal Dependency Settings file.

- Update OracleAS Web Cache, Oracle Internet Directory, Oracle Enterprise Manager 10*g*, and OracleAS Portal site data, as defined in the Portal Dependency Settings file.
- Update the Portal Dependency Settings file based on configuration information stored in the portal schema. This is useful for creating entries for migrated portals, and also to restore settings if the Portal Dependency Settings file becomes corrupt.
- Create or delete provisioning profiles in Oracle Internet Directory of an OracleAS Portal instance. Refer to Section 6.1.6.3, "Relationship Between OracleAS Portal and Oracle Directory Integration Platform" for more information about provisioning profiles.

  OracleAS Portal uses a provisioning profile to receive notifications when user or group privilege information in Oracle Internet Directory changes. This enables OracleAS Portal to keep its authorization information synchronized with the information stored in Oracle Internet Directory. By default, this provisioning profile is enabled.

The configuration script file is named `ptlconfig` (on UNIX) and `ptlconfig.bat` (on Windows). It is located in *ORACLE_HOME*`/portal/conf`, where *ORACLE_HOME* is the OracleAS Portal and OracleAS Wireless middle-tier home.

You can use this script as follows:

```
ptlconfig -dad <dad> -pw <portal schema password or Oracle Internet Directory password> [-em]
[-oid]
[-site] [-wc] [-dipreg] [-dipunreg] [-sso [-host <host name> -port <port number> [-ssl]] ]|
-encrypt |
-load -schema <schema username> -pw <schema password> -conn <connect string> [-lp ldap_ssl_port]|
```

When you run `ptlconfig`, the log file `ptlconfig.log` is created in the directory *ORACLE_HOME*`/portal/logs`. If an error displays while running `ptlconfig`, refer to the full message text in the log file to resolve the error.

`ptlconfig` can be run in the following three modes:

- Configuration Mode
- Encryption Mode
- Load Mode

A.1.1 Configuration Mode

Updates a specific OracleAS Portal instance from the Portal Dependency Settings file.

Table A–1 Configuration Mode

Parameter	Description	Example
-dad	Updates a specific OracleAS Portal instance from the Portal Dependency Settings file. This is the Portal DAD name.	`ptlconfig -dad portal`

Table A–1 (Cont.) Configuration Mode

Parameter	Description	Example
-pw	OracleAS Portal schema password or Oracle Internet Directory administrator password. **Note:** You can provide either the portal schema password or the Oracle Internet Directory administrator password, to authenticate.	`ptlconfig -dad portal -pw welcome1`
-em	Updates Oracle Enterprise Manager 10*g* data as defined in the Portal Dependency Settings file.	`ptlconfig -dad portal -em`
-oid	Updates Oracle Internet Directory data as defined in the Portal Dependency Settings file.	`ptlconfig -dad portal -oid`
-site	Configures OracleAS Portal to work with the Oracle HTTP Server when configuration changes are required in OracleAS Portal due to changes in the Oracle HTTP Server component. For example, changes in the HTTP server host, port, or protocol. It also configures OracleAS Portal as a partner application for OracleAS Single Sign-On as defined in the Portal Dependency Settings file. **Note:** The file `iasconfig.xml` does not model OracleAS Single Sign-On components. The `ptlconfig` tool gets the SSO details (for example, the schema name and password) from the Oracle Internet Directory instance that the portal instance is configured to use.	`ptlconfig -dad portal -site`
-wc	Updates OracleAS Web Cache data for a specific portal instance, as defined in the Portal Dependency Settings file.	`ptlconfig -dad portal -wc`
-dipreg	Used to create the provisioning profiles in Oracle Internet Directory. **Note:** In this release, running DIPREG to register provisioning profiles actually updates any existing profile. In previous releases, you had to first run DIPUNREG and then DIPREG again, which could result in a minor loss of changes. The new behavior ensures that there are no lost changes.	`ptlconfig -dad portal -dipreg`
-dipunreg	Used to delete the provisioning profiles in Oracle Internet Directory of the OracleAS Portal instance.	`ptlconfig -dad portal -dipunreg`

Table A–1 (Cont.) Configuration Mode

Parameter	Description	Example
-sso	Creates partner application entries in OracleAS Single Sign-On. When run without any additional parameters, partner application details are updated using the details from `iasconfig.xml`. See Section 5.4.3, "Register OracleAS Portal with OracleAS Single Sign-On" for more details on when you can use this parameter.	`ptlconfig -dad portal -sso`
-host	Name of the host that you want to register as a partner application with OracleAS Single Sign-On. This parameter is used with the `-sso` parameter.	`ptlconfig -dad portal -sso -host abc.company.com -port 7778`
-port	Port that is used for registration. This parameter is used with the `-sso` parameter.	`ptlconfig -dad portal -sso -host abc.company.com -port 7778`
-ssl	Indicates that the port is HTTPS. Used with the `-sso`, `-host` and `-port` properties. If not specified with `-host` and `-port`, it is assumed that it is an HTTP port.	`ptlconfig -dad portal -sso -host abc.company.com -port 7778 -ssl`

Note: Running `ptlconfig` in the `-sso` and `-site` modes updates the OracleAS Single Sign-On Query Path URL with the URL prefix of OracleAS Single Sign-On. If this URL is using the HTTPS protocol, the URL must be updated to use the HTTP protocol instead. Refer to the section "Setting the OracleAS Single Sign-On Query Path URL (HTTP)" in Section 6.3.2.1.2, "SSL to OracleAS Single Sign-On" for information on updating the OracleAS Single Sign-On Query Path URL.

A.1.2 Encryption Mode

Encrypts any plain text passwords in the Portal Dependency Settings file. For example:

```
ptlconfig -encrypt
```

A.1.3 Load Mode

Creates and updates entries in `iasconfig.xml` with the configuration settings of a specific portal schema.

Table A–2 Load Mode

Parameter	Description	Example
-schema	Name of the portal schema.	`ptlconfig -load -schema portal30 -pw welcome1 -conn abc.company.com:1521:s901dev3`

Table A–2 (Cont.) Load Mode

Parameter	Description	Example
-pw	Portal schema password.	`ptlconfig -load -schema portal30 -pw welcome1 -conn abc.company.com:1521:s901dev3`
-conn	Connect string to the portal repository. Refer to Table 7–3, " DAD Settings", for a description of the valid connect string formats.	`ptlconfig -load -schema portal30 -pw welcome1 -conn abc.company.com:1521:s901dev3`
-lp	Used to connect to Oracle Internet Directory for getting OracleAS Single Sign-On information. This is the LDAP SSL port of Oracle Internet Directory. See the section "Updating iasconfig.xml" in Section B.1, "Configuring and Managing the OracleAS Portal Instance" for more details on when you should use this parameter.	`ptlconfig -load -schema portal30 -pw welcome1 -conn abc.company.com:1521:s901dev3 -lp 4889`

Note: Names are used only as a way to reference elements, in `iasconfig.xml`. For example, the name of a middle tier, **`midtier`**`.abc.company.com`, might be changed to **`iAS1`**`.abc.company.com`.

A.2 Portal Dependency Settings File

The following sections describe the Portal Dependency Settings file in more detail:

- Name and Location
- Configuration Elements
- Sample Portal Dependency Settings File
- Updating the Portal Dependency Settings File
- Post-Installation Mapping in the Portal Dependency Setting File
- Common Configuration Mapping in the Portal Dependency Settings File

A.2.1 Name and Location

The name of the Portal Dependency Settings file is `iasconfig.xml`, and is located by default in *`ORACLE_HOME`*`/portal/conf`, where *`ORACLE_HOME`* is the OracleAS Portal and Oracle Application Server Wireless middle-tier home.

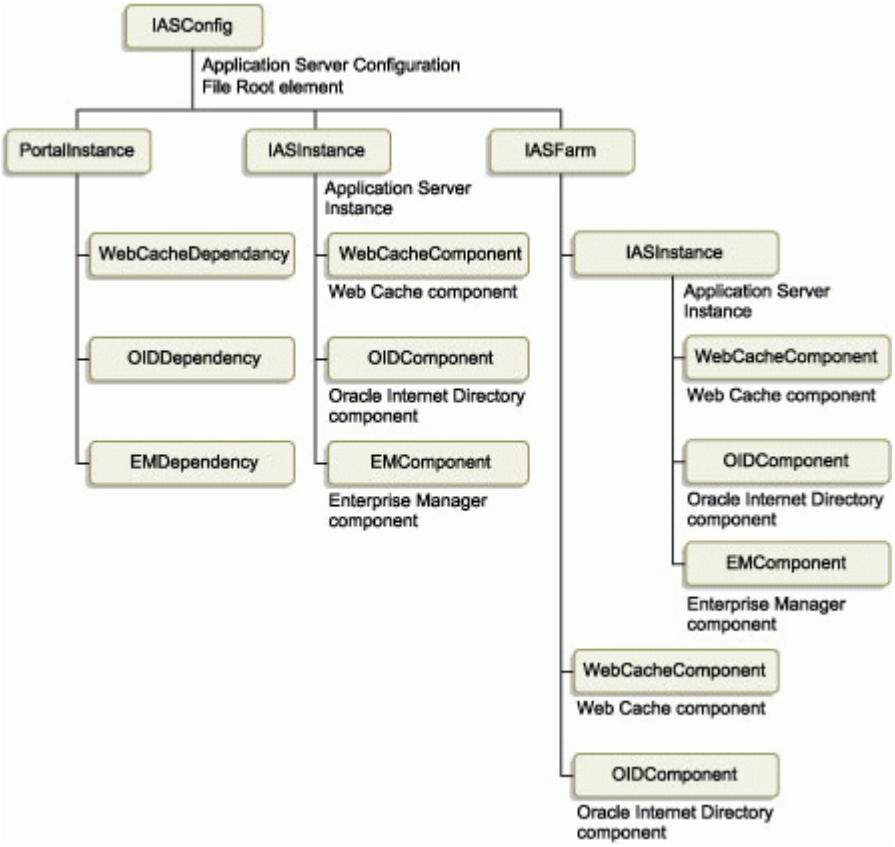
When using any of the tools that access the Portal Dependency Settings file, you can override the default location of the file by setting the environment variable IASCONFIG_LOC to the directory in which your file is stored, for example:

```
set IASCONFIG_LOC=/usr/local/as101202
```

A.2.2 Configuration Elements

The Portal Dependency Settings file is an XML file, that is made up of a number of elements that describe the settings of specific Oracle Application Server components and the dependencies portal instances have on them. Figure A–1 shows all the elements that can be modeled in the Portal Dependency Settings file. The Portal Dependency Settings file definition is modeled in the schema file `iasconfig.xsd`, which is located in *`ORACLE_HOME`*`/portal/conf`.

Figure A–1 Elements in the Portal Dependency Settings file



The individual elements are:

- IASFarm
- IASInstance
- PortalInstance
- WebCacheComponent
- OIDComponent
- EMComponent
- WebCacheDependency
- OIDDependency
- EMDependency

IASFarm

The *IASFarm* element represents a logical farm of Oracle Application Server instances, commonly used when modeling a multiple middle-tier scenario front-ended by a load balancing router (LBR). See Section 5.3, "Configuring Multiple Middle Tiers with a Load Balancing Router" for more information.

Table A–3 Element IASFarm

Attribute Name	Type	Description
Name	String	Unique farm name
Host	String	Published host name that users will enter in their Web browser to access OracleAS Portal

IASInstance

The *IASInstance* element represents a specific Oracle Application Server instance, which usually maps to an Oracle home.

Table A–4 Element IASInstance

Attribute Name	Type	Description
Name	String	Oracle Application Server instance name (for example, `as101202.host.domain`)
Host	String	Host computer

PortalInstance

These are the OracleAS Portal instance settings.

Table A–5 Element PortalInstance

Attribute Name	Type	Description
DADLocation	String	The name and location of the OracleAS Portal DAD (for example, `/pls/portal`).
ConnectString	String	OracleAS Metadata Repository connect string
SchemaUsername	String	OracleAS Portal schema user name
SchemaPassword	String	OracleAS Portal schema password

WalletPath	String	Path to the directory containing the Oracle Wallet containing the trust points for SSL access to the OracleAS Single Sign-On URL specified in `SSOQueryPath`.
		Use `file:` followed by the full path to the directory that contains the wallet, as shown in the following examples:
		On Windows:
		`WalletPath="file:C:\Oracle\Wallets"`
		On UNIX:
		`WalletPath="file:/export/home/oracle/wallets"`

WalletPassword	String	Password for the Oracle Wallet specified in `WalletPath`

Table A–5 (Cont.) Element PortalInstance

Attribute Name	Type	Description
SSOQueryPath	String	URL for HTTP/HTTPS access to OracleAS Single Sign-On from the portal repository. If `SSOQueryPath` is an HTTPS URL, then `WalletPath` and `WalletPassword` are required. If `SSOQueryPath` is not specified, it will be derived from the OracleAS Single Sign-On login URL.

Note: The `WalletPath`, `WalletPassword`, and `SSOQueryPath` attributes are optional. However, if `SSOQueryPath` (whether explicitly specified in `iasconfig.xml` or derived from the OracleAS Single Sign-On login URL) is an HTTPS URL, then the wallet information is required.

WebCacheComponent

These are the OracleAS Web Cache settings.

Table A–6 Element WebCacheComponent

Attribute Name	Type	Description
ListenPort	Integer	Listening port
InvalidationPort	Integer	Invalidation port
InvalidationUsername	String	Invalidation user name
InvalidationPassword	String	Invalidation password
SSLEnabled	String	Flag to indicate whether the listening port is SSL enabled. The value can be either `TRUE` or `FALSE`.

OIDComponent

These are the Oracle Internet Directory settings.

Table A–7 Element OIDComponent

Attribute Name	Type	Description
AdminPassword	String	Oracle Internet Directory administrator's password.
PortSSLEnabled	String	Flag to indicate whether the LDAP port is SSL enabled. The value can either be `TRUE` or `FALSE`.
LDAPPort	Integer	LDAP port that Oracle Internet Directory is running on.
AdminDN	String	Oracle Internet Directory administrator's distinguished name.

Note: The `orclSizeLimit` attribute, in the root node of the Oracle Internet Directory server, is a system operational attribute used to specify the maximum number of entries that can be returned by a search. The value of this attribute comes into effect when configuring OracleAS Portal with the Oracle Internet Directory server. All the groups, which belong to an identity management realm, are searched in the Oracle Internet Directory server to synchronize their profiles in the OracleAS Portal schema.

The value of the `orclSizeLimit` attribute must be configured such that it is large enough to accommodate the maximum number of groups for any single identity management realm in the Oracle Internet Directory server. This change is required only while configuring the Oracle Internet Directory server, following which the value of this attribute can be restored to an appropriate value.

You can view and set some of the operational attributes for each Oracle Directory server to which you are connected by using the Oracle Directory Manager. Refer to the section on Setting System Operational Attributes by Using Oracle Directory Manager in the *Oracle Internet Directory Administrator's Guide* for more details.

EMComponent

These are the Oracle Enterprise Manager 10*g* Application Server Control Console settings.

Table A–8 Element EMComponent

Attribute Name	Type	Description
ConsoleHTTPPort	Integer	Listening port
SSLEnabled	String	Flag to indicate whether the listening port is SSL enabled. The value can either be `TRUE` or `FALSE`.

WebCacheDependency

This is the OracleAS Portal instance reference to the OracleAS Web Cache it is using.

Table A–9 Element WebCacheDependency

Attribute Name	Type	Description
ContainerType	String	The type of the container the OracleAS Web Cache component is running under. This can be either *IASInstance* or *IASFarm*.
Name	String	*IASInstance* name or the unique *IASFarm* name, depending on *ContainerType*.
InvalidationHost	String	This attribute should be set if the OracleAS Web Cache host is different from the published host that a user enters to access OracleAS Portal. **Note:** This is an optional attribute. See "Specify the OracleAS Portal Published Address and Protocol" later in this appendix, for the scenario in which this attribute is used.

OIDDependency

This is the OracleAS Portal instance reference to the Oracle Internet Directory it is using.

Table A–10 Element OIDDependency

Attribute Name	Type	Description
ContainerType	String	The type of the container the Oracle Internet Directory component is running under. This can be either *IASInstance* or *IASFarm*.
Name	String	*IASInstance* name or the unique *IASFarm* name, depending on *ContainerType*.
LDAPSSLPort	String	LDAP SSL port value - Used to configure OracleAS Single Sign-On details for upgraded OracleAS Portal instances. See the section "Updating iasconfig.xml" in Section B.1, "Configuring and Managing the OracleAS Portal Instance" for more information.

EMDependency

This is the Oracle Enterprise Manager 10*g* Application Server Control Console managing this OracleAS Portal instance.

Table A–11 Element EMDependency

Attribute Name	Type	Description
ContainerType	String	The type of the container the Oracle Enterprise Manager 10*g* Application Server Control Console is being managed by. This should be set to *IASInstance*.
Name	String	*IASInstance* name

A.2.3 Sample Portal Dependency Settings File

The XML in Example A–1 represents the contents of a sample Portal Dependency Settings file.

Example A–1 Sample Portal Dependency Settings file

```
<IASConfig XSDVersion="1.0">

   <IASInstance Name="iAS-1.abc.company.com" Host="abc.company.com">
      <WebCacheComponent ListenPort="3002" InvalidationPort="3003"
InvalidationUsername="invalidator" InvalidationPassword="welcome1"
SSLEnabled="false"/>
   </IASInstance>

   <IASInstance Name="iAS-2.abc.company.com" Host="xyz.company.com">
      <OIDComponent AdminPassword="welcome1" PortSSLEnabled="false"
LDAPPort="3002" AdminDN="cn=orcladmin"/>
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
   </IASInstance>

   <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="welcome1" ConnectString="xyz.company.com:1521:s901dev3">
```

```
      <WebCacheDependency ContainerType="IASInstance"
Name="iAS-1.abc.company.com"/>
      <OIDDependency ContainerType="IASInstance" Name="iAS-2.abc.company.com"/>
      <EMDependency ContainerType="IASInstance" Name="iAS-1.abc.company.com"/>
   </PortalInstance>

</IASConfig>
```

In this example, the OracleAS Portal instance is:

- Accessed from the Database Access Descriptor (DAD) `/pls/portal`.
- Dependent on:
  - OracleAS Web Cache component running in Oracle Application Server instance **iAS-1**
  - Oracle Internet Directory component running in Oracle Application Server instance **iAS-2**
  - Oracle Enterprise Manager 10*g* Application Server Control Console component running in Oracle Application Server instance **iAS-1**

A.2.4 Updating the Portal Dependency Settings File

If the Portal Dependency Settings file is accessible over a network file system, you can share the file across multiple hosts, avoiding the need to manually replicate it every time the file is modified. If the installation is running on an operating system which supports symbolic links, it is recommended that you use this mechanism to reference a shared file, instead of setting the IASCONFIG_LOC environment variable.

If, however, the Portal Dependency Settings file is not accessible over the network, you must ensure that the file is kept up-to-date with changes to your site topology. The Portal Dependency Settings file is used to configure the OracleAS Portal schema with details of OracleAS Web Cache, Oracle Internet Directory, and Oracle Enterprise Manager 10*g* that it is using. It is not required that it is copied into each individual middle tier in your site, but you must ensure that any changes to the components modeled in the file that affect OracleAS Portal configuration are updated in the file.

Let's use the configuration defined in Section 5.3, "Configuring Multiple Middle Tiers with a Load Balancing Router", to demonstrate how the Portal Dependency Settings file is kept up-to-date.

1. The Portal Dependency Settings file gets first created in Section 5.3.1, "Step 1: Install a Single Portal and Wireless Middle Tier (M1)", during the installation. Example 5–1, "iasconfig.xml After the First Middle-Tier Installation" shows how it looks.

   This file will be located on computer `m1.abc.com`, typically in *ORACLE_HOME*`/portal/conf` of the middle tier that has just been installed.

2. In Section 5.3.1, "Step 1: Install a Single Portal and Wireless Middle Tier (M1)", the Portal Dependency Settings file is manually changed. Example 5–2, "iasconfig.xml File Edited to Include Farm Element" shows this.

   This file will be on computer `m1.abc.com`, typically in *ORACLE_HOME*`/portal/conf` of the middle tier installed in Step 1. You use the `ptlconfig` tool as shown in Section A.1, "Portal Dependency Settings Tool", after you make changes to the file. For example:

   ```
   ptlconfig -dad <portal_dadname> -wc -site
   ```

Any future changes to the OracleAS Web Cache, Oracle Internet Directory, or Oracle Enterprise Manager 10*g* settings in `iasconfig.xml` should be made using the Application Server Control Console, or manually on `m1.abc.com`. If you edit `iasconfig.xml` manually, then you must also use the `ptlconfig` tool again after you make changes.

Note: Changes to OracleAS Portal's OracleAS Web Cache settings can also be made on the **Portal Web Cache Settings** page. See Section 7.3.3, "Portal Web Cache Settings Link" for more information.

Typically, the host name and port number, by which OracleAS Portal is addressed, uses the OracleAS Web Cache host name and port number. This is because, in a simple configuration, browser requests go directly to OracleAS Web Cache. However, in a configuration that has a load balancing router (LBR), or reverse proxy server front-ending OracleAS Web Cache, the host name and port number defined on this page may need to reflect that of the LBR, or reverse proxy server.

In this configuration, you want OracleAS Web Cache invalidation messages to be sent directly to the OracleAS Web Cache host, as opposed to the LBR, or reverse proxy server. In the scenario where your published host name is different from the host name used for OracleAS Web Cache invalidation, you can use the Portal Dependency Settings file `iasconfig.xml` to establish these settings. See the section "Specify the OracleAS Portal Published Address and Protocol" for details.

---

3. In Section 5.3.5, "Step 5: Configure the New Middle Tier (M2) to Run Your Existing Portal", the Portal Dependency Settings file on `m2.abc.com` needs to be updated manually with the settings defined in the `iasconfig.xml` file on `m2.abc.com`.

Specify the OracleAS Portal Published Address and Protocol

Typically, the host name and port number, by which OracleAS Portal is addressed, uses the OracleAS Web Cache host name and port number. This is because, in a simple configuration, browser requests go directly to OracleAS Web Cache. However, in a configuration that has a reverse proxy server front-ending OracleAS Web Cache, the host name and port number defined should reflect that of the reverse proxy server.

In this configuration, you want OracleAS Web Cache invalidation messages to be sent directly to the OracleAS Web Cache host, as opposed to the reverse proxy server. In the scenario where your published host name is different from the host name used for OracleAS Web Cache invalidation, you can use the Portal Dependency Settings file, to establish these settings.

To configure this appropriately, perform the following steps:

1. Edit `iasconfig.xml` (located by default in *`ORACLE_HOME`*`/portal/conf`) and specify a new property, *InvalidationHost*, within the WebCacheDependency element. The InvalidationHost property should point to the OracleAS Web Cache host. A sample WebCacheDependency entry in `iasconfig.xml` would look like this:

```
<WebCacheDependency ContainerType="IASInstance" Name="Farm-1.abc.com"
InvalidationHost="internal.company.com"/>
```

2. To prevent access to Oracle Enterprise Manager 10*g* from the outside, the Oracle Enterprise Manager 10*g* link provided by OracleAS Portal needs to be changed back to point to the internal server. To do this, edit `iasconfig.xml` and:

   a. Add a new `IASInstance` element pointing to the OracleAS Web Cache host. This element should contain the `EMComponent` entry.

   b. Set the `EMDependency` entry to point to the internal server.

   An example is shown here:

```
<IASInstance Name="iAS.internal.company.com" Host="internal.company.com">
   <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
</IASInstance>

<PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="@Beyh8p2bOWELQCsA5zRtuYc="
ConnectString="cn=iasdb,cn=oraclecontext">
   <EMDependency ContainerType="IASInstance" Name="iAS.internal.company.com"/>
</PortalInstance>
```

3. Run `ptlconfig` (typically located in the directory *MID_TIER_ORACLE_HOME*`/portal/conf`) as shown in the following example:

```
ptlconfig -dad portal -site -wc -em
```

4. Optionally, re-register the Wireless gateway URL with the load-balancer's address. See Section 5.11, "Configuring OracleAS Wireless" for more information.

See Also:

- Section 7.3.3, "Portal Web Cache Settings Link" for more information about the Portal Web Cache Settings page.
- Appendix A, "Using the Portal Dependency Settings Tool and File" for more information about the Portal Dependency Settings File and Tool.

A.2.5 Post-Installation Mapping in the Portal Dependency Setting File

When OracleAS Portal is installed, appropriate entries are created in the Portal Dependency Settings file, based on what is installed.

In an Application Server installation, the dependencies of OracleAS Portal on Oracle Application Server Web Cache and Oracle Internet Directory are added to the Portal Dependency Settings file. Existing information is not updated if duplicate entries are encountered during the installation. Instead, a warning is output to the installation log file that the entries already exist.

See Also: Chapter 3, "Installing OracleAS Portal" for more information about the different installation types.

Note: By default, the Portal Dependency Settings file is accessed from *ORACLE_HOME*`/portal/conf`, where *ORACLE_HOME* is the OracleAS Portal and Oracle Application Server Wireless middle-tier home. However, if the IASCONFIG_LOC environment variable is set, the location defined by this variable is used.

In a single computer OracleAS Portal and OracleAS Wireless installation, where OracleAS Web Cache and Oracle Internet Directory instances already reside on the same computer, entries to the Portal Dependency Settings file are created as shown in Example A–2:

Example A–2 Single Computer OracleAS Portal and OracleAS Wireless Installation

```
<IASConfig XSDVersion="1.0">

   <IASInstance Name="iAS-1.abc.company.com" Host="abc.company.com">
      <OIDComponent AdminPassword="welcome1" PortSSLEnabled="false"
LDAPPort="3002" AdminDN="cn=orcladmin"/>
   </IASInstance>

   <IASInstance Name="iAS-2.abc.company.com" Host="abc.company.com">
      <WebCacheComponent ListenPort="3002" InvalidationPort="3003"
InvalidationUsername="invalidator" InvalidationPassword="welcome1"
SSLEnabled="false"/>
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
   </IASInstance>

   <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="welcome1" ConnectString="xyz.company.com:1521:s901dev3">
      <WebCacheDependency ContainerType="IASInstance"
Name="iAS-2.abc.company.com"/>
      <OIDDependency ContainerType="IASInstance" Name="iAS-1.abc.company.com"/>
      <EMDependency ContainerType="IASInstance" Name="iAS-2.abc.company.com"/>
   </PortalInstance>

</IASConfig>
```

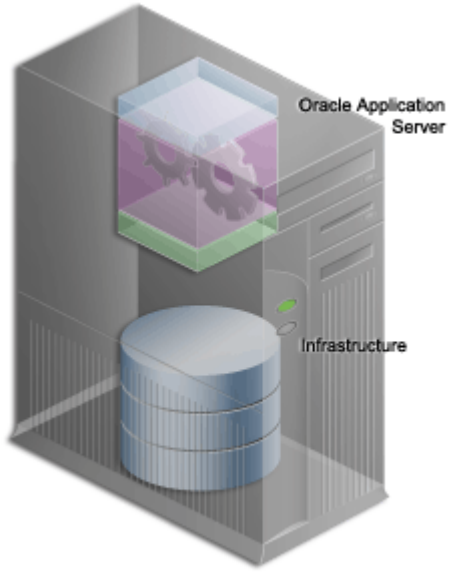
A.2.6 Common Configuration Mapping in the Portal Dependency Settings File

This section shows what the Portal Dependency Settings file looks like in the recommended topologies.

OracleAS Portal and OracleAS Wireless Developer Configuration: Medium Sized Computers

The topology for this common configuration is seen in Figure A–2.

Figure A–2 OracleAS Portal and OracleAS Wireless Developer Configuration



This configuration assumes that both the application server and the infrastructure are installed on the same computer, called **Host 1**.

When you install OracleAS Portal and OracleAS Wireless on **Host 1** in `ias.host1.xyz.com`, referencing the Oracle Internet Directory instance in `infra.host1.xyz.com`, the Portal Dependency Settings file looks like Example A–3:

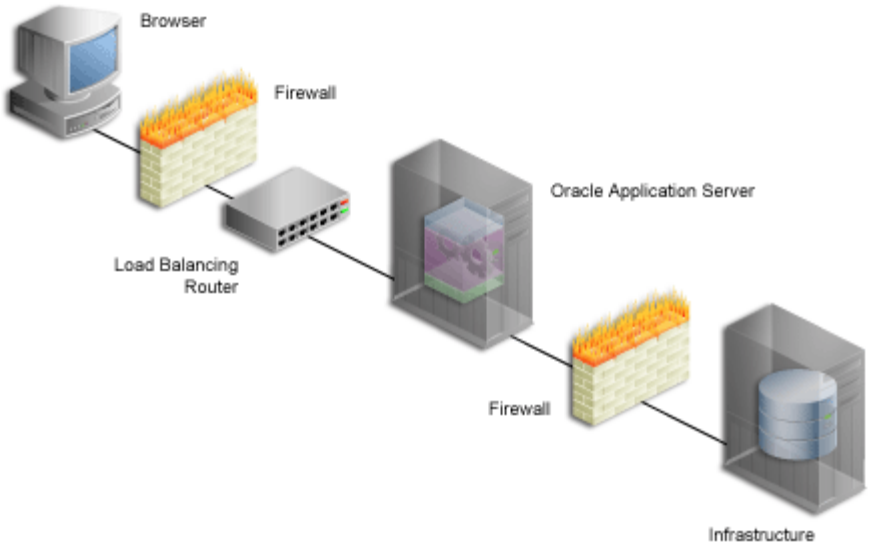
Example A–3 OracleAS Portal and OracleAS Wireless Developer Configuration

```
<IASConfig XSDVersion="1.0">

   <IASInstance Name="ias.host1.xyz.com" Host="host1.xyz.com">
      <WebCacheComponent ListenPort="7778" InvalidationPort="3003"
InvalidationUsername="invalidator" InvalidationPassword="welcome1"
SSLEnabled="false"/>
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
   </IASInstance>

   <IASInstance Name="infra.host1.xyz.com" Host="host1.xyz.com">
      <OIDComponent AdminPassword="welcome1" PortSSLEnabled="false"
LDAPPort="3002" AdminDN="cn=orcladmin"/>
   </IASInstance>

   <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="welcome1" ConnectString="host1.xyz.com:1521:iasdb">
      <WebCacheDependency ContainerType="IASInstance" Name="ias.host1.xyz.com"/>
      <OIDDependency ContainerType="IASInstance" Name="infra.host1.xyz.com"/>
      <EMDependency ContainerType="IASInstance" Name="ias.host1.xyz.com"/>
   </PortalInstance>

</IASConfig>
```

Enterprise Data Center Configuration: Multiple Departments Sharing the Same Data Center

The topology for this common configuration is seen in Figure A–3.

Figure A–3 Enterprise Data Center Configuration



This configuration assumes that the application server and the infrastructure are installed on different computers.

As shown in Figure A–3, the OracleAS Web Cache cluster front-ending OracleAS Portal is not yet known. When you install the application server (OracleAS Portal and OracleAS Wireless installation) on host **Host 1**, referencing the Oracle Internet Directory on host **Host 2**, the configuration will look like Example A–4:

Example A–4 Enterprise Data Center Configuration

```
<IASConfig XSDVersion="1.0">

   <IASInstance Name="infra.host2.xyz.com" Host="host2.xyz.com">
      <OIDComponent AdminPassword="welcome1" PortSSLEnabled="false"
LDAPPort="3002" AdminDN="cn=orcladmin"/>
   </IASInstance>

   <IASInstance Name="ias.host1.xyz.com" Host="host1.xyz.com">
      <WebCacheComponent ListenPort="7778" InvalidationPort="3003"
InvalidationUsername="invalidator" InvalidationPassword="welcome1"
SSLEnabled="false"/>
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
   </IASInstance>

   <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="welcome1" ConnectString="host1.xyz.com:1521:iasdb">
      <WebCacheDependency ContainerType="IASInstance" Name="ias.host1.xyz.com"/>
      <OIDDependency ContainerType="IASInstance" Name="infra.host2.xyz.com"/>
      <EMDependency ContainerType="IASInstance" Name="ias.host1.xyz.com"/>
   </PortalInstance>

</IASConfig>
```

If you want the application server on **Host 1** to be front-ended by OracleAS Web Cache, you need to manually edit the Portal Dependency Settings file. First, remove the existing OracleAS Web Cache entry and then create an OracleAS Web Cache entry that belongs to a farm. The modified Portal Dependency Settings file will now look like Example A–5:

Example A–5 Enterprise Data Center Configuration - Front-Ended by OracleAS Web Cache

```
<IASConfig xmlns="http://www.oracle.com/iAS/iASConfigFile" XSDVersion="1.0">

   <IASInstance Name="infra.host2.xyz.com" Host="host2.xyz.com">
      <OIDComponent AdminPassword="welcome1" PortSSLEnabled="false"
LDAPPPort="3002" AdminDN="cn=orcladmin"/>
   </IASInstance>

   <IASFarm name="Farm_1" host="frontend.xyz.com">
      <WebCacheComponent ListenPort="7778" InvalidationPort="3003"
InvalidationUsername="invalidator" InvalidationPassword="welcome1"
SSLEnabled="false"/>
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
   </IASFarm>

   <IASInstance Name="ias.host1.xyz.com" Host="host1.xyz.com">
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
   </IASInstance>

   <PortalInstance DADLocation="/pls/portal" SchemaUsername="portal"
SchemaPassword="welcome1" ConnectString="host1.xyz.com:1521:iasdb">
      <WebCacheDependency ContainerType="IASFarm" Name="Farm_1"/>
      <OIDDependency ContainerType="IASInstance" Name="infra.host2.xyz.com"/>
      <EMDependency ContainerType="IASInstance" Name="ias.host1.xyz.com"/>
   </PortalInstance>

</IASConfig>
```

The OracleAS Portal instance now references the virtual OracleAS Web Cache front-ending it.

B

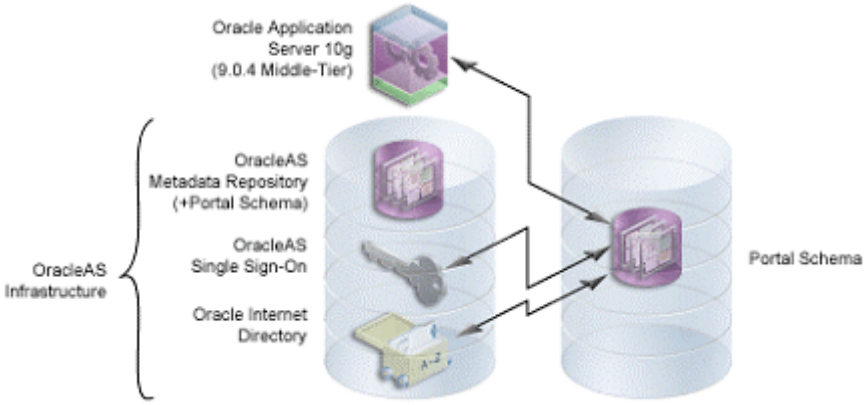
Configuring and Managing an Upgraded Oracle Application Server Portal Instance

Components of Oracle Application Server 10*g* Release 3 (10.1.3) are, by default, only aware of a OracleAS Portal schema in the Oracle Application Server Metadata Repository. The portal schema, in which Oracle Application Server Portal metadata is stored, is an indivisible component of the OracleAS Metadata Repository in a default installation of Oracle Application Server, or in a customer database installation performed by using the Oracle Application Server Repository Creation Assistant (OracleAS RepCA).

However, if you have a portal schema that has been upgraded from an earlier release and is not located in an OracleAS Metadata Repository, you will not be able to take advantage of some of the new management services for Oracle Application Server 10*g* Release 3 (10.1.3). Instead, you must perform some manual configuration steps. These limitations, however, do not affect the run time of the portal, they only affect the manner in which you administer and change the configuration of the portal. This appendix discusses how to configure such a portal.

The following illustration shows a typical deployment topology in which an Oracle Application Server 10*g* Release 3 (10.1.3) middle tier uses the services of a portal schema, which is in turn configured to use the services of Oracle Identity Management.

Figure B–1 Typical Deployment Topology



The database on the left contains the OracleAS Metadata Repository, which contains an empty default portal schema. The database on the right contains another portal schema, which is not contained in an OracleAS Metadata Repository. It is the management of the latter schema that is discussed in this appendix.

This situation is typically the result of either upgrading a Portal 3.0.9 to Oracle Application Server 10*g* Release 3 (10.1.3), or the upgrade of Oracle9*i*AS Portal (9.0.2) that has been installed in a customer database and then upgraded to Oracle Application Server 10*g* Release 3 (10.1.3). In many cases the Database Access Descriptor (DAD) name of the portal is not "portal", which is the default DAD name for OracleAS Portal installed in an OracleAS Metadata Repository.

Note: To maintain forward compatibility when creating a portal in a customer database, you must now use OracleAS RepCA. In doing so, your schema will be part of an OracleAS Metadata Repository.

B.1 Configuring and Managing the OracleAS Portal Instance

The following sections detail the additional steps that are required to manage your OracleAS Portal instance when its schema resides outside the OracleAS Metadata Repository. The areas where additional configuration is required are:

- Changing the OracleAS Portal Schema Password
- Changing Oracle Identity Management Services
- Updating the Oracle Enterprise Manager 10g targets.xml File
- Updating iasconfig.xml When the Database Containing the Portal Schema Has Been Reconfigured
- Performing Advanced Configurations with ptlconfig

WARNING: Before you perform any of the configuration tasks in the preceding list, you must update the Portal Dependency Settings file `iasconfig.xml` with information about your portal. See section "Updating iasconfig.xml" to follow instructions.

Updating iasconfig.xml

To enable the use of the `ptlconfig` tool and the Application Server Control for managing an OracleAS Portal instance, you have to create an entry for the upgraded portal in Portal Dependency Settings file `iasconfig.xml`. To do this, perform the following steps:

Normally, configuring the OracleAS Single Sign-On data using the `ptlconfig` tool, to get the OracleAS Single Sign-On information, involves the use of OracleAS Metadata Repository . This works well for portals that are installed out-of-the-box. However, upgraded portals are not registered with the OracleAS Metadata Repository. In such a case, the OracleAS Single Sign-On information will have to be obtained from the OracleAS Single Sign-On server by querying the LDAP directory. To do this, the Oracle Internet Directory SSL port is required to make a connection to the LDAP store.

Perform the following steps to update the `iasconfig.xml` file with the Port information:

1. Run the `ptlconfig` tool as follows:

```
ptlconfig -load -schema <schema username> -pw <schema password> -conn <connect
string> [-lp ldap_ssl_port]
```

   For a description of the parameters used in the preceding script, refer to Table B–1.

Table B–1 ptlconfig Parameters

Parameter	Description
-load	Creates and updates entries in `iasconfig.xml` with the configuration settings for a specific portal schema.
-schema	Name of the portal schema.
-pw	Portal schema password.
-conn	Connect string to the portal schema.

The `iasconfig.xml` file has an optional property named LDAPSSLPort for the OIDDependency element, as shown in Example B–1. This property contains the LDAP SSL port value, which is used to configure OracleAS Single Sign-On details for upgraded OracleAS Portal instances.

Example B–1 Example LDAPSSLPort Entry

```
<PortalInstance DADLocation="/pls/portal30" SchemaUsername="portal30"
SchemaPassword="welcome1" connectString="dbserver.company.com:1521:orcl">
      <WebCacheDependency ContainerType="IASInstance"
Name="midtier.abc.company.com"/>
      <OIDDependency ContainerType="IASInstance" LDAPSSLPort="4339"
Name="infra.xyz.company.com"/>
      <EMDependency ContainerType="IASInstance"
Name="midtier.abc.company.com"/>
</PortalInstance>
...
<IASConfig>
```

2. Open the `iasconfig.xml` file located by default in *ORACLE_HOME*`/portal/conf`, where *ORACLE_HOME* is the OracleAS Portal and Oracle Application Server Portal and Wireless middle-tier home.
3. Update the `LDAPSSLPort` property with the correct Oracle Internet Directory SSL port information.

B.1.1 Changing the OracleAS Portal Schema Password

Typically, you use Application Server Control to change the OracleAS Portal schema password, but in the case of the portal instance whose schema resides outside the OracleAS Metadata Repository, you have to change the portal schema password using SQL*Plus.

Follow these steps to change schema passwords directly in the database:

1. Connect to the database as a user with SYSDBA privileges.
2. Enter the following command:

```
SQL> ALTER USER <schema> IDENTIFIED BY <new_password>;
```

For example, to change the PORTAL30 schema password to "abc123":

```
SQL> ALTER USER PORTAL30 IDENTIFIED BY abc123;
```

Next, update the schema password in the Portal Dependency Settings file, `iasconfig.xml` as follows:

1. Locate the file `iasconfig.xml`. By default, it is located in the following directory:

MID_TIER_ORACLE_HOME/portal/conf

> **Note:** The environment variable IASCONFIG_LOC is sometimes used to specify another directory for the location of this file.

2. Edit the `iasconfig.xml` file, by changing the value specified for the `SchemaPassword` attribute to your new password. The following example shows the changes made to the `PortalInstance` element in bold text:

```
<PortalInstance DADLocation="/pls/portal"
SchemaUsername="portal" SchemaPassword="abc123"
connectString="dbserver.company.com:1521:orcl">
   <WebCacheDependency ContainerType="IASInstance"
Name="midtier.abc.company.com"/>
   <OIDDependency ContainerType="IASInstance"
Name="infra.xyz.company.com"/>
   <EMDependency ContainerType="IASInstance"
Name="midtier.abc.company.com"/>
</PortalInstance>
```

3. Encrypt the plain text password that you added in `iasconfig.xml` by running the following command:

```
ptlconfig -encrypt
```

> **Note:** You do not have to run `ptlconfig` in another mode at this time.

After this, you have to update the Database Access Descriptor (DAD) with the new password. Perform the following steps to update the DAD:

1. Display the OracleAS Portal home page in Application Server Control Console. See Section 7.3, "Using Application Server Control Console to Monitor and Administer OracleAS Portal" for more information.
2. Navigate to the Application Server instance in which you want to update the DAD.
3. Select **HTTP Server** from the System Components table.
4. Click **Administration**.
5. Click **PL/SQL Properties**.
6. In the DADs section, select the DAD you want to update.
7. In the Database Connectivity Information section, change the password in the Password field.
8. Click OK.
9. Restart the Oracle HTTP Server.
10. Restart the OC4J_Portal instance.

    To do this, navigate to the Application Server instance, select the **OC4J_Portal** check box, and click **Restart**.

This will update the `dads.conf`, `oradav.conf`, and `targets.xml` files.

In addition, for a portal schema that resides outside Oracle Application Server Metadata Repository, you must also update the password change in the `targets.xml` file. This file is used by the Oracle Enterprise Manager 10*g* monitoring technology to effectively monitor the OracleAS Portal target.

To update the password change in the `targets.xml` file, perform the following steps:

1. Back up the *ORACLE_HOME*`/sysman/emd/targets.xml` file.
2. Open the *ORACLE_HOME*`/sysman/emd/targets.xml` file using a text editor.
3. Search for the OracleAS Portal target for which `TYPE="oracle_portal"`, and the portal_DAD value corresponds to the OracleAS Portal instance for which the password is being updated.
4. Update the `portalDatabaseSchemaPassword` property as shown in the following example:

```
<Property NAME="portalDatabaseSchemaPassword" VALUE="abc123"
ENCRYPTED="False"/>
```

> **Note:** If the password is updated, then you must enter the updated value in plain text and set `ENCRYPTED` to `FALSE`. These properties will be encrypted again on reloading the targets file.

5. Save the `targets.xml` file, and then reload the targets file by using the following command:

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

If OracleAS Portal is monitored using an Oracle Enterprise Manager 10*g* Grid Control Central Agent also, then these steps must also be performed in the Grid Control Central Agent Oracle home.

For Grid Control, the `targets.xml` file is located in the Oracle home containing the Grid Control Central Agent:

ORACLE_HOME`/sysman/emd/targets.xml`

B.1.2 Changing Oracle Identity Management Services

If your portal is using the Oracle Identity Management services from the Oracle Application Server Infrastructure that is registered with the Oracle Application Server middle tier, then additional steps need to be performed whenever changes are made to the Oracle Identity Management services.

You may, for example, need to change the configuration if the administrative password of Oracle Internet Directory changes.

Normally, you make these changes using the Application Server Control, but in this case, you have to modify `iasconfig.xml` manually.

Perform the following steps to update `iasconfig.xml` with a new Oracle Internet Directory host name or port number:

1. Locate the file `iasconfig.xml`. By default it is located in the following directory:

```
MIDDLE_TIER_ORACLE_HOME/portal/conf
```

> **Note:** To maintain forward compatibility when creating a portal in a customer database, you must now use OracleAS RepCA. In doing so, your schema will be part of an OracleAS Metadata Repository.

2. Edit the `iasconfig.xml` file, by changing the values specified for the `OIDComponent` attributes in the IASInstance element. The following example shows the changes made to the `IASInstance` element in bold text:

```
<IASInstance Name="infra.xyz.company.com" Host="xyz.company.com"
Version="10.1.2">
      <OIDComponent AdminPassword="welcome9" PortSSLEnabled="false"
LDAPPort="3002" AdminDN="cn=orcladmin"/>
      <EMComponent ConsoleHTTPPort="1814" SSLEnabled="false"/>
</IASInstance>
```

3. Encrypt any plain text passwords in `iasconfig.xml` by running the following command:

```
ptlconfig -encypt
```

4. Update the portal schema with the new configuration settings in `iasconfig.xml`, by running the following command:

```
ptlconfig -dad <portal_dad_name> -site -oid
```

   Where `portal_dad_name` is your portal DAD name.

   For example:

```
ptlconfig -dad portal30 -site -oid
```

B.1.3 Updating the Oracle Enterprise Manager 10*g* targets.xml File

OracleAS Portal can be monitored using either the Oracle Enterprise Manager 10*g* Application Server Control or the Oracle Enterprise Manager 10*g* Grid Control.

The Application Server Control is included when you install Oracle Application Server. From the OracleAS Portal perspective, this is an administration console for Oracle Application Server.

The Grid Control provides full enterprise management and extensive notification and alerting services.

You need to update the Oracle Enterprise Manager 10*g* `targets.xml` file for the following reasons:

- To enable monitoring of OracleAS Portal using Oracle Enterprise Manager 10*g*

  Application Server Control and Grid Control each use information stored in their own `targets.xml` file to manage and monitor their targets. To manage and monitor your portal instance, you must edit the `targets.xml` file manually.

- To update any database connection changes in a nondefault portal

  If you have a nondefault portal instance whose schema resides outside OracleAS Metadata Repository, every time there is a change in the database connection information you must update the relevant properties in the `targets.xml` file.

To edit the `targets.xml` file, perform the following steps for the Application Server Control and then repeat these steps for the Grid Control if you have installed it.

1. Locate the `targets.xml` file.

For the Application Server Control, it is located in the Oracle home of the Oracle Application Server middle tier:

MID_TIER_ORACLE_HOME/sysman/emd/targets.xml

For the Grid Control, it is located in the Oracle home containing the Grid Control Central Agent:

ORACLE_HOME/sysman/emd/targets.xml

Note: You cannot use the Target Creation user interface in Grid Control to create a portal target.

2. Make a backup copy of the `targets.xml` file before you modify it.
3. Copy and paste the following template for the new target entry in the `targets.xml` file:

```
<Target TYPE="oracle_portal" NAME="%Name%"
DISPLAY_NAME="%DisplayName%" VERSION="1.0">
 <Property NAME="version" VALUE="%Version%" />
 <Property NAME="OracleHome" VALUE="%OracleHome%" />
 <Property NAME="PortalListeningHostPort"
VALUE="%PortalListeningHostPort%" />
 <Property NAME="HTTPMachine" VALUE="%HTTPMachine%" />
 <Property NAME="HTTPPort" VALUE="%HTTPPort%" />
 <Property NAME='HTTPProtocol' VALUE="%protocol%" />
 <Property NAME="portal_DAD" VALUE="%portal_DAD%" />
 <Property NAME="portalDatabaseMachineName" VALUE="%Database Host%" />
 <Property NAME="portalDatabasePort" VALUE="%Database Port%" />
 <Property NAME="portalDatabaseSchemaPassword" VALUE="%password%"
ENCRYPTED="FALSE" />
 <Property NAME="portalDatabaseSchemaUsername" VALUE="%portal schema name%" />
 <Property NAME="OidRepSchemaName" VALUE="SIDFormat/ServiceNameFormat" />
 <!-- If OidRepSchemaName is set to SIDFormat: -->
 <Property NAME="portalDatabaseSID" VALUE="mySID" />
 <!-- If OidRepSchemaName is set to ServiceNameFormat: -->
 <Property NAME="portalDatabaseServiceName" VALUE="myServiceName" />
 <Property NAME="startComponent"
VALUE="%iasName%.%machine%_OC4J_Portal"/>
 <Property NAME='PPESuccessfulResponsesCriticalThreshold'
VALUE='80' />
 <Property NAME='PPESuccessfulResponsesWarningThreshold' VALUE='90'
/>
 <Property NAME='modplsqlSuccessfulResponsesCriticalThreshold'
VALUE='80' />
 <Property NAME='modplsqlSuccessfulResponsesWarningThreshold'
VALUE='90' />
 <Property NAME='portletResponseCriticalThreshold' VALUE='4500' />
 <Property NAME='portletResponseWarningThreshold' VALUE='4000' />
 <CompositeMembership>
   <MemberOf TYPE="oracle_ias"
NAME="%iasName%.%machine%.%domain%"/>
 </CompositeMembership>
</Target>
```

4. Edit the properties in the portal target and add your instance-specific property values. Table B–2 and Table B–4 list the relevant `targets.xml` properties and their descriptions.

Table B–2 Relevant targets.xml Properties for Monitoring OracleAS Portal

Attribute	Description	Examples
NAME	The target name, using the following format: %iasName%.%machine_name%._Portal:%DAD name%	my1012PW.machine1.abc.com_Portal:portal30
DISPLAY_NAME	The target display name visible in the user interface. Format: Portal:%DAD name%	Portal:portal30
version	The portal version number. Ensure that you update the version number to reflect the middle-tier version number, so the correct version of the portal middle tier is monitored. See Table B–3 for more information.	10.1.2.0.2
OracleHome	The Oracle home path.	/u01/app/oracle/product/my1012PW
PortalListeningHostPort	The entry point to the portal excluding the DAD information.	http://machine1.abc.com:7778
HTTPMachine	The computer name of the middle-tier computer running Oracle HTTP Server.	machine1.abc.com
HTTPPort	Oracle HTTP Server port.	7778
HTTPProtocol	Oracle HTTP Server protocol.	http or https
portal_DAD	The portal DAD.	portal30

Table B–3 shows the values that you must use for the version attribute described in Table B–2, depending on the version of the OracleAS Portal middle tier on which the targets reside.

Table B–3 Version Number Mapping for the version Attribute

Version Number of the OracleAS Portal Middle Tier	Value to Enter for the version Attribute
9.0.2.*x*	9.0.2
9.0.4 and 9.0.4.*x*, except 9.0.4.1	9.0.4
9.0.4.1 and 9.0.4.1.*x*	9.0.4.1
10.1.2.0.0 and 10.1.2.0.1	10.1.2.0.0
10.1.2.0.2	10.1.2.0.2
Versions later than 10.1.2.0.2	The exact version number

Table B–4 Relevant targets.xml Properties for Updating Database Connection Changes

Attribute	Description	Example
portalDatabaseMachineName	Host name of the computer running the database in which OracleAS Portal is installed.	www.abc.com
portalDatabasePort	Port Number.	1521
portalDatabaseSchemaPassword	The OracleAS Portal database account password. The password is typically set at installation but you can change it by specifying a new password. **Note**: If the password is updated, then you must enter the updated value in plain text and set ENCRYPTED to FALSE. Then on reloading the targets file these properties will be encrypted again.	portal30
portalDatabaseSchemaUserName	The OracleAS Portal database account username.	portal30
OidRepSchemaName	The connect string format. The following two connect string formats are supported: ■ SIDFormat ■ ServiceNameFormat In the case of a nondefault portal repository, this property is not used for the portal schema name. It is used to indicate the type of connection to the portal repository. For example, SIDFormat or ServerNameFormat.	SIDFormat
portalDatabaseSID	The SID of the database running the nondefault portal schema. Set this property only if OidRepSchemaName is set to SIDFormat. Use this format when the connect string is in the format, host:port:sid. For example: myhost.oracle.com:1521:mydb	mySID
portalDatabaseServiceName	The Service Name of the database running the nondefault portal schema. Set this property only if OidRepSchemaName is set to ServiceNameFormat. Use this format when the connect string is in the format, host:port:service_name. For example: www.abc.com:1521:www.xyz.com	myServiceName

Suppose you have an existing target portal and you now want to add a portal called portal30, which uses a SIDFormat connect string, you must add the template and then make the changes shown in bold text:

```
<Target TYPE="oracle_portal"
NAME="my1012PW.machine1.abc.com_Portal:portal30"
DISPLAY_NAME="Portal:portal30" VERSION="1.0">
 <Property NAME="version" VALUE="10.1.2.0.2"/>
 <Property NAME="OracleHome" VALUE="/u01/app/oracle/product/my1012PW"/>
 <Property NAME="PortalListeningHostPort"
VALUE="http://machine1.abc.com:7778"/>
 <Property NAME="HTTPMachine"
VALUE="machine1.abc.com"/>
 <Property NAME="HTTPPort" VALUE="7778"/>
<Property NAME="HTTPProtocol" VALUE="http"/>
 <Property NAME="portal_DAD" VALUE="portal30"/>
 <Property NAME="portalDatabaseMachineName" VALUE="www.abc.com" />
 <Property NAME="portalDatabasePort" VALUE="1521" />
 <Property NAME="portalDatabaseSchemaPassword"
VALUE="4c958c5c661cabd683d42b2e663da358" ENCRYPTED="TRUE" />
 <Property NAME="portalDatabaseSchemaUsername" VALUE="portal30" />
 <Property NAME="OidRepSchemaName" VALUE="SIDFormat" />
 <Property NAME="portalDatabaseSID" VALUE="mySID"/>
 <Property NAME="startComponent"
VALUE="my1012PW.machine1.abc.com_OC4J_Portal"/>
 <Property NAME="PPESuccessfulResponsesCriticalThreshold" VALUE="80"/>
 <Property NAME="PPESuccessfulResponsesWarningThreshold" VALUE="90"/>
 <Property NAME="portletResponseCriticalThreshold" VALUE="4500"/>
 <Property NAME="portletResponseWarningThreshold" VALUE="4000"/>
 <Property NAME="modplsqlSuccessfulResponsesCriticalThreshold"
VALUE="80"/>
 <Property NAME="modplsqlSuccessfulResponsesWarningThreshold" VALUE="90"/>
 <CompositeMembership>
   <MemberOf TYPE="oracle_ias" NAME="my1012PW.machine1.abc.com"/>
 </CompositeMembership>
</Target>
```

5. Navigate to the *ORACLE_HOME*`/bin` directory of the Application Server Control or the Grid Control of which you have just edited the `targets.xml` file and run the following command:

   ```
   emctl reload
   ```

6. Using a Web browser, navigate to the Oracle Enterprise Manager 10*g* Application Server Control Console (or Grid Control) to verify your new target.

   For details, see Section 7.2.1, "Accessing the Application Server Control Console" (or Section 7.1, "Using the Grid Control Console").

   The new portal instance should appear in the list of **System Components**.

B.1.4 Updating iasconfig.xml When the Database Containing the Portal Schema Has Been Reconfigured

If you have reconfigured the database containing the portal schema such that the connection details for the database (for example, the server location, TNS port, or SID) have changed then you will need to edit the entry for the reconfigured portal in Portal Dependency Settings file `iasconfig.xml`. To do this, perform the following steps:

1. Locate the `targets.xml` file.

   For the Application Server Control, it is located in the Oracle home of the Application Server middle tier:

   MID_TIER_ORACLE_HOME`/sysman/emd/targets.xml`

For the Grid Control, it is located in the Oracle home containing the Grid Control Central Agent:

ORACLE_HOME/sysman/emd/targets.xml

> **Note:** You cannot use the Target Creation user interface in Grid Control to create a portal target.

2. Edit the `iasconfig.xml` file and change the `connectString` for the portal instance to be reconfigured. The following example shows an updated entry:

```
<PortalInstance DADLocation="/pls/portal30"
SchemaUsername="portal30" SchemaPassword="welcome1"
connectString="abc.company.com:1521:db901dev">
   <WebCacheDependency ContainerType="IASInstance"
Name="midtier.abc.company.com"/>
   <OIDDependency ContainerType="IASInstance"
Name="infra.xyz.company.com"/>
   <EMDependency ContainerType="IASInstance"
Name="midtier.abc.company.com"/>
</PortalInstance>
```

3. Update the OracleAS Metadata Repository with the new configuration settings in `iasconfig.xml`, by running the following command:

```
ptlconfig -dad <dad_name> -wc -oid -em -site
```

Where `dad_name` is the name of the portal DAD.

> **Note:** You cannot use the instructions for changing the Metadata Repository used by a middle-tier instance described in the *Oracle Application Server Administrator's Guide*, because the portal schema is not contained in an OracleAS Metadata Repository.

B.1.5 Performing Advanced Configurations with ptlconfig

When making advanced configuration changes by using the `ptlconfig` command the documentation examples in this guide often use the default DAD name portal. For example, you may be instructed to run the `ptlconfig` tool as follows:

```
ptlconfig -dad portal
```

Remember to always substitute your actual DAD name if it is not "portal" (For example, portal30).

B.1.6 Conclusion

Upgraded OracleAS Portal instances whose schema resides outside of an OracleAS Metadata Repository require manual steps to perform a few configuration changes that an OracleAS Metadata Repository contained portal instance is able to complete by using the graphical user interface of Oracle Enterprise Manager 10*g*. The additional steps outlined in this appendix can be performed with a minimum of additional effort to achieve the same configuration and management operations.

C

Using OracleAS Portal Installation and Configuration Scripts

After installing OracleAS Portal as part of the Oracle Application Server installation, several scripts are available for post-installation configuration.

The specific topics covered in this appendix include:

C.1 Managing the Invalidation Message Processing Job Using cachjsub.sql

OracleAS Portal uses caching to improve its performance. One type of caching it uses is the invalidation-based caching. In invalidation-based caching, OracleAS Portal caches various objects (pages, portlets, and so on) for a set amount of time. When these objects are requested, they are retrieved from the cache, if available; otherwise they are regenerated from the Oracle Application Server Metadata Repository. The cache for these objects will expire when the *maxcache* time has been reached, or when the objects are explicitly invalidated (expired) by invalidation messages.

OracleAS Portal uses invalidation messages when it needs to expire objects in the cache. Invalidation messages are categorized as hard and soft invalidations. Hard invalidations take effect immediately, that is, the objects that they intend to invalidate expire from cache immediately. Soft invalidations take effect when they are processed

by the invalidation processing job. The frequency by which the invalidation job executes is configurable. This is done using the `cachjsub.sql` script.

To change the execution frequency of the invalidation processing job:

1. Locate the following directory:

   ORACLE_HOME`/portal/admin/plsql/wwc`

2. On the database where the portal schema is installed, log on to SQL*Plus with the appropriate user name and password for that schema.

   For example:

   ```
   sqlplus portal/portal
   ```

3. Enter the following command to update the execution frequency of the invalidation job:

   ```
   SQL> @cachjsub.sql <start_time> <start_time_fmt> <interval_mins>
   ```

   `cachjsub.sql` takes three parameters:

   - *start_time* is either when the first job should be run or START.
   - *start_time_fmt* is the Oracle date format model to be applied to the value of *start_time*. Refer to the database documentation library for more information about the Oracle date format model.
   - *interval_mins* is how many minutes each run is scheduled apart.

   Note: If START is provided for the first parameter, the second parameter is ignored, and it will default the start time to the current time.

   Example 1:

   ```
   SQL> @cachjsub.sql START null 120
   ```

   Example 2:

   ```
   SQL> @cachjsub.sql '02-22-2005 7:30' 'MM-DD-YYYY HH:MI' 1440
   ```

   Example 3:

   ```
   SQL> @cachjsub.sql '06-14-2005 15:30' 'MM-DD-YYYY HH24:MI' 60
   ```

   Note: Example 3 shows time in the 24-hour format.

C.2 Configuring for IP Check During Session Cookie Validation

As part of the process of validating the session cookie of a user's request (even if that user is PUBLIC), OracleAS Portal can be configured to perform a comparison between the IP address stored in the cookie with the IP address of the current client. Only if the two values are the same will OracleAS Portal consider the request legitimate.

By default, OracleAS Portal does not perform this check. When a proxy exists between the user's client and the portal, the IP address stored in the session cookie is that of the proxy, and not that of the client.

Depending on the network configuration into which the Oracle Application Server is installed, it may be possible to enable IP checking in cookie validation, for added security.

To change the state of IP checking in cookie validation, use SQL*Plus to update data in the portal schema as detailed in Table C–1.

Table C–1 Enabling and Disabling the IP Check

	Portal Schema	SSO Schema

Enable IP Checking	`update wwsec_enabler_config_info$`
	`set url_cookie_ip_check = 'Y';`
	`commit;`
		`update wwsec_enabler_config_info$`
		`set url_cookie_ip_check ='Y';`
		`update wwsso_ls_configuration_info$`
		`set cookie_ip_check = 'Y';`
		`commit;`

Disable IP Checking	`update wwsec_enabler_config_info$`
	`set url_cookie_ip_check = 'N';`
	`commit;`
		`update wwsec_enabler_config_info$`
		`set url_cookie_ip_check ='N';`
		`update wwsso_ls_configuration_info$`
		`set cookie_ip_check = 'N';`
		`commit;`

C.3 Using the secupoid.sql Script

By default, OracleAS Portal connects to Oracle Internet Directory using LDAP without SSL. If the Oracle Internet Directory server is configured for an SSL port, though, OracleAS Portal can be configured to use LDAP over SSL, also known as LDAPS.

See Also: *Oracle Internet Directory Administrator's Guide*

To configure OracleAS Portal to use SSL to connect to Oracle Internet Directory, you must run the `secupoid.sql` script. This script enables you to change the following OracleAS Portal configuration parameters related to Oracle Internet Directory:

- Oracle Internet Directory host name
- Oracle Internet Directory port
- application Oracle Internet Directory password
- SSL setting

When you install OracleAS Portal, it is automatically configured to use an Oracle Internet Directory server. However, you may want to change some settings, such as whether to use SSL, after installation. To change to an SSL connection for Oracle Internet Directory, simply run the *ORACLE_HOME*`/portal/admin/plsql/wwc/secupoid.sql` script in the PORTAL schema to specify the LDAPS port instead of the LDAP port, and indicate that you want to use SSL.

Running the secupoid.sql Script

This section shows a sample execution of `secupoid.sql` from SQL*Plus.

In the example, Oracle Internet Directory was initially configured to run LDAP on port `389`. Later, an LDAPS port was activated on `636`. Because the server name does not change, we retain the old value, update the port, and indicate that we want to use SSL by setting the `Use SSL?` value to `Y`. When you run the script, it displays the current configuration and lets you replace any of the configurable settings. The script also enables you to update OracleAS Portal's Oracle Internet Directory cache after running it. Because activating SSL does not change any of the Oracle Internet Directory information cached by OracleAS Portal, it is not usually necessary to refresh the cache in this case.

```
SQL> @secupoid
Current Configuration
--------------------
OID Host: oid.domain.com
OID Port: 389
Application DN:
orclApplicationCommonName=PORTAL.040820.123756.096286000,cn=Portal,cn=Products,cn=OracleContext
Application Password: 3E8C2D1B87CB61011757239C5AA9B390
Use SSL? N

PL/SQL procedure successfully completed.

Updating OID Configuration Entries
Press [Enter] to retain the current value for each parameter
For SSL Connection to LDAP, specify "Y"es or "N"o
-----------------------------------------------
Enter value for oid_host:
Enter value for oid_port: 636
Enter value for app_password:
Enter value for use_ssl_to_connect_to_ldap: Y
Enter value for refresh_with_new_settings: N

PL/SQL procedure successfully completed.

No errors.
```

After executing the script, OracleAS Portal is configured for LDAPS access of Oracle Internet Directory.

When `secupoid.sql` is run, it can change the port number of Oracle Internet Directory stored in the portal schema of the OracleAS Metadata Repository. Running `secupoid`, however, does not change the values stored in `iasconfig.xml`. You must manually update the **LDAPPort** and **PortSSLEnabled** attributes in the **OIDComponent** element in `iasconfig.xml` so that subsequent runs of `ptlconfig` will not reverse the new settings. Refer to Section A.2, "Portal Dependency Settings File" for more information on the Portal Dependency Settings File.

See Also: *Oracle Application Server Security Guide*

C.4 Using the secjsdom.sql Script

If you have your Oracle Internet Directory and OracleAS Portal servers residing in different domains, you must explicitly set the JavaScript domain for OracleAS Portal such that it can resolve user and group lists of values. To do this, you must use the `secjsdom.sql` script located in the directory *`ORACLE_HOME`*`/portal/admin/plsql/wwc`.

Suppose your installation has OracleAS Portal configured to use an Oracle HTTP Server other than Oracle Delegated Administration Services. In this situation, you must have a common domain, so that the values can be transferred from the list of values displayed by Oracle Delegated Administration Services to the page displayed by OracleAS Portal.

To create a common domain for this scenario, follow the steps in Example C–1:

Example C–1 Creating a Common Domain

1. Log in to SQL*Plus as PORTAL.
2. Run the following SQL script:

```
SQL> @secjsdom.sql <domain_name>
```

If, in Example C–1, the Oracle Delegated Administration Services servlet is running on a computer `infra.abc.com` and OracleAS Portal is running on a computer `portal.abc.com`, then the `secjsdom.sql` script should be invoked like this:

```
@ SQL> @secjsdom.sql abc.com
```

Performing this procedure enables you to run Oracle Internet Directory lists of values from OracleAS Portal in either Netscape, or Internet Explorer. When using lists of values, a transit window is displayed in addition to the list of values itself. The transit window is required to pass values to OracleAS Portal without forcing pages to reset their domain.

To reset a common domain that was defined, run the `secjsdom.sql` script as shown in Example C–2:

Example C–2 Resetting a Previously Defined Common Domain Using secjsdom.sql

1. From your operating system command prompt, go to the following directory:

```
DESTINATION_MID_TIER_ORACLE_HOME/portal/admin/plsql/wwc
```

2. Using SQL*Plus, connect to the OracleAS Portal schema as the owner and run the following commands:

```
@secjsdom ''
commit;
```

See Also: *Oracle Application Server Security Guide*

C.5 Configuring the Portal Session Cookie

OracleAS Portal uses a session cookie to maintain session state for portal applications. For portal to work correctly, the client browser must be configured to accept cookies from the server. Upon installation, the portal session cookie has a default name, scope, and security that are set appropriately for most installations. This section describes these defaults, and how they can be changed if needed.

C.5.1 Configuring the Cookie Name

By default the portal's session cookie is named `portal` after the default Database Access Descriptor (DAD) used to access the portal schema. You can use Oracle Enterprise Manager 10*g* to change the cookie name, if it needs to explicitly be set to something else. To do this, you must access the **DAD Edit** page in the Oracle

Enterprise Manager 10*g* Application Server Control Console. This page is located under **mod_plsql services** of the OracleAS Portal middle-tier component. The cookie name can be set on the **Document Alias and Session Parameter** page. To change the name of the cookie, provide the desired name in the **Session Cookie Name** field of the Session Cookie section.

C.5.2 Configuring the Scope of the Cookie

In cases where you want access to the same portal from two middle tiers at the same time, or if you want to open the portal cookie domain as required by the PL/SQL Adapter functionality, you must define the scope of the OracleAS Portal session cookie to be sent to all the middle-tier servers involved in the architecture. By default, the session cookie's domain is scoped to the host from which it was generated. The path for the cookie is set to "/".

Note: You should make these changes when there is no traffic on the portal, otherwise existing sessions will experience session errors (ORA-20000) after you change the session cookie name.

For example, if the cookie was generated from `www.company.com`, then the cookie domain is `www.company.com`. However, let's say that another server, `portal.company.com` is also a middle-tier server that needs access to that session cookie. Then the cookie domain would need to be widened so that the `portal.company.com` server can also see the cookie.

Follow these steps to modify the scope of the portal session cookie:

1. Locate the following directory:

   ORACLE_HOME/portal/admin/plsql/wwc

2. On the database where your OracleAS Portal schema is installed, log on to SQL*Plus as the portal schema. For example:

   ```
   sqlplus portal/portal_pwd
   ```

3. Enter the following command:

   ```
   SQL> @ctxckupd
   OracleAS Portal
   Current Settings for Portal Session Cookie:
   Cookie Domain : Only send cookie back to originating host:port
   Set Cookie as Secure: Y
   Enter the domain for the session cookie: .company.com
   Should cookie be flagged as secure for HTTPS sessions? (Y/N): N
   Settings changed to
   Cookie Domain : .company.com
   Do not set cookie as secure. (N)
   SQL>
   ```

   This enables you to set the cookie domain for the session cookie. In this example, the cookie domain is set to `.company.com`.

Notes:

- If you want to use different listeners or keep the session cookie throughout different domains, specify a Cookie Domain to be the host name only. For example, if you access OracleAS Portal from two computers:

  — `machine1.us.company.com:3000`

  — `machine2.us.company.com:4000`

  When running `ctxckupd.sql`, set the cookie domain to `.us.company.com`.

- The cookie domain also determines the scope of the NLS_LANGUAGE cookie, which is a persistent cookie that determines the user's preferred language. This NLS_LANGUAGE cookie is set when selecting languages in the set language portlet.

C.5.3 Securing the Cookie

In this release of OracleAS Portal, the script `ctxckupd.sql` contains an additional option, `Set Cookie as Secure`.

The default location for this script is *ORACLE_HOME*`/portal/admin/plsql/wwc`. When you run this script, you see the following output:

```
SQL> @ctxckupd
OracleAS Portal
Current Settings for Portal Session Cookie:
Cookie Domain : Only send cookie back to originating host:port
Set Cookie as Secure: Y
Enter the domain for the session cookie...
Leave blank to scope to originating host:
Should cookie be flagged as secure for HTTPS sessions? (Y/N): N
Settings changed to
Cookie Domain : Only send cookie back to originating host:port
Do not set cookie as secure. (N)
SQL>
```

Set Cookie as Secure indicates that the cookie should be sent back to the server if the request is over an HTTPS connection only. This setting ensures that the session cookie is not transmitted over an insecure connection when it needs to be protected. By default, this option is set to *Yes* and is sufficient for most deployments.

In some cases, you may need to set the **Set Cookie as Secure** option to **No**. For example, if your portal is accessed over both HTTP and HTTPS and you want the session cookie to be shared across both protocols (possible if they are running on the default ports `80` (HTTP) and `443` (HTTPS)). In this instance, when **Set Cookie as Secure** is set to **No**, the same cookie produced over an HTTPSrequest, is sent over any subsequent HTTP requests.

C.6 Managing the Session Cleanup Job

OracleAS Portal and OracleAS Single Sign-On perform session management similar to other Web-based applications. Sessions are tracked with cookies. Session information is stored in a table in the OracleAS Portal and OracleAS Single Sign-On schema. When a user logs out, the session information is marked inactive. A DBMS job subsequently cleans up the inactive rows.

The session table accumulates a number of rows that are flagged as active. When a user shuts down the browser instead of logging out, the row is "active", even though it is not actually in use. The cleanup job cleans up the active rows that are older than a specified duration.

When OracleAS Portal is installed, a DBMS job is installed to perform session cleanup of the session table, `WWCTX_SSO_SESSION$`. The cleanup job is set to run every 24 hours. The first scheduled cleanup occurs 24 hours after the installation of the job.

When the job runs, it deletes all inactive sessions and all sessions marked active (`WWCTX_SSO_SESSION$.ACTIVE = 1`), that are older than 7 days (`WWCTX_SSO_SESSION$.SESSION_START_TIME < sysdate - 7`).

These default settings can be modified by running some job management scripts in the OracleAS Portal schema to manage portal sessions, or in the OracleAS Single Sign-On schema to manage OracleAS Single Sign-On sessions. They utilize the same session management infrastructure.

Follow these steps to obtain the current cleanup job information:

1. Locate the following directory:

   ORACLE_HOME/portal/admin/plsql/wwc

2. On the database where the OracleAS Portal or OracleAS Single Sign-On schema is installed, log in to SQL*Plus with the appropriate user name and password for that schema.

   For example:

   ```
   sqlplus portal/portal
   ```

3. Enter the following command to get the current job information:

   ```
   SQL> @ctxjget
   ```

   The command results in the display of the currently installed job information, as returned by the `DBMS_JOB` package:

   ```
   The session cleanup job is job ID 7381
   dbms_job.isubmit(job=>7381,what=>'begin execute immediate''begin
   wwctx_sso.cleanup_sessions(p_hours_old => 168); end;''; exception when
   others then null; end;',next_date=>to_date('2001-04-17:14:07:20',
   'YYYY-MM-DD:HH24:MI:SS'),interval=>'SYSDATE + 24/24',no_parse=>TRUE);

   PL/SQL procedure successfully completed.
   ```

The results indicate which procedure is executed, what parameters are passed to it, and when the next invocation is to occur. This particular example indicates that the job is to clean up active sessions that are a week old (168 hours). It also indicates that the next scheduled job execution is on 4/17/2001 at 5:14 pm, and the job should run every 24 hours thereafter.

If the job execution must be modified, either to adjust the age of sessions that should be deleted, or to increase or decrease the frequency of cleanup, you can run the `ctxjsub.sql` script to submit modified execution parameters.

Follow these steps to submit modified job execution parameters:

1. Locate the following directory:

   ORACLE_HOME/portal/admin/plsql/wwc

2. On the database where the OracleAS Portal or OracleAS Single Sign-On schema is installed, log on to SQL*Plus with the appropriate user name and password for that schema. For example:

```
sqlplus portal/portal
```

3. Enter the following command to submit new cleanup job information:

```
@ctxjsub <hours_old> <start_time> <time_format> <interval_hours>
```

Table C–2 lists the `ctxjsub` parameters.

Table C–2 ctxjsub Parameters

Parameter	Description
`hours_old`	The age of an active session that should be deleted.
`start_time`	The time that the next job should run.
`time_format`	The time format string that specifies how `start_time` is formatted.
`interval_hours`	The amount of time, in hours, between runs of the cleanup job.

For example:

```
SQL> @ctxjsub 200 '04/17/2001 10:00' 'MM/DD/YYYY HH24:MI' 12
```

The job information is displayed, similar to:

```
Created path for job id.
DBMS_JOB id = 7381
Cleanup job updated. Job ID = 7381

PL/SQL procedure successfully completed.
```

The cleanup job submission script can be run any number of times to modify the execution parameters. Each invocation updates the job information associated with the job ID for the cleanup job. This job ID is maintained in the preference store so that the job information is updated instead of submitting multiple jobs.

You can also specify a `start_time` of START, in which case, the `time_format` parameter is ignored, but you still need to pass it a value (such as NOW). The result is to run the job `<interval_hours>` hours from now:

```
SQL> @ctxjsub 168 START NOW 24
```

This submits the job as it does in the installation.

If you want the cleanup job to execute immediately, then obtain the job ID by calling `ctxjget.sql`. Once you know the job ID, you can execute the job by issuing the following command in the product schema:

```
SQL> exec dbms_job.run(7381);
```

In the preceding example, 7381 is the job ID returned by the call to `ctxjget.sql`. When you execute a job in this manner, the next automated invocation of the job occurs at `interval_hours` after this manual invocation. To run the job on the original schedule, resubmit the `start_time` desired using `ctxjsub.sql`.

C.7 Timing and Caching Statistics

All OracleAS Portal pages can be run in a special mode in which timing and caching information is displayed. If you want to see this debug information on every page you can set the Parallel Page Engine Parameter `showPageDebug` to `true` in the `web.xml` file.

> **See Also:** Appendix D, "Configuring the Parallel Page Engine"

If you want to see the debug information for just a few select pages and portlets, you can control the logging level by the `_debug` URL parameter. For example, to see the timing statistics for the following OracleAS Portal page:

```
http://abc.com/servlet/page?_pageid=21
```

You can manually insert `&_debug=3`

To make:

```
http://abc.com/servlet/page?_pageid=21&_debug=3
```

Possible values for `_debug` are *0*, *1*, *2*, *3*, *4*, and *5*.

Values greater than *1* will potentially raise the **logmode** value for the duration of the request, and trigger all request log messages to be echoed into the page response.

> **Note:** All values greater than *0* cause `_debug=1` to be propagated in back end requests.

Table C–3 shows the results of `_debug` values:

Table C–3 _debug Values for Timing and Caching Statistics

Value	Timing and Caching Statistics?	Flag Forwarded to Providers? (as value 1)	logmode Raised to a Minimum of	Log Messages Written to Page Response?
0	Yes	-	-	-
1	Yes	Yes	-	-
2	Yes	Yes	debug	Yes
3	Yes	Yes	request	Yes
4	Yes	Yes	content	Yes
5	Yes	Yes	parsing	Yes

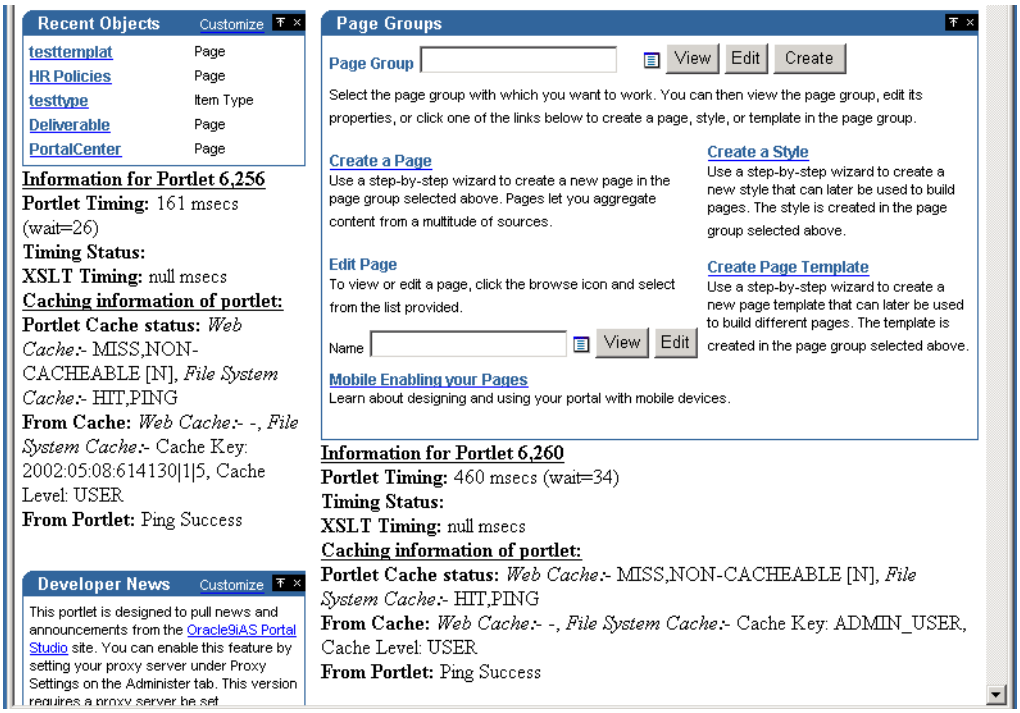
`urlDebugMode` and `urlDebugUsers` are additional parameters that can be used to restrict the use of `_debug` on a URL. See Appendix D, "Configuring the Parallel Page Engine" for more information.

The following statistics are available when the portal page is run in debug mode:

- Portlet Statistics
- Page Statistics
- Additional Summary Statistics

Figure C–1 shows a page that is running in the `_debug=0` mode:

Figure C–1 Portal Page Running in Debug Mode



C.7.1 Portlet Statistics

In Figure C–1, you can see a number of Portlet related statistics listed under each portlet. Each Portlet has a unique internal reference identification number. This number is used in the "Information for Portlet" summary. For the portlet in the top left corner of Figure C–1, you can see that this number is 6256.

For each portlet the following statistics are listed:

C.7.1.1 Portlet Timing Information

- **Portlet Timing** (msecs) (wait msecs)

  Indicates how many milliseconds it took to retrieve the portlet, and how long the request was queued, also in milliseconds.

- **Timing Status**

  This is deprecated and no longer in use.

- **XSLT Timing** (msecs)

  Displays the number of milliseconds that were needed to retrieve the XSL style sheet, in case the portlet is an XML portlet.

C.7.1.2 Portlet Caching Information

- **Portlet Cache status** Web Cache (values) File System Cache (values)

  This is the Cache status from both OracleAS Web Cache and the portal cache.

  Valid values for OracleAS Web Cache are:

  - MISS, or NEW [M] indicating a cache miss in OracleAS Web Cache and that the content that is generated by the portlet is new.

- MISS, or STALE [G] indicating a cache miss, due to stale content in OracleAS Web Cache.
- HIT [H] indicating an OracleAS Web Cache hit.

Valid values for File System Cache are:

- HIT_PING indicating a cache hit for a validation-based portlet.
- HIT_EXPIRES indicating a cache hit for an expiry-based portlet.
- MISS_STALE indicating a cache miss due to stale content in the Cache. This applies to both expiry, and validation-based portlets.
- MISS_NEW indicating a cache miss and that the content that is generated by the portlet is new. This applies to both expiry, and validation-based portlets.

If a portlet uses the File System Cache, then the information mentioned in the preceding text will be listed. Otherwise it will be null.

If there is a hit on OracleAS Web Cache, no details about File System Cache will be displayed because the content is served directly out of OracleAS Web Cache. Additionally, if a portlet does not use OracleAS Web Cache, then no Web Cache information will be printed.

- **From Cache:Web Cache** Cache Expires (seconds), Age in Cache (secs), File System Cache (values).

  Information from both OracleAS Web Cache and File System Cache will be printed here based on the type of caching that the portlet uses.

  See Also: *Oracle Application Server Web Cache Administrator's Guide*

  "Cache Expires" lists the number of seconds after which the portlet content in OracleAS Web Cache will expire.

  "Age in Cache" lists the number of seconds that the portlet content has been Cached in OracleAS Web Cache.

  "File System Cache" displays the information obtained from the File System Cache about Cache Key, Cache Expiry and about the Cache Level in case of a cache hit, with the Cache Status of either HIT_PING, or HIT_EXPIRES.

  In case of a cache hit, the Cache Key and Cache Level (for Validation-based portlets) and Cache Expires and Cache Level (for expiry-based portlets) are displayed, with the Cache Status value of either HIT_PING or HIT_EXPIRES.

  For Validation-based and Expires-based portlets, "None" is printed when there is a cache miss due to the portlet content being new. (Cache Status: MISS_NEW) The portlet is contacted to get the new Cache Key, Cache Expiry and Cache Level.

  For Validation-based portlets, if the content in the Cache has become stale resulting in a cache miss, the current values in the cache for Cache Key and Cache Level are displayed. In this case, the portlet is contacted to get the updated Cache Key and the level (Cache Status: MISS_STALE).

  For Expires-based portlets, when the content in the cache has become stale resulting in a cache miss, a value of INVALID in the Expires field and Cache Level are displayed. In this case, the portlet is contacted to get the updated Cache Expiry and Cache Level (Cache Status: MISS_STALE).

- **From Portlet**: (Cache Key) (Cache Level)

This is the information obtained from the portlet about File System Cache Key, Cache Expiry, and Cache Level when there is a cache miss and when portlet is contacted for the updated, or new values (Cache Status: MISS_NEW, or MISS_ STALE). Note that there is no OracleAS Web Cache related information displayed in this section.

For Validation-based portlets, when there is a cache hit and if the ping is successful, meaning the content in the Cache is still valid, then the portlet does not return a new Cache Key and Cache Level; instead it will indicate that the cache is still valid. In this case, "Ping Success" is displayed (Cache Status: HIT_PING).

For Expires-based portlets, when there is a cache hit and if the content has not expired, then the portlet is not contacted for the content. In this case, "Not contacted" is displayed (Cache Status: HIT_EXPIRES).

Following are a few examples that show different caching scenarios and the resulting output. Note that the other page and portlet related output is not shown here.

Note: In this release, page portlets are requested as portlets, separate from the container page definition. Therefore, portlet and page caching information is displayed for each page portlet in the debug output.

Example C–3 Caching Information Debug Output 1

Portlet Cache: File System Cache, **Caching Type:** Validation-based, **Status:** MISS, STALE.

```
Caching information for portlet:
Portlet Cache status: File System Cache:- MISS,STALE
From Cache: File System Cache:- Cache Key: 42, Cache Level: USER
From Portlet: Cache Key: 44, Cache Level: USER
```

Example C–4 Caching Information Debug Output 2

Portlet Cache: File System Cache, **Caching Type:** Expires-based, **Status:** MISS, NEW.

```
Caching information for portlet:
Portlet Cache status:File System Cache:-  MISS,NEW
From Cache: File System Cache:-None
From Portlet: Cache Expires: 1, Cache Level: USER
```

Example C–5 Caching Information Debug Output 3

Portlet Cache: File System Cache, Web Cache, **Caching Type:** Validation and Invalidation-based, **Status:** MISS, NEW in File System Cache and Web Cache.

```
Caching information for portlet:
Portlet Cache status: Web Cache:- MISS,NEW [M], File System Cache:- MISS,NEW
From Cache: Web Cache:- Cache Expires: 86400 secs, Age in Cache: 0 secs , File
System Cache:- None
From Portlet: Cache Key: 9.0.2.2.1502:04:18:09:19:56, Cache Level: SYSTEM
```

Example C–6 Caching Information Debug Output 4

Portlet Cache: Web Cache, **Caching Type:** Invalidation-based, **Status:** HIT in Web Cache.

```
Caching information for portlet:
Portlet Cache status: Web Cache:- HIT [H]
From Cache: Web Cache:- Cache Expires: 86400 secs, Age in Cache: 58 secs
From Portlet: -
```

C.7.2 Page Statistics

Every page has a unique internal reference identification number, similar to the portlets on the page, shown in Figure C–1.

For the page, the following statistics are listed:

- **Elapsed Time** (msecs)

  This is the total amount of time required to generate the page calculated in the Parallel Page Engine (PPE). The actual generation time in the browser can be higher, due to network overhead.

  Elapsed time is made up of page meta WAIT time and Stream time. Page meta WAIT time is the time taken to wait on content through an HTTP connection. Stream time is the time taken streaming and assembling the content pieces. Stream time is in turn composed of the following elements:

  - Page meta time
  - Time waiting for portlets to complete
  - Time taken streaming content to the browser

  Effectively, elapsed time is the total amount of time (in milliseconds) that it takes to put the page together, from the time the request was received to the last byte being written to the browser.

- **Page meta-time** (msecs) (wait = msecs)

  Displays the time that it takes to retrieve the page meta data. The wait time (msecs) represents how long the request was queued.

- **Page meta Cache Status** (Web Cache values), (Cache Expires msecs), (Age in Cache msecs), (File System Cache values)

  Represents the cache status from both OracleAS Web Cache and portal cache. Valid values for OracleAS Web Cache are MISS, or NEW and HIT. Valid values for portal cache are HIT, or PING, and MISS, or STALE. The Web Cache Expires value and the Age in Cache are both measured in milliseconds.

- **Login meta-time** (msecs) (wait msecs)

  Displays the time (in milliseconds) that it takes to retrieve the login meta data. The wait time represents the total amount of time (in milliseconds) that the request spends in the request queue.

- **Login meta Cache Status**

  Similar to **Page meta Cache Status** mentioned earlier, represents the cache status for the login meta data from both Web Cache and portal cache.

C.7.3 Additional Summary Statistics

- **Stream info** (msecs)

  Represents (in milliseconds) how long it takes for the page to stream to the browser.

- **processing** (msecs)

  Processing time (in milliseconds) for streaming.

- **write** (msecs)

  The write lines can repeat several times. The lines represent each physical buffer write to the stream itself. This are one set for each buffer write.

- **flush** (msecs)

  The flush logs indicate that the writing stream was flushed. This is logged to keep track of the number of network round trips.

C.8 Using the cfgiasw Script to Configure Mobile Settings

If you want to change portal's references to OracleAS Portal or OracleAS Wireless' portal service URLs, you must use the `cfgiasw.pl` script to manually update the references. The script file is located here:

ORACLE_HOME`/assistants/opca/`

Running the script without parameters will print its usage to the screen, which is shown next:

Usage:

```
perl cfgiasw.pl -s portal_schema
     -w ias wireless url
     -h portal home page url
     -c connect_string
```

Table C–4 Oracle Application Server Wireless Configuration Parameters

Parameter	Description
-s	Oracle Database schema for OracleAS Portal database objects. Default = `PORTAL`
-w	The URL of the Oracle Application Server Wireless gateway for mobile requests to OracleAS Portal. This parameter is not mandatory (no default). The value for this parameter must be enclosed in double quotation marks.
-h	The URL of the OracleAS Portal home page. This is used within portal to determine the character set of the OracleAS Portal middle tier. This information is required when creating an Oracle Application Server Wireless service This parameter is not mandatory (no default). The value for this parameter must be enclosed in double quotation marks.
-c	Connect string for database (no default).

Note:

- Ensure that you are using the Perl version that is available as part of the Oracle Application Server installation, by setting the path variable as follows:

  For Windows:

  ```
  PATH ORACLE_HOME\perl\5.6.1\bin\MSWin32-x86
  ```

  For Solaris or Linux:

  ```
  PATH ORACLE_HOME/perl/bin
  ```

- While running the `cfgiasw` script you are prompted for the password. Specify the portal schema password for the script to proceed.

For non-hosted Portals, the OracleAS Wireless' Portal service URL reference can be set in the **Mobile** tab of the **Global Settings** page, except the URL of the OracleAS Portal home page, which can only be set using the `cfgiasw` script.

This script is used to set references to both the OracleAS Wireless Portal Service URL and the OracleAS Portal home page URL, in OracleAS Portal. It can be used in a hosted environment to set the URL references, and will affect all subscribers, because this information is not configured separately for each subscriber. For example:

```
perl cfgiasw.pl -s portal -c portal_db -w "http://<iaswhost>:<port>/ptg/rm?PAoid=$wireless_service_id"
```

In the preceding example, if a mobile device makes a request to the OracleAS Portal directly without being mediated by an Oracle Application Server Wireless server, OracleAS Portal redirects the client to the URL specified here. This URL should be the OracleAS Portal's service URL on the Oracle Application Server Wireless server, in the form:

```
http://<host>:<port>/ptg/rm?PAoid=<service_id>
```

If this setting is blank, then mobile client requests made directly to OracleAS Portal receive an HTTP status indicating that their request is not supported.

See Section 4.6, "Configuring Mobile Support in OracleAS Portal" for configuring other mobile settings in OracleAS Portal.

C.9 Using the cfgxodnc.pl Script to Change the Mobile Device Component of the Cache Key

The cache key used by OracleAS Portal is composed of numerous components. One of these components is based on the URL, and another is based on the OracleAS Wireless header, `X-Oracle-Device.Class`. These components allow portlet content to be cached based on the class of the mobile device used. Examples of device classes include pcbrowser, pdabrowser, microbrowser, and so on.

You can enable portlet content to be cached based on the name of a specific device rather than the device class. To do this, the `X-Oracle-Device.Class` header in the device component of the cache key must be replaced with the `X-Oracle-Device.Name` header.

To ensure that OracleAS Portal works properly with portlet content that is cached based on the value of the `X-Oracle-Device.Name` header, you must do the following:

- Enable OracleAS Portal to use this header. Refer to Section C.9.1, "Adding the PlsqlCGIEnvironmentList Parameter to the dads.conf File" for the steps to be performed.
- Disable caching *or* configure OracleAS Portal to cache content based on the `X-Oracle-Device.Name` header. To configure OracleAS Portal to cache portlet content based on the `X-Oracle-Device.Name` header, you must perform the following tasks:
  - Section C.9.2, "Running the cfgxodnc.pl script"
  - Section C.9.3, "Adding the useDeviceNameCacheKeys parameter to the PPE Configuration file"
  - Section C.9.4, "Clearing Cached Data"

  See Also: *Oracle Application Server Web Cache Administrator's Guide* for the procedure to disable caching of portal content

C.9.1 Adding the PlsqlCGIEnvironmentList Parameter to the dads.conf File

To enable OracleAS Portal to use the `X-Oracle-Device.Name` header, you must add a new parameter, `PlsqlCGIEnvironmentList`, to the `dads.conf` file for the Oracle Application Server instance. To edit the `dads.conf` file, perform the following steps:

1. Open the `dads.conf` file located in the following directory:

   ORACLE_HOME`/Apache/modplsql/conf/dads.conf`

2. Add the following entry to the file:

   ```
   PlsqlCGIEnvironmentList HTTP_X_ORACLE_DEVICE_NAME
   ```

3. Save the `dads.conf` file.

   Note: It is recommended that you edit the `dads.conf` file using Application Server Control Console.

   If you manually edit the `dads.conf` file, then you must add the necessary mod_rewrite and mod_oc4j directives to the `httpd.conf` and `mod_oc4j.conf` files respectively. To do this, perform the steps mentioned in Section E.2, "DAD Configuration File (dads.conf)" using the Application Server Control Console.

4. Run the following command to restart Oracle HTTP Server:

   MID_TIER_ORACLE_HOME`/opmn/bin/opmnctl restartproc type=ohs`

If you now try to access OracleAS Portal using a mobile device, then content will be rendered based on the mobile device used. The cache will also have an increased capacity.

C.9.2 Running the cfgxodnc.pl script

To enable OracleAS Portal to use the `X-Oracle-Device.Name` header, run the `cfgxodnc.pl` script in the `on` mode. This script is available at the following location:

ORACLE_HOME/assistants/opca/

Usage:

```
perl cfgxodnc.pl -s portal_schema
     -c portal_connect_string
     -on|-off
```

To run this script, you must specify all the parameters.

Table C–5 The cfgxodnc Script Parameters

Parameter	Description
-s	Oracle Database schema for OracleAS Portal database objects. Default = PORTAL
-c	Connect string for database (no default).
-on/-off	Option to enable or disable use of X-Oracle-Device.Name in the cache key.

Note: Ensure that you are using the Perl version that is available as part of the Oracle Application Server installation, by setting the path variable as follows:

For Windows:

```
PATH ORACLE_HOME\perl\5.6.1\bin\MSWin32-x86
```

For Solaris or Linux:

```
PATH ORACLE_HOME/perl/bin
```

Note: While running the cfgxodnc.pl script you are prompted for the password. Specify the portal schema password for the script to proceed.

The following is an example showing the usage of the cfgxodnc.pl script to enable the X-Oracle-Device.Name header:

```
perl cfgxodnc.pl -s PORTAL -c portal_database -on
```

The cache size increases when the X-Oracle-Device.Name header is used in the device component of the cache key. If you revert to using the X-Oracle-Device.Class header, then the cache size decreases again.

You can revert to using the X-Oracle-Device.Class header in the device component of the cache key by running the cfgxodnc.pl script in the off mode.

C.9.3 Adding the useDeviceNameCacheKeys parameter to the PPE Configuration file

To use device names instead of device classes when building cache keys, add the useDeviceNameCacheKeys parameter to the PPE configuration file by performing the following steps:

1. Open the web.xml file, located in the directory *MID_TIER_ORACLE_HOME*/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF.

2. Add the `useDeviceNameCacheKeys` parameter as shown in bold face in the following example:

```
<servlet>
<servlet-name>page</servlet-name>
   <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
          . . .
          <init-param>
             <param-name>useDeviceNameCacheKeys</param-name>
             <param-value>true</param-value>
          </init-param>
          . . .
</servlet>
```

3. Save the `web.xml` file.
4. Run the following command to synchronize the manual configuration changes:

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

5. Run the following commands to restart your Oracle Application Server instance:

```
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl stopall
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl startall
```

C.9.4 Clearing Cached Data

To ensure that new cache keys are built based on the device name, you must clear all cached data, both in OracleAS Web Cache and OracleAS Portal File System Cache. Refer to Section 5.8.3, "Managing Portal Content Cached in OracleAS Web Cache" for information about clearing data cached in OracleAS Web Cache. Refer to Section 4.5.5, "Clearing the Portal Cache" for information about clearing data cached in the File System Cache.

C.10 Using the Category and Perspective Scripts

To ensure that all new category and perspective pages are created without errors and that all existing category and perspective pages display their associated items and pages as expected, you must run the category and perspective scripts.

The scripts required are:

```
ORACLE_HOME/portal/admin/plsql/wws/pstdefin.sql
ORACLE_HOME/portal/admin/plsql/wws/pstpgshw.sql
ORACLE_HOME/portal/admin/plsql/wws/pstundef.sql
ORACLE_HOME/portal/admin/plsql/wws/pstpgcre.sql
ORACLE_HOME/portal/admin/plsql/wws/pstprcpg.sql
```

To run these scripts:

1. Delete the current category or perspective templates.
2. Connect to OracleAS Portal using SQL*Plus as the portal schema user.
3. Configure the `pstdefin.sql` file with:
   - Page group information. You can re-create the pages in a single page group, several page groups or all page groups.
   - Page information. You can re-create category pages only, perspective pages only, or both.

Descriptions for these settings are in the `pstdefin.sql` file. If necessary, use the script `pstpgshw.sql` to retrieve information from OracleAS Portal to configure the `pstdefin.sql` file.

4. Run the script `pstpgcre.sql` to apply the changes. For example:

```
SQL> @pstpgcre.sql
```

If a template exists in the page group when the new pages are created, new category and perspective pages are created based on that template. If you delete the template before running the scripts or the template is missing, then a new template is created in the page group and the new pages are based on this template.

C.11 Using the PDK-Java Preference Store Migration and Upgrade Utility

A preference store is a mechanism for storing information like user preference data, portlet and provider settings, or even portlet data, while using OracleAS Portal. Currently, PDK-Java has two `PreferenceStore` implementations, `DBPreferenceStore` and `FilePreferenceStore`. `DBPreferenceStore` persists data using a JDBC compatible relational database and `FilePreferenceStore` persists data using the file system.

This utility allows users to migrate existing data between different preference stores (for example, from `FilePreferenceStore` to `DBPreferenceStore`) and to upgrade from previous releases of PDK-Java and OracleAS Portal to manage portlet preference data generated by existing portlets. The tool allows upgrading users to ensure that their existing locale-specific portlet preference data uses a naming format compatible with the latest PDK and OracleAS Portal releases.

If you have already installed OracleAS PDK, you can manage the information stored in the preference store by using the Preference Store Migration and Upgrade Utility, which is included in the `pdkjava.jar` file. For a complete description of the syntax of the utility, use the following command:

```
java -classpath ORACLE_HOME/portal/jlib/pdkjava.jar
oracle.portal.provider.v2.preference.MigrationTool
```

You can run the Preference Store Migration and Upgrade Utility in either of the following modes based on the `-mode` you select while running the command:

- Upgrade mode
- Migration mode

> **Note:** After running the utility, it is recommended that you restart OC4J_Portal and Oracle HTTP Server to ensure that the latest preference store information is used.

Upgrade Mode

Use an upgrade mode to upgrade data in place, and to modify existing locale-specific preferences in the preference store so that the naming format used is compatible with the current version of OracleAS Portal and a given `localePersonalizationLevel` setting.

Table C–6 describes the upgrade modes in which you can run the utility.

Table C–6 Upgrade Modes in Which to Run the Utility

Mode	Description
`file`	Use when data in a `FilePreferenceStore` must be upgraded.
`db`	Use when data in a `DBPreferenceStore` must be upgraded.

An upgrade mode can be used in the following scenarios:

- You have upgraded from OracleAS PDK 9.0.4.0.0 or earlier and want to use existing portlets with the default `localePersonalizationLevel` setting of `language` (In earlier releases, the default setting was `locale`).
- You have upgraded from OracleAS Portal 9.0.2.0.0 or earlier and want to use existing portlets with a `localePersonalizationLevel` setting of `locale` (OracleAS Portal now uses different names for some locales and therefore some existing data must be remapped).
- You want to change the `localePersonalizationLevel` for an existing portlet from `locale` to `language` or vice-versa.

When using an upgrade mode, you must use the `-remap` option to specify the `localePersonalizationLevel` (`language` or `locale`) that you are upgrading to. You can also use the `-countries` option to specify a prioritized list of ISO country codes, indicating your order of preference in case of a collision between remapped preferences for different countries. For example, if you specify `-remap language -countries GB,US` in the command, it means that if the utility comes across both US English and British English preferences (`en_US` and `en_GB`) in a given preference store, it will remap the British English preference to become the English-wide preference (`en`).

Note: While running the utility in `db` mode, for the `pref1User` and `pref1password` properties, use the values specified in the JDBC connection definition in the `<j2ee-home>/config/data-sources.xml` file.

Migration Mode

Use a migration mode to copy data from a source preference store to a target preference store. When the utility is run in this mode, the preference stores for all the portlet definitions are updated.

Table C–7 describes the migration modes in which you can run the utility.

Table C–7 Migration Modes in Which to Run the Utility

Mode	Description
`filetodb`	Use when data must be copied from a `FilePreferenceStore` to a `DBPreferenceStore`.
`filetofile`	Use when data must be copied from one `FilePreferenceStore` to another `FilePreferenceStore` that is in a different location.
`dbtofile`	Use when data must be copied from a `DBPreferenceStore` to a `FilePreferenceStore`.

Table C–7 (Cont.) Migration Modes in Which to Run the Utility

Mode	Description
`dbtodb`	Use when data must be copied from one `DBPreferenceStore` to another `DBPreferenceStore` that is based on a different database table.

When using a migration mode, you can use the `-remap` and `-countries` options to specify that the data should be upgraded in the course of being migrated, that is, locale-specific preferences should be remapped appropriately.

The other options accepted by the utility are used to specify the properties of the preference stores involved in the upgrade or migration process. These options must correspond to the tags you specify in `provider.xml` to describe the preference stores. For more information about the properties you can set on a preference store, refer to the *PDK-Java XML Provider Definition Tag Reference* document on Portal Center:

`http://portalcenter.oracle.com`

Properties beginning with the prefix `-pref1` correspond to properties of the source preference store (in an upgrade mode this is the only preference store). For example, specifying `-pref1UseHashing true -pref1RootDirectory j2ee/home/applications/jpdk/jpdk/WEB-INF/providers/sample` would set the `useHashing` and `rootDirectory` properties of a source FilePreferenceStore.

Note: If you installed the Oracle Application Server Portal Developer Kit on a standalone Oracle Containers for J2EE (OC4J) instance, or if you downloaded the preconfigured standalone OC4J with OracleAS PDK, then the source preference store will be available in the following location:

ORACLE_HOME`/j2ee/home/applications/jpdk/jpdk/WEB-INF/providers/sample`

If you installed OracleAS Portal as part of the Oracle Application Server release, then the source preference store will be available in the following location:

ORACLE_HOME`/j2ee/OC4J_Portal/applications/jpdk/jpdk/WEB-INF/providers/sample`

When one of the migration basic modes is selected, properties beginning with the prefix `-pref2` correspond to properties of the target preference store. For example, specifying `-pref2User portlet_prefs -pref2Password portlet_prefs -pref2URL jdbc:oracle:thin:@myserver.mydomain.com:1521:mysid` would set the database connection details on a target DBPreferenceStore.

Following are some examples to illustrate the usage of the utility:

```
java -classpath $ORACLE_HOME/portal/jlib/pdkjava.jar
 oracle.portal.provider.v2.preference.MigrationTool -mode file -remap language
 -countries GB,US -pref1UseHashing true -pref1RootDirectory
 j2ee/home/applications/jpdk/jpdk/WEB-INF/providers/sample

java -classpath $ORACLE_HOME/portal/jlib/pdkjava.jar
 oracle.portal.provider.v2.preference.MigrationTool -mode filetodb -remap locale
 -countries AR,MX -pref1UseHashing true -pref1RootDirectory
```

```
 j2ee/home/applications/jpdk/jpdk/WEB-INF/providers/sample -pref2User portlet_
prefs -pref2Password portlet_prefs -pref2URL
 jdbc:oracle:thin:@myserver.mydomain.com:1521:mysid
```

C.12 Using the Schema Validation Utility

The Schema Validation Utility (SVU) is used to clean up and report any data inconsistencies in the Portal schema. The SVU performs validations on Page Group objects and DB Provider objects.

Some of the benefits of using the SVU are:

- It prevents OracleAS Portal import from failing due to data inconsistencies between the source and target Portal instances.
- It prevents OracleAS Portal patching from failing. For example, when applying the 9.0.4.1 patch to a 9.0.4.0 version of Portal.

Schema Validation Utility can be run in the following scenarios:

- During the OracleAS Portal Export and Import process.
- When errors such as ORA-1403, ORA-1422, and ORA-4088 are seen when using the OracleAS Portal user interface.

For a more detailed explanation of how the validation is performed, and to download the SVU script, `svu_rept.sql`, log in to Oracle *Metalink* at `http://metalink.oracle.com` and read the article *Schema Validation Utility*. The Doc ID for this article is **286619.1**.

There are two ways of running the Schema Validation Utility, which are:

- Using the Schema Validation Utility with OracleAS Portal Export and Import
- Using the Standalone Schema Validation Utility

C.12.1 Using the Schema Validation Utility with OracleAS Portal Export and Import

In OracleAS Portal Export and Import, the SVU is run automatically in CLEANUP mode by default in the background during export and import, in the following stages:

1. Before exporting - To clean up any data inconsistencies that exist on the source instance.
2. Before importing - To clean up any data inconsistencies that exist on the target instance that could affect the import process.
3. After importing - To clean up any data inconsistencies that may have been introduced by the import process.

C.12.2 Using the Standalone Schema Validation Utility

The SVU can be run in standalone mode when there are data inconsistencies reported or observed. To run the utility in standalone mode, you need to execute the script `svu_rept.sql` as the OracleAS Portal schema owner (PORTAL):

```
SQL> @svu_rept.sql
```

You will be prompted to specify the mode and type to run the script.

Mode:

- **REPORT** - Reports data inconsistencies.

- **CLEANUP** - Cleans up data inconsistencies.

Type:

- **ALL** - Validates both page group objects and DB Provider objects.
- **PAGEGROUP** - Validates page group objects only.
- **DBPROV** - Validates DB Provider objects only.

After you have provided the mode and type, you will be prompted to specify a path to save the log file. You can enter a path like `c:\temp\svu.log` here. Run the SVU in REPORT mode first, before running it in CLEANUP mode.

IMPORTANT:

- **Always take a valid backup of the database before running the SVU in CLEANUP mode.**
- **If you run the SVU in CLEANUP mode and then in REPORT mode, inconsistencies should not be reported. If any inconsistencies are reported, you must contact Oracle Support Services.**

D

Configuring the Parallel Page Engine

The Oracle Application Server Portal architecture is designed around a three-tier architecture that allows any browser to connect to it. This flexible architecture allows each component (browser, Oracle HTTP Server listener, Oracle Database 10*g*, and OracleAS Portal) to be upgraded individually as required.

A part of the OracleAS Portal middle tier, the Parallel Page Engine (PPE) is a servlet that runs under Oracle Containers for J2EE and services page requests. The PPE reads page metadata, calls providers for portlet content, accepts provider responses, and assembles the requested page in the specified page layout.

D.1 Configuring PPE Parameters

When a page is requested from OracleAS Portal, the request is made from the browser to the Oracle HTTP Server listener. The returned page is comprised of many types of portlets. A portlet is an area on a portal page that contains data from a particular data source.

The PPE obtains the page metadata from the Oracle Application Server Metadata Repository and is responsible for assembling the portlets on the page.

D.1.1 Setting PPE Configuration Parameters

Starting from Oracle9*i*AS release 9.0.2 and later, all of the servlets are installed under OC4J, based upon the application deployment. All of the configuration parameters for PPE are entered in the `web.xml` file, that is, in the `<servlet>` element with a `<servlet-name>` value of `page`. In the default installation, this file can be found at the following location:

MID_TIER_ORACLE_HOME`/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF/`

D.1.2 PPE Configuration Settings

Table D–1 describes each of the different configuration parameters available for use with the PPE. Each parameter affects the operation of the PPE in a different manner. Some are simply for logging, while others can affect the performance of the engine or OracleAS Portal itself. In most cases, the default values should be sufficient; however, there may be configurations where this is not the case. Each parameter is described with its syntax, description, and default.

Table D–1 Parallel Page Engine (PPE) Parameters

PPE Setting	Syntax	Description	Default Value

`x509certfile`	`<init-param>`
	`<param-name>x509certfile</param-name>`
	`<param-value>c:\certificates\trustedcerts.txt</param-value>`
	`</init-param>`
		Specifies a file containing a list of certificates to be implicitly trusted by HTTPClient. These certificates are added as trust points to all connections made by HTTPClient using SSL. Once this setting is in use, all SSL connections must be trusted. Otherwise, HTTPClient will throw an exception in the PPE.
		Note that SSL connections are made from the PPE for two reasons, and this configuration affects both:
		loopback requests to the portal, for example, for PMD.
		show calls to Providers.
		Note that the file specified here can be obtained from a wallet by exporting all trusted certificates, but the comments in the resultant file must be removed. Alternatively, it can be created manually.
			trust points not used

`versionOnSplashScreen`	`<init-param>`
	`<param-name>versionOnSplashScreen</param-name>`
	`<param-value>false</param-value>`
	`</init-param>`
		Indicates whether the PPE must display version information on the splash screen.	`false`

`useScheme`	`<init-param>`
	`<param-name>useScheme</param-name>`
	`<param-value>http</param-value>`
	`</init-param>`
		Overrides the scheme (HTTP or https) used when the PPE makes requests to the portal. The default, if not specified, is to always use the page request scheme. Note that you must set the `useScheme` and `usePort` parameters.
		You need to specify these in scenarios where public access is through **https** on **port A**, and you want to set PPE requests to use a faster **http** connection on **port B**.
			Use page request scheme

Table D–1 (Cont.) Parallel Page Engine (PPE) Parameters

PPE Setting	Syntax	Description	Default Value

`usePort`	`<init-param>`
	`<param-name>usePort</param-name>`
	`<param-value>8888</param-value>`
	`</init-param>`
		Overrides the port used when the PPE makes requests to the portal. The default, if not specified, is to always use the page request port. Note that you must set the `useScheme` and `usePort` parameters.
		You need to specify these in scenarios where public access is through **https** on **port A**, and you want to set PPE requests to use a faster **http** connection on **port B**.
			Use page request port

`useDeviceNameCacheKeys`	`<init-param>`
	`<param-name>useDeviceNameCacheKeys</param-name>`
	`<param-value>false</param-value>`
	`</init-param>`
		This key is used to specify whether the mobile device name or device class must be used while building cache keys. The default is for the device class to be used.
		If set to `true`, then the device name is used to build cache keys.
		Refer to Section C.9, "Using the cfgxodnc.pl Script to Change the Mobile Device Component of the Cache Key" for more information.
			`false`

`urlDebugUsers`	`<init-param>`
	`<param-name>urlDebugUsers</param-name>`
	`<param-value>fred,bill,ben</param-value>`
	`</init-param>`
		This is specified to indicate the list of users allowed to use the *_debug* URL parameter, subject to the value restriction in the `urlDebugMode` parameter. If this is not specified, all users can use it subject to the value restriction.
		The format is a comma-delimited list of portal user names, with leading and trailing spaces being ignored.
			none required

Table D–1 (Cont.) Parallel Page Engine (PPE) Parameters

PPE Setting	Syntax	Description	Default Value

`urlDebugMode`	`<init-param>`
	`<param-name>urlDebugMode</param-name>`
	`<param-value>1</param-value>`
	`</init-param>`
		Specifies the highest value of the *_debug* URL parameter that the PPE should honor. Possible values for *_debug* are:
		`none`, `0`, `1`, `2`, `3`, `4`, and `5`
		If a value higher than that allowed is received by the PPE, it is reduced to the highest value permitted, or ignored if no value is allowed.
		The values build incrementally. For example, at debug value `2`, values for debug level `1` and `0` are also recorded.
			`1`

`stall`	`<init-param>`
	`<param-name>stall</param-name>`
	`<param-value>120</param-value>`
	`</init-param>`
		If the response headers are returned, but the data itself lags behind, then a stall comes into affect. This value keeps the PPE from holding on to connections forever. Once the response headers are received, the PPE makes every effort to wait as long as is feasible to retrieve all of the data. Set this value appropriately if the portlets being requested are large, or running over a slow network.
		Note that the upper limit of this parameter should be set to a response time acceptable by a Web user (typically a few seconds).
			65 sec

`showPageDebug`	`<init-param>`
	`<param-name>showPageDebug</param-name>`
	`<param-value>false</param-value>`
	`</init-param>`
		If you set `showPageDebug` to `true`, the Page timing information is shown on every request.
		Refer to Section C.7, "Timing and Caching Statistics" for a description of the timing and caching statistics.
			`false`

Table D–1 (Cont.) Parallel Page Engine (PPE) Parameters

PPE Setting	Syntax	Description	Default Value

`showError`	`<init-param>`
	`<param-name>showError</param-name>`
	`<param-value>true</param-value>`
	`</init-param>`
		When a portlet times out, or something within the PPE goes wrong with a particular portlet request, an error is displayed to the user. The messages tend to be generic, but do give the user some information and an indication that the page did not display as expected. If you set this to `false`, no messages are displayed to the user.	`true`

`resourceUrlKey`	`<init-param>`
	`<param-name>resourceUrlKey</param-name>`
	`<param-value>`*`KEY`*`</param-value>`
	`</init-param>`
		This key is used by the PPE to calculate checksums for URLs that are requested by WSRP and JPDK resource proxying.
		For WSRP resource proxying to work, the key must be set to an alpha-numeric value of 10 characters or more.
		In addition, for JPDK proxying, a JNDI environment variable, also called `resourceUrlKey`, must be set for the provider.
			none

`requesttime`	`<init-param>`
	`<param-name>requesttime</param-name>`
	`<param-value>30</param-value>`
	`</init-param>`
		This is the default time out assigned to portlet requests that do not have their own time out value specified. It is applied as the amount of time (in seconds) allowed before response headers are returned by the server. Time outs are weighted by where they originate. If the portlet sets its own time out value, then that is the time out that is used. If no portlet time out is available, then the provider registration time out is used. If neither of these is present, then the `requesttime` is used.
		Note that the upper limit of this parameter should be set to a response time acceptable by a Web user (typically a few seconds).
			30 sec

Table D–1 (Cont.) Parallel Page Engine (PPE) Parameters

PPE Setting	Syntax	Description	Default Value
queueTimeout	<init-param> <param-name>queueTimeout</param-name> <param-value>10</param-value> </init-param>	The amount of time a request should stay in the queue before being timed out. This parameter can be used if requests for portlets are timing out, but the requests are never being sent. Although this points to other performance problems that could be solved by alternative configurations, this option is available to allow requests to stay in the queue for longer or shorter periods of time.	10 sec
proxyHost proxyPort	<init-param> <param-name>proxyHost</param-name> <param-value>ph.comp.com</param-value> </init-param> <init-param> <param-name>proxyPort</param-name> <param-value>8888</param-value> </init-param>	This is the host name and port number of a proxy server that may be required to request data from the Oracle Application Server. These parameters are only required if a proxy server is in use between PPE and the Oracle Application Server listener.	n/a
poolSize	<init-param> <param-name>poolSize</param-name> <param-value>25</param-value> </init-param>	This represents the number of connections that the PPE is capable of making at any one time. This value can be raised or lowered based upon performance needs. Setting the number higher makes more threads and connections available for use; however, this uses more resources.	25
offlinePath offlinePathMxml	<init-param> <param-name>offlinePath</param-name> <param-value>/path/offline.html</param-value> </init-param> <init-param> <param-name>offlinePathMxml</param-name> <param-value>/path/offline.xml</param-value> </init-param>	By setting either of these, the PPE is set to display the desired off-line message. There are two available messages: one for an HTML browser and one for a mobile enabled device.	null
minTimeout	<init-param> <param-name>minTimeout</param-name> <param-value>5</param-value> </init-param>	This is the minimum timeout allowed to be used by a Portlet. Thus, if the minTimeout is set to 5, and a portlet sends a timeout of 2, the minTimeout value of 5 would be applied to that portlet.	5 sec

Table D–1 (Cont.) Parallel Page Engine (PPE) Parameters

PPE Setting	Syntax	Description	Default Value
maxParallelPortlets	<init-param> <param-name>maxParallelPortlets</param-name> <param-value>20</param-value> </init-param>	Used to specify the maximum number of portlet requests for a given page, that should be allowed, to execute at the same time. Allowed values are: 0 - Indicates no restriction (beyond the number of fetchers available). Any positive integer - Indicates a restriction on simultaneous requests.	20
maxParallelPagePortlets	<init-param> <param-name>maxParallelPagePortlets</param-name> <param-value>20</param-value> </init-param>	Used to limit the number of page portlet requests that are allowed to execute at the same time in the PPE. Allowed values are as follows: 0 - Indicates no restriction (beyond the number of fetchers available). Any positive integer - Indicates a restriction on simultaneous requests.	10

Table D–1 (Cont.) Parallel Page Engine (PPE) Parameters

PPE Setting	Syntax	Description	Default Value
logmode	<init-param> <param-name>logmode</param-name> <param-value>debug</param-value> </init-param>	Enables the PPE to run in debug mode. This mode writes debug information to the PPE log file. This mode does cause some degradation in performance because large amounts of information are being written to disk. The PPE log file (application.log) by default is located at: *ORACLE_HOME*/j2ee/OC4J_Portal/application-deployments/portal/ Allowed values are: none - No debug messages perf - Performance messages only debug - General debug messages request - Details of requests made by the PPE content - Details of the content of requests made by the PPE parsing - Details of metadata parsing all - All debug messages The values build incrementally. For example, at logging level request, the output for logging levels debug and perf will also be recorded.	none - no debug messages
jspSrcAlias	<init-param> <param-name>jspSrcAlias</param-name> <param-value>/PATH</param-value> </init-param>	The Alias for the jsp engine, like /portal/jsp or some other path.	/jsp/
jspRoot	<init-param> <param-name>jspRoot</param-name> <param-value>/JSP PATH/</param-value> </init-param>	The relative path where JSP files for JSP Pages can be found.	jsp
httpsports	<init-param> <param-name>httpsports</param-name> <param-value>433:444</param-value> </init-param>	This is a colon (':') separated list of ports on which OracleAS Portal is configured to use SSL.	null

Table D–1 (Cont.) Parallel Page Engine (PPE) Parameters

PPE Setting	Syntax	Description	Default Value

`enableWebCacheStaticRules`	`<init-param>`
	`<param-name>enableWebCacheStaticRules</param-name>`
	`<param-value>false</param-value>`
	`</init-param>`
		This key is not used if you are running 10*g* Release 2 (10.1.4) of the portal repository, and is provided only for backward compatibility when you use a 10*g* Release 2 (10.1.4) middle tier with an earlier release of the portal repository.
		If you are using an earlier release of the portal repository, then consider the following:
		If set to `false`, PPE includes the `no-store` directive in the surrogate control response header of an assembled page. This overrides any static cacheability rule defined in OracleAS Web Cache, and ensures that the assembled page is not cached in the Web Cache.
		If set to `true`, PPE does not include the `no-store` directive in the surrogate control response header of an assembled page. This allows the use of static cacheability rules for caching the assembled page in OracleAS Web Cache.
		Note: It is recommended to use the default value, `false`, as setting it to `true` makes cached content accessible using the URL only and this affects security. Portal data that is cached in OracleAS Web Cache is secured by the presence of secure OracleAS Portal HTTP headers in the request. A setting of `true` means that fully assembled pages may be requested by URL alone and will be returned from the cache.
			`false`

`dmsLogging`	`<init-param>`
	`<param-name>dmsLogging</param-name>`
	`<param-value>false</param-value>`
	`</init-param>`
		If you set `dmsLogging` to `true`, the PPE outputs data for DMS Logging.	`true`

Table D–1 (Cont.) Parallel Page Engine (PPE) Parameters

PPE Setting	Syntax	Description	Default Value

`cacheEncryptionKey`	`<init-param>`
	`<param-name>cacheEncryptionKey</param-name>`
	`<param-value>KEY</param-value>`
	`</init-param>`
		This key is used to obscure the headers used for caching using OracleAS Web Cache. This allows for a more secure cache key, and makes retrieving a cached object more difficult for unwanted requests.
		This key is not used if you are running 10*g* Release 2 (10.1.4) of the portal repository, and is provided only for backward compatibility when you use a 10.1.4 middle tier with an earlier release of the portal repository.
			Server Context information

E

Using Oracle Application Server Configuration Files

This appendix provides information about the configuration files and tables that can affect the connection to and the behavior of the Oracle Application Server and its components in the middle tier and on other computers to which it is connecting.

Specific topics covered include:

- Oracle HTTP Server Configuration File (httpd.conf)
- DAD Configuration File (dads.conf)
- Oracle Database Connection Configuration
- Web Cache Configuration Files
- OracleAS Single Sign-On Configuration Table
- OracleAS Single Sign-On's Partner Application Table
- Local HOSTS File
- Using Oracle Enterprise Manager 10g

E.1 Oracle HTTP Server Configuration File (httpd.conf)

The Oracle HTTP Server configuration file, `httpd.conf`, contains configuration information for running the Oracle HTTP Server. The content of this file includes information about listening ports, server names, virtual hosts, proxy configurations, and the like. This file also configures Secure Sockets Layer (SSL) support by defining information such as certificates and other HTTPS configuration directives. This file is available at the following location:

ORACLE_HOME`/Apache/Apache/conf/httpd.conf`

If you create additional virtual hosts in Oracle HTTP Server, then you must add the `RewriteEngine` and `RewriteOptions` mod_rewrite directives for the virtual host that is used by OracleAS Portal, in the `httpd.conf` file as shown in the following example (shown in bold text):

```
NameVirtualHost *:7778
<VirtualHost *:7778>
  ServerName www.xyz.com
  Port 7779
  ServerAdmin you@your.address
  RewriteEngine On
  RewriteOptions inherit
</VirtualHost>
```

E.2 DAD Configuration File (dads.conf)

This file contains the configuration parameters for the PL/SQL Database Access Descriptor (DAD). A DAD is a set of values that specifies how a database server should fulfill a HTTP request.

You can add a portal DAD or update a DAD by editing the `dads.conf` file. You can update the `dads.conf` file either by using Oracle Enterprise Manager 10*g* Application Server Control Console or by manually editing the file.

If you manually update the `dads.conf` file, then you must also add the necessary mod_rewrite and mod_oc4j directives in the `httpd.conf` and `mod_oc4j.conf` files respectively. You can do this in either of the following two ways:

- Using Application Server Control Console
- Manually Editing the dads.conf File

Using Application Server Control Console

Perform the following tasks using the Application Server Control Console:

1. Access the mod_plsql Configuration Pages. For details about accessing these pages, refer to the *Oracle Application Server mod_plsql User's Guide*.
2. Select the portal DAD and click **Edit**.
3. Click **Apply** without making any changes in the Edit mode.
4. Restart the Oracle HTTP Server and OC4J_Portal.

This ensures that the required mod_rewrite and mod_oc4j directives are added.

Manually Editing the dads.conf File

Based on the type of updates you make in the `dads.conf` file, perform all or some of the following tasks:

1. If you added a new portal DAD in the `dads.conf` file, then you must add the following Rewrite directives to the `httpd.conf` file:

   ```
   RewriteRule (^/pls/<dad>/.*) /portal$1 [PT]
   RewriteRule (^/pls/<dad>$) /portal$1 [PT]
   ```

   where <*dad*> is the name of the new DAD. For example:

   ```
   RewriteRule (^/pls/mydad/.*) /portal$1 [PT]
   RewriteRule (^/pls/mydad$) /portal$1 [PT]
   ```

2. If you modified a DAD name in the `dads.conf` file, then you must update the Rewrite directives described in the previous step with the new DAD name.
3. If you have manually created or updated any CGI environment variable in the `dads.conf` file, then you must update this variable in the `mod_oc4j.conf` file also. For example, an environment variable `TEST_APP` is available in the following format in the `dads.conf` file:

   ```
   PlsqlCGIEnvironmentList TEST_APP
   ```

   In the `mod_oc4j.conf` file, this variable is available in the following format:

   ```
   Oc4jEnvVar TEST_APP
   ```

   For details about updating the `mod_oc4j.conf` file, refer to the *Oracle HTTP Server Administrator's Guide*.

> **Note:** In the `mod_oc4j.conf` file, you must specify the environment variable only once, even if two DADs use the same environment variable.

4. To synchronize the manual configuration changes done on the middle tier, run the following commands:

```
MID_TIER_ORACLE_HOME/dcm/bin/dcmctl updateConfig -ct ohs
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=OC4J_Portal
```

E.3 Oracle Database Connection Configuration

SQL*Net configuration files define the entries that can be used as connect strings in the DADs. Typically, the `tnsnames.ora` and `sqlnet.ora` files in the location *ORACLE_HOME*`/network/admin` contain information on how Oracle Application Server can connect to the database where the OracleAS Portal installation is located. If you want to connect to the portal repository, you need to ensure that the TNS_ADMIN variable is properly configured. For example, in the C shell, enter the following at a command-line prompt:

```
setenv TNS_ADMIN path
```

Here, `path` points to the directory containing the `tnsnames.ora` file.

> **Note:** This command differs depending on the shell used.

If you want the Oracle Application Server installation to reference SQL*Net configuration files from another location, you must configure the TNS_ADMIN environment variable in `opmn.xml`, by performing the following steps:

1. Edit *ORACLE_HOME*`/opmn/conf/opmn.xml`.
2. Locate the tag for `ias-instance` in this file.
3. Locate the tag named `<environment>` within this tag.
4. Add a new variable `TNS_ADMIN` pointing to the path you want to use for SQL*Net name resolution. For example:

```
<ias-instance id="as1014.myinstance.abc.com">
      <environment>
         ...
         <variable id="TNS_ADMIN" value="/u01/app/oracle/network/admin/>
         ...
      </environment>
```

5. Restart the middle tier by issuing the following commands:

```
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl stopall
MID_TIER_ORACLE_HOME/opmn/bin/opmnctl startall
```

For more details on SQL*Net configuration, refer to the *Oracle Database Net Services Administrator's Guide* in the Oracle Database 10*g* documentation library. For more details on configuring Oracle Process Manager and Notification Server (OPMN), refer to the *Oracle Application Server Administrator's Guide*.

E.4 Web Cache Configuration Files

The following OracleAS Web Cache configuration files can be found in the *ORACLE_HOME*/webcache directory:

- webcache.xml
- internal.xml
- internal_admin.xml

See Also: *Oracle Application Server Web Cache Administrator's Guide*

E.5 OracleAS Single Sign-On Configuration Table

The WWSEC_ENABLER_CONFIG_INFO$ table is the configuration table for the Single Sign-On enabler stack. Typically, modifications to this table are handled by running the Portal Dependency Settings tool (ptlconfig), in the case of advanced configurations. This section is provided for additional information about the SSO configuration table. Modifications are not to be made directly, but instead by using the Portal Dependency Settings tool, ptlconfig. See Appendix A, "Using the Portal Dependency Settings Tool and File" for more information.

Each partner application to the OracleAS Single Sign-On has such a table for configuration information. One such table exists in the OracleAS Portal schema and the OracleAS Single Sign-On schema, because the OracleAS Single Sign-On application is also a partner application. This table defines the login URL for the OracleAS Single Sign-On that this partner Application is configured to use.

It is important to understand how the LSNR_TOKEN is used in the enabler configuration table, to help you plan what entries are required depending on your configuration.

This table may have more than one entry. There is one entry for each way the application's server is addressed. Understanding this requires a review of the authentication sequence. For the purpose of this discussion, the main flows include:

- Initial request to the requested URL
- Redirect to the OracleAS Single Sign-On for authentication
- Redirect to OracleAS Portal's success URL (wwsec_app_priv.process_signon)
- Redirect back to the requested URL

The OracleAS Single Sign-On (SSO) partner enabler APIs read the WWSEC_ENABLER_CONFIG_INFO$ table for configuration information. Similarly, in the OracleAS Single Sign-On, the OracleAS Single Sign-On's private APIs read the WWSSO_PAPP_CONFIGURATION_INFO$ table. In the latter table, the URL should be redirected to each partner application.

Because each partner application's success URL is stored in the OracleAS Single Sign-On's partner application configuration table, to support multiple host names for the partner application, each distinct host name requires its own partner application entry on the OracleAS Single Sign-On. This is so that each one can specify a success URL that has the same hostname as the partner application, so that the session cookie can be scoped appropriately. Furthermore, the domain to which cookies are scoped includes the server name (ServerName) and port, so server.domain.com:80 is treated as a different cookie domain from server.domain.com:8080.

Each entry in the enabler configuration table is then selected based on the host name and port that was used by the partner application.

For example, let's say that you wanted OracleAS Portal to be accessible from `http://www.xyz.com` and `http://www.abc.com`. In this case, two partner applications must be registered in the OracleAS Single Sign-On. One is defined for the `www.xyz.com` host and the other for the `www.abc.com` host. Each one specifies a success URL that is appropriate:

- `http://www.xyz.com/portal/pls/portal/portal.wwsec_app_priv.process_signon` for the `www.xyz.com` partner
- `http://www.abc.com/portal/pls/portal/portal.wwsec_app_priv.process_signon` for the `www.abc.com` application

Each of these partner application entries on the OracleAS Single Sign-On would have a distinct site ID, site token, and encryption key. OracleAS Portal's enabler configuration table has one row for each partner application, for example:

```
LSNR_TOKEN    SITE_ID   LS_LOGIN_URL …

www.xyz.com   1321      https://www.login.com/pls/…

www.abc.com   1322      https://www.login.com/pls/…
```

E.6 OracleAS Single Sign-On's Partner Application Table

The configuration table on the OracleAS Single Sign-On's side is the partner application Table, `WWSSO_PAPP_CONFIGURATION_INFO$`. Maintenance of this table is typically done using the OracleAS Single Sign-On application's user interface for adding or editing partner applications.

For an initial installation on a single database instance, running the Portal Dependency Settings Tool, `ptlconfig`, populates both the OracleAS Single Sign-On's partner configuration table and OracleAS Portal's enabler configuration table. For example:

```
ptlconfig -dad portal -sso
```

E.7 Local HOSTS File

The HOSTS file on a network host defines mappings of IP names to IP addresses. Normally, a Domain Name Server (DNS) provides the mapping of IP name to IP address. In some of the configurations described in Chapter 4, "Performing Basic Configuration and Administration", a host may need to be addressed in an internal network with a domain name that is not defined within the internal network. In these cases, the server's HOSTS file can provide the necessary name resolution.

E.8 Using Oracle Enterprise Manager 10*g*

You can use Oracle Enterprise Manager 10*g* Application Server Control Console for administering OracleAS Portal. Application Server Control Console is a Web-based tool that enables you to perform some of the management tasks described in this book. Refer to Chapter 7, "Monitoring and Administering OracleAS Portal" for more information about using Oracle Enterprise Manager 10*g*.

See Also: *Oracle Application Server Administrator's Guide*

F

Integrating JavaServer Pages with OracleAS Portal

OracleAS Portal gives you the ability to create various kinds of Web pages. You can supplement this ability with JavaServer Pages (JSPs).

This appendix describes how you can secure OracleAS Portal to allow access to only approved JSPs, and prevent unauthorized access by JSPs to portlet content. It also describes the steps required to allow access for protected external JSPs that require login.

The following topics are covered in this appendix:

- Using the JavaServer Page Configuration File
- Setting Up a JAZN File for External Communication

F.1 Using the JavaServer Page Configuration File

Because almost any JSP using the tag library can request OracleAS Portal portlet content, there is a need for a secure way to ensure that only approved JSPs obtain access. You can control this through two mechanisms:

- The `<portal:usePortal>` tag in the JSP
- An external JSP configuration file

The configuration file identifies the OracleAS Portal instances, and page groups within those instances, to which an external JSP is allowed access.

See Section F.1.1, "Contents of Your JavaServer Page Configuration File" for the specific coding requirements of the configuration file.

Your completed configuration file must then be identified to OracleAS Portal. See Section F.1.3, "Location of Your JavaServer Page Configuration File" for an explanation of the step.

This section contains the following sub-sections:

- Contents of Your JavaServer Page Configuration File
- Example JavaServer Page Configuration File
- Location of Your JavaServer Page Configuration File
- External JavaServer Page Login

F.1.1 Contents of Your JavaServer Page Configuration File

The required tags are:

- <jps>
- <portal>
- <database>
- <url>
- <cookie>
- <pageGroups>
- <pageGroup>

F.1.1.1 The <jps> Tag

The <jps> tag is a container tag that provides a list of OracleAS Portal instances to which external JSPs can have access.

Opening tag

```
<jps version="1.0">
```

Version must be set to 1.0 for the current OracleAS Portal release.

Closing tag

```
</jps>
```

F.1.1.2 The <portal> Tag

The <portal> tag describes an individual OracleAS Portal instance.

Opening tag

```
<portal name="MyPortal" default="true">
```

Closing tag

```
</portal>
```

Table F–1 The <portal> Tag's Attributes

Attribute	Value
name	Any descriptive name given to an OracleAS Portal instance. The name must be unique within the configuration file.
default	A true or false flag indicating whether this portal is the default instance that is used if a *usePortal* tag does not specify a portal name. If you provide no value, default is set to false.

Only **one** default portal is allowed for each configuration file.

F.1.1.3 The <database> Tag

The <database> tag provides database connection information about a given OracleAS Portal instance. For example:

```
<database data-source="jdbc/MyPortal"/>
```

The data-source attribute value is the name of the data source, which must be specified in the data-sources.xml file located in the *J2EE_HOME*/config directory.

Here is an example of a data-source definition:

```
<data-source
   class="com.evermind.sql.DriverManagerDataSource"
```

```
   name="MyPortal"
   location="jdbc/MyPortal"
   xa-location="jdbc/xa/MyPortal"
   ejb-location="jdbc/MyPortal"
   connection-driver="oracle.jdbc.driver.OracleDriver"
   username="portal_app"
   password="portal_app"
   url="jdbc:oracle:thin:@xyz.oracle.com:1521:orcl"
   inactivity-timeout="30"
/>
```

The `username` and `password` attributes must be set to the OracleAS Portal application schema user name and password.

F.1.1.4 The <url> Tag

The `<url>` tag provides connection information to the OracleAS Portal instance. For example:

```
<url protocol="http" host="defg.oracle.com" port="7500" path="portal/pls/portal"/>
```

Table F–2 The <url> Tag's Attributes

Attribute	Value
`protocol`	The name of the protocol used to connect to the OracleAS Portal instance. Currently, only HTTP and HTTPS protocols are supported. If you do not specify a protocol attribute, the default will be `http`.
`host`	The computer name for the OracleAS Portal middle tier.
`port`	Port number. If no port is specified, the default number will be `80`.
`path`	For this release, path must be set to `/portal/pls/<PORTAL-DAD-NAME>`.

F.1.1.5 The <cookie> Tag

The `<cookie>` tag describes the OracleAS Portal cookie. For example:

```
<cookie name="portal" maxAge="-1" path="/" domain=".oracle.com"/>
```

Table F–3 The <cookie> Tag's Attributes

Attribute	Value
`name`	The name of the cookie. This must be the same as the OracleAS Portal instance cookie name. *name* is a required attribute of the cookie tag.
`maxAge`	The maximum age of the cookie, specified in seconds. Specify a value of `-1` if you want the cookie to persist until browser shutdown. *maxAge* is a required attribute of the cookie tag.
`path`	The path on the server to which the browser returns this cookie. *path* is a required attribute of the cookie tag.
`domain`	This attribute should be specified only if changes were made to the SSO portlet cookie configuration. See the SSO documentation.

F.1.1.6 The <pageGroups> Tag

The `<pageGroups>` tag forms a container for the pageGroup tags. This tag has no attributes.

Opening tag

```
<pageGroups>
```

Closing tag

```
</pageGroups>
```

F.1.1.7 The <pageGroup> Tag

The `<pageGroup>` tag describes each individual page group's properties. For example:

```
<pageGroup name="JPSDemo" key="welcome" default="true"/>
```

Table F–4 The <pageGroup> Tag's Attributes

Attribute	Value
`name`	The page group name. This must be the name given to the page group when it was created in OracleAS Portal.
`key`	The page group's key. The value must match the Access Key value that was assigned to the page group in OracleAS Portal. (Note that a page group identified here must have JSP Access enabled.)
`default`	A flag set to `true` or `false` indicating whether or not this page group is the default page group within this OracleAS Portal instance. A default page group is the one used in the *usePortal* tag if no page group name is supplied. If no value provided for default in this pageGroup tag, it will be set to `false`.

Only **one** default page group is allowed for each portal instance.

F.1.2 Example JavaServer Page Configuration File

The following is an example of a JSP configuration file:

Example F–1 Example JavaServer Page Configuration File

```
<jps version="1.0">
   <portal name="MyPortal" default="true">
      <database data-source="jdbc/MyPortal"/>
      <url host="xyz.oracle.com" port="7500" path="/portal/pls/portal"/>
      <cookie name="portal" maxAge="-1" path="/" />
      <pageGroups>
         <pageGroup name="JPSDemo"  key="welcome" default="true"/>
         <pageGroup name="JPSDemo2" key="welcome" default="false"/>
      </pageGroups>
   </portal>
   <portal name="AnotherPortal">
      <database data-source="jdbc/AnotherPortal"/>
      <url protocol="http" host="abc.oracle.com" port="8888"
         path="/portal/pls/portal90"/>
      <cookie name="portal90" maxAge="-1" path="/" />
      <pageGroups>
         <pageGroup name="JPSDemo"  key="welcome"/>
```

```
        <pageGroup name="JPSDemo1"  key="welcome1"/>
        <pageGroup name="JPSDemo2"  key="welcome2"/>
        <pageGroup name="JPSDemo3"  key="welcome3"/>
        <pageGroup name="JPSDemo4"  key="welcome4"/>
      </pageGroups>
   </portal>
</jps>
```

F.1.3 Location of Your JavaServer Page Configuration File

By default, the name of the configuration file is assumed to be `wwjps.xml`, and the default location of the file is:

J2EE_HOME`/applications/portal/portal/WEB-INF`

However, your configuration file can have any other name, and can be located anywhere in the file system.

You specify the location using a context parameter in the `web.xml` file, which is located in the directory *J2EE_HOME*`/applications/portal/portal/WEB-INF`.

The context parameter in the `web.xml` file is:

```
<context-param>
   <param-name>oracle.webdb.service.ConfigLoader</param-name>
   <param-value>/WEB-INF/wwjps.xml</param-value>
      <description>This parameter specifies the location of the JPS
         configuration file</description>
</context-param>
```

F.1.4 External JavaServer Page Login

External JSPs can be categorized by their login requirements:

- Public JSPs, which do not require login (or to which users log in through the OracleAS Portal login link)
- Protected JSPs, which do require login

Protected external JSPs have additional setup requirements. These are explained in the next section.

F.2 Setting Up a JAZN File for External Communication

The following steps are required only for protected external JSPs. That is, external JSPs that require login.

In the external JSPs, if you need to log in to the portal, you need to use the following tag syntax:

```
<portal:usePortal id="AnyPortal" pagegroup="AnyPageGroup" login="true" />
```

When you execute this JSP, you will be redirected to OracleAS Single Sign-On if you are not already logged on. To make this work, look at the following sections:

- Setting Up mod_osso (if not already set up)
- Setting Up JAZN with LDAP

F.2.1 Setting Up mod_osso

By default, your Oracle HTTP Server is registered with OracleAS Single Sign-On. If that has been changed, and re-registration is necessary, refer to the *Oracle Application Server Single Sign-On Administrator's Guide*.

F.2.2 Setting Up JAZN with LDAP

JAZN is the internal name for a Java Authentication and Authorization Service (JAAS) provider. JAAS is a Java package that enables applications to authenticate and enforce access controls upon users. The use of JAZN in OracleAS Portal is limited to the authentication of external JSPs.

Confirm that the JAZN is working with the LDAP. (You can use the demo provided by the JAZN.)

Do the following additional step:

- Go to *J2EE_HOME*`/application-deployments/portal/orion-application.xml` and add the following:

```
<jazn provider="LDAP" location="ldap://<OIDHOST>:389" default-realm="oracle">
<jazn-web-app auth-method="SSO" />
</jazn>
```

Port number `389` is a default port for LDAP servers. However, any other port can be assigned. Contact your Oracle Internet Directory Administrator to obtain `<host>` and `<port>` information.

See Also: For more information:

- Section F.2, "Setting Up a JAZN File for External Communication"
- *Oracle Containers for J2EE Services Guide*

G

Using the wwv_context APIs

The `wwv_context` package contains procedures to create and maintain Oracle Text indexes used by OracleAS Portal. This appendix describes the content of this package in the following sections:

- Procedures
- Functions
- Constants
- Exceptions

> **Note:** See Chapter 8, "Configuring the Search Features in OracleAS Portal" for more information about Oracle Text indexes and how they are used in OracleAS Portal.

G.1 Procedures

The `wwv_context` package contains these procedures:

add_attribute_section

commit_sync

create_index

create_missing_indexes

create_prefs

createindex

drop_all_indexes

drop_index

drop_invalid_indexes

drop_prefs

dropindex

optimize

set_parallel_degree

set_sync_memory

set_use_doc_index

set_use_url_index

G.1.1 add_attribute_section

```
procedure add_attribute_section(
    p_attributeid     in number,
    p_attributesiteid in number
)
```

Adds a new section to the section groups used by the Item and Page indexes. The section group corresponds to an attribute. This changes the index metadata only, it does not update the index data itself. The new sections can be searched but the indexes themselves are not changed.

The indexes are changed only if they exist; if the indexes do not exist, this procedure has no effect.

Parameters:

`p_attributeId` - ID of the attribute section to add.

`p_attributeSiteId` - Site ID of the attribute section to add.

G.1.2 commit_sync

```
procedure commit_sync(
    p_index       in varchar2
    p_commit_sync in boolean
)
```

(Oracle Database 10*g* or later) Specifies whether an Oracle Text index is synchronized immediately after data is committed to your portal, or needs to be synchronized manually using `wwv_context.sync` (see sync).

If you choose to synchronize an index manually or your database is earlier than Oracle Database 10*g*, you can run wwv_context.sync directly or create a job that calls wwv_context.sync at regular intervals. wwv_context.sync ignores any index where commit_sync is set to true. See also, Section 8.3.5.4, "Scheduling Index Synchronization".

Note: The `commit_sync` property is available only in database versions Oracle Database 10*g* or later. In earlier versions, Oracle Text indexes can be synchronized manually, or according to a synchronization schedule (certain dates and times). See also Section 8.3.5.1, "Synchronizing Oracle Text Indexes".

Parameters:

`p_index` - The name of the index. One of the Index Name Constants.

`p_commit_sync` - Determines whether the index synchronizes immediately after a commit. Set `commit_sync` to `true` if you want the index to synchronize automatically whenever data is committed in your portal. Set `commit_sync` to `false` if you want to synchronize the index manually using `wwv_context.sync`.

Exception:

`INVALID_INDEX` – The name of the index was not recognized.

G.1.3 create_index

```
procedure create_index(
    p_index in varchar2
)
```

Creates a specific, named Oracle Text index. See also, Section 8.3.3, "Oracle Text Indexes".

Use this procedure for troubleshooting purposes only. Under normal circumstances, use create_missing_indexes to create *all* of the indexes that are missing, or createindex to drop invalid indexes and then re-create the preferences and missing indexes.

Parameters:

`p_index` - The name of the index you want to create. One of the Index Name Constants.

Exceptions:

`INVALID_INDEX` - The name of the index was not recognized.

G.1.4 create_missing_indexes

```
procedure create_missing_indexes(
    p_indexes out wwsbr_array
)
```

Creates all of the Oracle Text indexes that are missing. An index is considered to be present if it exists according to the view `ctx_user_indexes`.

This procedure does not check to see if the existing indexes are valid. Use the procedure drop_invalid_indexes to drop any indexes that are not entirely valid.

This procedure creates empty indexes. To populate the indexes you must mark them as 'requiring re-indexing' using the procedure touch_index(p_indexes wwsbr_array), and then you must synchronize the indexes.

This procedure does not create Oracle Text Datastore and Filter preferences; these preferences must already exist. Use the procedure create_prefs to create the preferences, if they do not exist.

Parameters:

p_indexes - Returns an array containing the list of indexes created.

G.1.5 create_prefs

```
procedure create_prefs
```

Creates the Datastore and Filter preferences which are both used when creating Oracle Text indexes. See also, Section 8.3.3.2, "Oracle Text Index Preferences".

This procedure does not create any of the Lexer preferences. Use the script `sbrimtlx.sql` located in the directory *`ORACLE_HOME`*`/portal/admin/plsql/wws` to create Lexer preferences. See also, Section 8.3.3.5, "Multilingual Functionality (Multilexer)".

Oracle Text preferences must exist before the indexes are created. Subsequent changes to these preferences do not take effect until the Oracle Text indexes are dropped and re-created.

G.1.6 createindex

```
procedure createindex(
    p_language in  varchar2 default wwnls_api.nls_default_language,
    p_message  out varchar2
)
```

Creates Oracle Text indexes used by OracleAS Portal. See Section 8.3.3, "Oracle Text Indexes" for more information.

This high level procedure performs the following tasks:

- Drops all existing preference objects.
- Drops any invalid indexes.
- Re-creates Oracle Text preferences.
- Creates indexes that are missing (initially empty).
- Marks all indexable OracleAS Portal content as requiring re-indexing, for all new indexes.
- Synchronizes indexes, that is, first populates and then optimizes the indexes.

This procedure is equivalent to:

```
wwv_context.drop_prefs;
wwv_context.drop_invalid_indexes;
wwv_context.create_prefs;
wwv_context.create_missing_indexes(l_indexes);
wwv_context.touch_index(l_indexes);
wwv_context.sync;
wwv_context.optimize;
```

G.1.7 drop_all_indexes

```
procedure drop_all_indexes
```

Drops *all* the Oracle Text indexes used by OracleAS Portal.

This procedure does not drop the Oracle Text preferences. Use the procedure drop_prefs to do this.

G.1.8 drop_index

```
procedure drop_index(
    p_index in varchar2
)
```

Drops a specific, named Oracle Text index. This procedure does not validate that the index exists.

Parameters:

`p_index` – The name of the index you want to drop. One of the Index Name Constants.

Exceptions:

`INVALID_INDEX` – The name of the index was not recognized.

G.1.9 drop_invalid_indexes

```
procedure drop_invalid_indexes
```

Drops invalid Oracle Text indexes only, that is, valid Oracle Text indexes are not dropped.

An index is considered to be valid, if the following status columns, in the following views, are all set to 'VALID':

- user_indexes.status
- user_indexes.domidx_status
- user_indexes.domidx_optstatus
- ctx_user_indexes.idx_status

If any status column is not valid or, if the index does not have an entry in both views, it is considered to be invalid and will be dropped. See Section 8.3.8, "Viewing the Status of Oracle Text Indexes" for more information.

G.1.10 drop_prefs

```
procedure drop_prefs
```

Drops the Oracle Text Datastore and Filter preferences. See also, Section 8.3.3.2, "Oracle Text Index Preferences".

Datastore and Filter preferences are used when creating the Oracle Text indexes. This procedure does not drop any of the Lexer preferences that are created using the script `sbrimtlx.sql`. The script is located in the directory *`ORACLE_HOME`*`/portal/admin/plsql/wws`.

G.1.11 dropindex

```
procedure dropindex(
    p_language in varchar2 default wwnls_api.nls_default_language,
    p_message  out varchar2
)
```

Drops all existing Oracle Text indexes used by OracleAS Portal. See also, Section 8.3.3, "Oracle Text Indexes".

This procedure is equivalent to:

```
wwv_context.drop_prefs;
wwv_context.drop_all_indexes;
```

G.1.12 optimize

```
procedure optimize(
    p_optlevel in varchar2 default ctx_ddl.optlevel_full,
    p_maxtime  in number   default null,
    p_token    in varchar2 default null
)
```

Optimizes all *existing* Oracle Text indexes used by OracleAS Portal. Each index is optimized by calling the Oracle Text procedure `ctx_ddl.optimize_index()`.

The parameters for this procedure are the same as those required by the Oracle Text procedure `ctx_ddl.optimize_index`.

Parameters:

`p_optlevel` - The optimization level, one of `FULL`, `FAST` or `TOKEN`.

`p_maxtime` - The time (in minutes) that Oracle Text spends optimizing the indexes.

`p_token` - Token to optimize (when doing TOKEN optimization).

You will find additional information in the *Oracle Text Reference* on the Oracle Technology Network (OTN), http://www.oracle.com/technology/products/text/index.html.

G.1.13 set_parallel_degree

```
procedure set_parallel_degree(
    p_index           in varchar2,
    p_parallel_degree in pls_integer
)
```

Sets the degree of parallelism used when an index is synchronized using the procedure `wwv_context.sync` (see sync). On a multi-processor computer you can run the synchronization operation in parallel. If you use multiple processors during synchronization it can speed up indexing, especially when you have large amounts of data to index.

The default setting is `1`, no parallelism. A number greater than 1 turns on parallel synchronization. If you specify a parallel degree that is higher than the total number of processors available in your database server, the degree of parallelism achieved during synchronization might be smaller than requested.

Note: This setting has no effect if the index synchronizes immediately after a commit (get_commit_sync returns `true`). See also, commit_sync.

You will find additional information in the *Oracle Text Reference* on OTN, http://www.oracle.com/technology/products/text/index.html.

Parameters:

`p_index` – The name of the index. One of the Index Name Constants.

`p_parallel_degree` – The degree of parallelism to use when the specified index is synchronized.

Exceptions:

`INVALID_SETTING` – The format or value of `p_parallel_degree` was not recognized.

`INVALID_INDEX` – The name of the index was not recognized.

`INTERNAL_EXCEPTION` – An unexpected internal error occurred.

G.1.14 set_sync_memory

```
procedure set_sync_memory(
    p_index  in varchar2,
    p_memory in varchar2
)
```

Specifies the amount of runtime memory that Oracle Text may use when synchronizing an index using the procedure `wwv_context.sync` (see sync). You can enter the memory value in bytes, or use the suffixes K, B or G to indicate that the value is in kilobytes, megabytes, or gigabytes respectively. For example, enter the value `10000` to specify 10000 bytes, or `10K` to specify 10 kilobytes.

When the memory specified becomes full, the data is written to the database. The more frequently this happens, the slower the indexing performance becomes and the Oracle Text indexes also become more fragmented. Fragmentation can slow down portal search queries. Specifying smaller amounts of memory will impact performance and increase index fragmentation, but might be useful when runtime memory is scarce.

If you set `p_memory` to `Null`, the default index memory setting is used. This default value is set using the configurable Oracle Text system parameter `DEFAULT_INDEX_MEMORY`. If you want to specify a different value, it must be less than the Oracle Text system parameter `MAX_INDEX_MEMORY`.

For more information, see *Oracle Text Reference* on OTN, http://www.oracle.com/technology/products/text/index.html.

Note: This setting has no effect if the index synchronizes immediately after a commit (get_commit_sync returns `true`). See also, commit_sync.

Parameters:

`p_index` - The name of the index. One of the Index Name Constants.

`p_memory` - The maximum amount of memory used to synchronize this index.

Exceptions:

`INVALID_SETTING` - The format or range of p_memory was not recognized.

`INVALID_INDEX` - The name of the index was not recognized.

`INTERNAL_EXCEPTION` - An unexpected internal error occurred.

G.1.15 set_use_doc_index

```
procedure set_use_doc_index(
p_value in boolean
)
```

Specifies whether the Document index is required. See also, Section 8.3.7, "Disabling Document and URL Indexing".

The value is cached for the duration of the session to avoid repeated requests for this information.

Parameters:

`p_value` - Either `true` or `false`. When set to `true`, the Document index is required.

G.1.16 set_use_url_index

```
procedure set_use_url_index(
p_value in boolean
)
```

Specifies whether the URL index is required. See Section 8.3.7, "Disabling Document and URL Indexing" for more information.

The value is cached for the duration of the session to avoid repeated requests for this information.

Parameters:

`p_value` - Either `true` or `false`. When set to `true`, the URL index is required.

G.1.17 sync

```
procedure sync
```

Synchronizes *all* Oracle Text indexes used by OracleAS Portal. Each index is synchronized by calling the Oracle Text procedure `ctx_ddl.sync_index()`. This procedure re indexes any rows that have been updated since the last synchronization. After synchronization, newly added or updated content can be searched. See also, Section 8.3.5.1, "Synchronizing Oracle Text Indexes".

Before synchronization, the pending queue is updated for the table `wwsbr_url$`. This table contains values for all the URLs attributes stored in OracleAS Portal. Rows from this queue are removed when the URL value is equal to the value of the constant `wwv_context_util.g_noindex`. Rows are set to this value to indicate that the original URL was not indexable by Oracle Text, for example, URLs such as those beginning with `https://` or `javascript:`.

You will find additional information on `ctx_ddl.sync_index` in Oracle Text Reference documentation on OTN, `http://www.oracle.com/technology/products/text/index.html`.

G.1.18 touch_index(p_indexes wwsbr_array)

```
procedure touch_index(
    p_indexes in wwsbr_array
)
```

Touches content for one or more indexes. When an index is touched, *all* the index content is marked as requiring synchronization. See Section 8.3.5.6, "Synchronizing All the Index Content" for more information.

Once index content is marked in this way, use the procedure sync to re index the marked content.

Note that this procedure operates across multiple virtual private portal subscribers, it is not confined to the current subscriber. The procedure switches to each subscriber in turn and returns back to the original subscriber when complete.

Parameters:

`p_indexes` - An array containing index names to touch. One or more of the Index Name Constants.

G.1.19 touch_index

```
procedure touch_index(
    p_index in varchar2 default null
)
```

Touches content for a single index or for all indexes. This procedure is a convenient way to touch a single, named index. Alternatively, you can use the procedure to touch all indexes, by passing the value *null*. See also, Section 8.3.5.6, "Synchronizing All the Index Content".

This procedure calls touch_index(p_indexes wwsbr_array) mentioned earlier.

Parameters:

`p_index` - The name of the index to touch, or `null` to touch all indexes. When specifying a name, use of one of the Index Name Constants.

Refer to Section G.1.18, "touch_index(p_indexes wwsbr_array)" for more information.

G.1.20 update_index_prefs

```
procedure update_index_prefs
```

Updates the current Oracle Text index datastore preferences. This procedure is valid only for database versions *earlier* than Oracle Database 10*g*.

When datastore preferences are modified after the indexes are created, the changes are not applied to the indexes automatically. Use this procedure to apply datastore preference changes to the Oracle Text indexes.

No action is taken for any indexes that are missing.

G.2 Functions

The `wwv_context` package contains these functions:

checkindex

doc_index

get_commit_sync

get_parallel_degree

get_sync_memory

get_use_doc_index

get_use_url_index

valid_doc_index

valid_url_index

url_index

G.2.1 checkindex

```
function checkindex(
    p_force in boolean default false
) return boolean
```

Checks whether all *required* Oracle Text indexes exist. The Document and URL indexes are optional, so the presence and validity of these indexes are determined by calls to valid_doc_index and valid_url_index. See also, Section 8.3.7, "Disabling Document and URL Indexing".

The value returned by `checkindex` is cached for the duration of the session. Whenever `true` is passed to `p_force`, the status of Oracle Text indexes is re-evaluated, regardless of any previously cached value.

Parameters:

`p_force` - Either `true` or `false`. When set to `true`, Oracle Text indexes are checked.

Returns:

Returns `true` if all required indexes exist and are valid.

G.2.2 doc_index

```
function doc_index
return boolean
```

Checks whether the Document index is required (using get_use_doc_index) and usable (using valid_doc_index).

Returns:

Returns `true` if the Document index is both required and valid.

G.2.3 get_commit_sync

```
function get_commit_sync(
  p_index in varchar2)
return boolean
```

Determines whether an index synchronizes immediately after data commits to your portal, or if it must be synchronized manually. See also, commit_sync.

> **Note:** The `commit_sync` property is not available for database versions *earlier* than Oracle Database 10*g*. This function returns `false` when called on an earlier database version.

Parameters:

`p_index` - The name of the index. One of the Index Name Constants.

Returns:

Returns `true` if the index is configured to synchronize immediately after data is committed to your portal. Returns `false` if the index is configured to synchronize manually.

G.2.4 get_parallel_degree

```
function get_parallel_degree(
    p_index in varchar2)
return boolean
```

Gets the degree of parallelism used when an index is synchronized using the procedure `wwv_context.sync` (see sync). On a multi-processor computer you can run the synchronization operation in parallel. If you use multiple processors during synchronization it can speed up indexing, especially when you have large amounts of data to index.

The default setting is `1`, no parallelism. A number greater than 1 turns on parallel synchronization. If the parallel degree is higher than the total number of processors available in your database server, the degree of parallelism achieved during synchronization might be smaller than that requested.

> **Note:** The parallelism setting has no effect if the index synchronizes immediately after a commit (get_commit_sync returns `true`).

You will find additional information in the *Oracle Text Reference* on OTN, http://www.oracle.com/technology/products/text/index.html.

Parameters:

`p_index` - The name of the index. One of the Index Name Constants.

Returns:

Returns the degree of parallelism used when synchronizing the specified index.

Exceptions:

`INVALID_INDEX` – The name of the index was not recognized.

`INTERNAL_EXCEPTION` – An unexpected internal error occurred.

G.2.5 get_sync_memory

```
function get_sync_memory(
    p_index in varchar2)
return boolean
```

Gets the amount of runtime memory (in bytes) that Oracle Text may use when synchronizing an index using the procedure `wwv_context.sync` (see sync).

When this memory becomes full, the data is written to the database. The more frequently this happens, the slower the indexing performance becomes and the Oracle Text indexes also become more fragmented. Fragmentation can slow down portal search queries. Small amounts of memory will impact performance and increase index fragmentation, but might be useful when runtime memory is scarce.

A `Null` value indicates that the default index memory setting is used. This default value is set using the configurable Oracle Text system parameter `DEFAULT_INDEX_MEMORY`. For more information, see *Oracle Text Reference* on OTN, http://www.oracle.com/technology/products/text/index.html.

> **Note:** The memory setting has no effect if the index synchronizes immediately after a commit (get_commit_sync returns `true`).

Parameters:

`p_index` - The name of the index. One of the Index Name Constants.

Returns:

Returns the amount of memory (in bytes) used when synchronizing the specified index.

Exceptions:

`INVALID_INDEX` – The name of the index was not recognized.

`INTERNAL_EXCEPTION` – An unexpected internal error occurred.

G.2.6 get_use_doc_index

```
function get_use_doc_index
return boolean
```

Determines whether the Document index is required. See also, Section 8.3.7, "Disabling Document and URL Indexing".

The value is cached for the duration of the session to avoid repeated requests for this information.

Returns:

Returns `true` if the Document index is required.

G.2.7 get_use_url_index

```
function get_use_url_index
return boolean
```

Determines whether the URL index is required. See also, Section 8.3.7, "Disabling Document and URL Indexing".

The value is cached for the duration of the session to avoid repeated requests for this information.

Returns:

Returns `true` if the URL index is required.

G.2.8 valid_doc_index

```
function valid_doc_index
return boolean
```

Determines whether the Document index is in a valid, usable state. See also, Section 8.3.7, "Disabling Document and URL Indexing". The function checkindex is called, if it has not yet been called.

The value is cached for the duration of the session to avoid repeated requests for this information.

Returns:

Returns `true` if the Document index exists and is valid.

G.2.9 valid_url_index

```
function valid_url_index
return boolean
```

Determines whether the URL index is in a valid, usable state. See also, Section 8.3.7, "Disabling Document and URL Indexing". The function checkindex is called, if it has not yet been called.

The value is cached for the duration of the session to avoid repeated requests for this information.

Returns:

Returns `true` if the URL index exists and is valid.

G.2.10 url_index

```
function url_index
return boolean
```

Checks whether the URL index is required (using get_use_url_index) and usable (using valid_url_index).

Returns:

Returns `true` if the URL index is both required and valid.

G.3 Constants

The `wwv_context` package contains these constants:

- Index Name Constants
- Oracle Text AUTO_FILTER Format Constants
- Oracle Text Job Constants
- URL Unsuitable for Indexing Constant

G.3.1 Index Name Constants

Use the following constants to identify the Oracle Text indexes used by OracleAS Portal:

- **Page index -** wwv_context.PAGE_TEXT_INDEX
- **Document index -** wwv_context.DOC_TEXT_INDEX
- **Perspective index -** wwv_context.PERSPECTIVE_TEXT_INDEX
- **Item index -** wwv_context.ITEM_TEXT_INDEX
- **Category index -** wwv_context.CATEGORY_TEXT_INDEX
- **URL index -** wwv_context.URL_TEXT_INDEX

PAGE_TEXT_INDEX

```
PAGE_TEXT_INDEX constant varchar2(30) := 'WWSBR_CORNER_CTX_INDX'
```

DOC_TEXT_INDEX

```
DOC_TEXT_INDEX constant varchar2(30) := 'WWSBR_DOC_CTX_INDX'
```

PERSPECTIVE_TEXT_INDEX

```
PERSPECTIVE_TEXT_INDEX constant varchar2(30) := 'WWSBR_PERSP_CTX_INDX'
```

ITEM_TEXT_INDEX

```
ITEM_TEXT_INDEX constant varchar2(30) := 'WWSBR_THING_CTX_INDX'
```

CATEGORY_TEXT_INDEX

```
CATEGORY_TEXT_INDEX constant varchar2(30) := 'WWSBR_TOPIC_CTX_INDX'
```

URL_TEXT_INDEX

```
URL_TEXT_INDEX constant varchar2(30) := 'WWSBR_URL_CTX_INDX'
```

G.3.2 Oracle Text AUTO_FILTER Format Constants

The Document and URL indexes uses AUTO_FILTER to convert documents into a plain text format suitable for indexing. Not all document types need to be filtered; some document types can be indexed directly. AUTO_FILTER uses the following settings to determine which documents require filtering:

- **BINARY_FORMAT** - indicates that a document is a format, other than plain text or HTML, but supported by AUTO_FILTER, such as PDF. Such documents are converted into an indexable text format (providing the binary format is supported by AUTO_FILTER).
- **TEXT_FORMAT** - indicates that a document is either plain text or HTML. When specified, the document is not filtered, but might be character set converted.
- **IGNORE** - indicates that a document need not be indexed at all; for example, image files.

> **Note:** AUTO_FILTER replaces the INSO_FILTER, which is now deprecated.

BINARY_FORMAT

```
BINARY_FORMAT constant varchar2(10) := 'BINARY';
```

TEXT_FORMAT

```
TEXT_FORMAT   constant varchar2(10) := 'TEXT';
```

IGNORE

```
IGNORE        constant varchar2(10) := 'IGNORE';
```

G.3.3 Oracle Text Job Constants

Use these constants for managing Oracle Text maintenance jobs:

- SYNC_JOB_PREF - the preference name for storing the synchronization job ID. Used by the index synchronization script `textjsub.sql`. See also, Section 8.3.5.4, "Scheduling Index Synchronization".
- OPTIMIZE_JOB_PREF - the preference name for storing the optimization job ID. Used by the index optimization script `optjsub.sql`. See also, Section 8.3.5.8, "Scheduling Index Optimization".

SYNC_JOB_PREF

```
SYNC_JOB_PREF constant varchar2(20) := 'text_sync_jobid';
```

OPTIMIZE_JOB_PREF

```
OPTIMIZE_JOB_PREF constant varchar2(20) := 'text_optimize_jobid';
```

G.3.4 URL Unsuitable for Indexing Constant

The absolute URL value used to indicate that a row should not be indexed. The URL index is created on the `wwsbr_url.absolute_url` column and this column is populated by a trigger.

If a URL is not suitable for indexing, such as URLs beginning with `javascript:`, this constant value is used. See also, Section 8.3.6.2, "Unsupported URLs".

G_NOINDEX

```
G_NOINDEX constant varchar2(15) := 'wwsbr_noindex'
```

G.4 Exceptions

INVALID_INDEX

The name of the index was not recognized.

```
INVALID_INDEX exception
```

INVALID_SETTING

An invalid value was specified for an API setting.

```
INVALID_SETTING exception
```

H

Using TEXTTEST to Check Oracle Text Installation

OracleAS Portal uses the Oracle Text functionality to extend its search capabilities. If you want to check that Oracle Text functionality is working correctly, you can use the utility `TEXTTEST`. This utility is located at *`MID_TIER_ORACLE_HOME`*`/portal/admin/texttest/texttest`.

This appendix contains the following sections:

> **Note:** This utility only checks Oracle Text functionality that is specifically required by OracleAS Portal.

H.1 When to Use TEXTTEST

Oracle Text functionality is now enabled in OracleAS Portal by default and therefore all new OracleAS Portal installations expect Oracle Text to be present and functioning correctly. The `TEXTTEST` utility is useful if you want to:

- Check that Oracle Text functionality is working correctly before installing an Oracle Text enabled portal.
- Determine whether a problem with Oracle Text searching functionality within OracleAS Portal is due to an Oracle Text installation issues.

If you choose to disable Oracle Text searching functionality in OracleAS Portal, you do not need to run this utility.

H.2 Before Running TEXTTEST

1. You need to run the `TEXTTEST` utility from an Oracle Application Server Oracle home and it requires access to:
   - A working Perl installation (`TEXTTEST` has been tested with Perl 5.6.1).

- Perl DBI and DBD::Oracle modules. The DBD::Oracle modules themselves require the Oracle Database client libraries.

  To ensure access, set the path `PATH $ORACLE_HOME/perl/bin:$PATH` and set the Perl library path `setenv PATH $ORACLE_HOME/perl/lib/5.6.1:$PATH`.

All of these are found in an Oracle Application Server Oracle home.

2. Ensure the correct Oracle home is selected.

   - For UNIX platforms, ensure the ORACLE_HOME environment variable is set and that the library path used by `ld` includes *`ORACLE_HOME`*`/ctx/lib`. The library path environment variable for the different UNIX platforms are as follows:

     Solaris, Tru64 UNIX, Linux: $LD_LIBRARY_PATH

     HP/UX: $SHLIB_PATH and $LD_LIBRARY_PATH

     IBM AIX: $LIBPATH

   - On Windows, use the Oracle home selector to choose the correct Oracle home.

   This is necessary so that the Perl DBD::Oracle module can find the correct Oracle client libraries. `TEXTTEST` also makes reference to the Oracle home environment variable. The Oracle home selected must be the Oracle Application Server Oracle home from where you intend to run the `TEXTTEST` utility.

3. Ensure that Perl can resolve the Perl module Portal::Text::Test.

   This module resides at:

   `ORACLE_HOME``/perl/lib/site_perl/5.6.1/Portal/Text/Test.pm`

   If you are using the Perl installation from the Oracle Application Server Oracle home, this is automatically included on the `@INC` path and no action is necessary. However, if you are using another Perl installation to run the utility, you may need to take steps to ensure that this location is included in the @INC path before running `TEXTTEST`. One way to do this, is to set the `PERL5LIB` environment variable to include:

   `ORACLE_HOME``/perl/lib/site_perl/5.6.1`
   `ORACLE_HOME``/perl/lib/5.6.1:$PERL5LIB`

4. If necessary, configure some of the tests that `TEXTTEST` will run.

   For example, if your Oracle Application Server installation is behind a firewall and you perform URL tests that access content on the Internet. See Section H.5, "Configuring TEXTTEST" for more information.

H.3 Running TEXTTEST

The `TEXTTEST` utility is located at *`MID_TIER_ORACLE_HOME`*`/portal/admin/texttest/texttest`. The default document directory is *`ORACLE_HOME`*`/Portal/admin/texttest/doc`.

You can run the `TEXTTEST` utility from the command line or DOS prompt. If you run the utility with no arguments, usage information is displayed. The command line arguments are detailed subsequently:

```
ORACLE_HOME/perl/bin/perl texttest -c sys_connect_string [-v] [-k] [-d document_
directory] [-t textcase_schema] [-p proxy] [-n noproxy]
```

Table H–1 TEXTTEST Parameters

Parameter	Description
-c	Connect string for the schema to connect with DBA privileges to create the test schema. For example, sys/change_on_install@orcl as sysdba
-v	Show verbose output.
-k	Keep test schema after tests.
-d	Document directory containing documents to upload. The document indexing tests use these uploaded documents. If not specified, TEXTTEST looks for a directory called 'doc' in the same location as this script.
-t	Name of the test schema. This is the schema that is created and in which the tests are run. Default is TEXTCASE. The password will be the same as the schema name. If it already exists, the existing schema will be used. However without the -k option it will still be dropped at the end of the test, so be careful.
-p	Proxy to use for the URL indexing tests, For example, global.uk.mycompany.com:80. The port is optional. The same proxy is used for both HTTP and FTP URLs.
-n	No proxy domains, comma separated list of up to 16 domains that the proxy will not be used for. For example, uk.mycompany.com,us.mycompany.com
-u	URL indexing test data file location.

The only mandatory argument is -c (the database connection information) and this must be a SQL*Plus style connect string. The schema specified is the one that is used to connect to the database. A separate schema will be created for running the tests.

The schema specified in the -c argument is not the schema used to run the tests. This schema needs DBA privileges. If you need to connect with a particular role, such as SYSDBA, when connecting to the SYS schema, specify this in the normal SQL*Plus format.

Note that if the -c argument contains spaces, you must add quotes. For example,

```
texttest -c 'sys/change_on_install@orcl as sysdba'
```

The -t argument specifies the name of the schema in which the tests are run. The default schema name is TEXTCASE. This schema is created in the early stages of the tests and is normally dropped at the end of the tests. You must ensure that this schema does not already exist in the database. If the test schema already exists, it is used but is dropped at the end of the testing.

H.4 Understanding TEXTTEST Results

By default, the output of the TEXTTEST is a simple statement of whether each test passed or failed, that is, OK or Not OK. For more detailed information about the tests and what causes them to fail, run TEXTTEST in verbose mode, that is, specifying the -v command line flag. The information displayed when the verbose mode is enabled is shown in more detail later.

See Section H.6, "Descriptions of TEXTTEST Tests" for details about why a test fails. Remember that if some of the tests fail, it may cause other tests to fail later on. For example, if the first connect to the database fails, then all subsequent tests also fail. Therefore, it is recommended that you investigate failures in the order they occur.

H.5 Configuring TEXTTEST

Use the file *ORACLE_HOME*/perl/lib/site_perl/5.6.1/Portal/Text/Config.pm to customize the default behavior of TEXTTEST. This file contains a Perl hash definition which itself contains definitions for various default values.

In most cases these values can be overridden by specifying command line arguments. Refer to Section H.3, "Running TEXTTEST" for details. If there is a default value defined in Config.pm and no command line value is specified, then the value from Config.pm is used.

Edit Config.pm if you want to change the default values permanently. This may be useful, for example, you always want to have proxy settings defined and you do not want to specify them every time on the command line. However, it is possible to successfully run TEXTTEST without modifying this configuration file.

H.5.1 Configuring Document Tests

OracleAS Portal uses Oracle Text functionality to search document content that is uploaded into the portal. When content is uploaded it is stored within OracleAS Portal database tables. Before it can be searched, the content must be indexed. During the indexing process Oracle Text processes each of the uploaded documents in turn. If the document is in a binary format (for example, a Word Document, or a Powerpoint document) it must be filtered and converted to plain text before it is indexed.

To test this functionality TEXTTEST creates a document table, uploads a number of files and attempts to filter them. The files that are uploaded are taken from a document directory. The default location is configured in Config.pm as *ORACLE_HOME*/Portal/admin/texttest/doc.

Oracle Text cannot filter all documents. Therefore, some documents that are expected to fail the indexing test can be placed into the document directory, with a specific error reported. Because the error is expected, the test should still pass when the error occurs.

To test this behavior, you can configure a list of expected exceptions in an exceptions file. This file lists the file name and the expected error. You must enter one file name in each line followed by the expected error, separated by a space. If the file name contains a space, it should be escaped using \ as an escape character.

The error is treated as a Perl regular expression so it does not need to contain the whole error message. At the simplest level, you can specify part of the error string and this will match. This enables you to specify just the error code, for example. More complicated Perl regular expressions are also permissible. Refer to perldoc (on perlre page) for more information on Perl regular expressions. If the expected error is simply *, any exception is expected, and no failure whilst indexing this document will cause a test failure.

For example, the file might contain these four lines:

```
searchnotes.zip DRG-11207: user filter command exited with status 1
# The following PDF has security and cannot be filtered
my\ secured\ pdf.pdf DRG-11207
search.jar *
```

The first line includes the entire error. The second line is a comment that is ignored. The third line treats any DRG-11207 error as a expected. The fourth line can fail with any error and the test still passes.

By default, the document indexing exceptions file is called `index_exceptions` and it is located in the document indexing directory (configured in `Config.pm`). If the location is specified as a relative path, it is relative to the document directory.

Note that due to limitations in the Perl DBD::Oracle module it's not possible to stream the documents from the file system to the database. Instead the entire document is loaded into memory before being uploaded to the database. This means that it is necessary to have enough memory to contain the entire document. Only enough space for each document at a time is required.

H.5.2 Configuring URL Tests

OracleAS Portal uses Oracle Text functionality to fetch URLs that are listed as URL attributes, either on URL items, other items or pages. Once the fetched content is indexed and becomes searchable.

`TEXTTEST` tests this functionality by creating a similar URL index. The test data for URL testing consists of a list of URLs. `TEXTTEST` loads the URLs from a URL data file. Each line in the data file contains a URL to attempt to index. It may also optionally contain an error message. If that error is found while indexing the corresponding URL, it is accepted as an expected error and does not cause the test to fail.

The expected error message is taken as a Perl regular expression that is matched against the error obtained from indexing. You must separate the expected error from the URL by a space character. If the expected error is specified as * then any error is treated as expected and does not cause the test to fail. For example:

```
http://www.oracle.com
http://www.google.com DRG-11614: URL store: communication with host specified in
http://www.google.com timed out
http://www.imaginaryurl.com DRG-11612: URL store: unknown host specified in
http://www.imaginaryurl.com
http://www.anotherimaginaryurl.com DRG-11612
http://www.expectederror.com *
```

The first URL is expected to be found. An error is reported if it cannot be indexed.

`http://www.google.com` is expected to timeout (perhaps because the portal is behind a firewall and no proxies are specified). If this failure occurs the test will still pass.

`http://www.imaginaryurl.com` is expected to fail with an unknown host error.

`http://www.anotherimaginaryurl.com` is also expected to fail with an unknown host error. Note that it's not necessary to specify the whole error string. Because it's treated as a regular expression just the error code will match. If it fails with this error the test will still pass.

`http://www.expectederror.com` will never cause the test to fail. We have said that regardless of any errors that occur, we should still pass the test.

Section 8.3.10, "Viewing Indexing Errors" describes some of the most common Oracle Text URL error messages.

Expected and unexpected errors are reported when `TEXTTEST` is run in verbose mode (`-v` command line flag). When `TEXTTEST` is run it opens the URL data file and uses it to populate the URL test table. This enables you to amend and augment the list of URLs used for testing by changing the contents of the file.

The default location for the URL data file is specified in the file `Config.pm`. Alternatively you can specify a URL test data file using the `-u` command line argument when running `TEXTTEST`. For example:

```
texttest -c 'sys/change_on_install@orcl as sysdba' -u ORACLE_
HOME/Portal/texttest/url
```

ORACLE_HOME`/Portal/texttest/url` is the default location for the URL data file, within the Oracle Application Server Oracle home.

You may change the URL details if you think a specific URL is causing problems in your portal installation. Or perhaps, your Oracle Application Server installation resides behind a firewall and you wish to change the URL test data to include URLs that are local to your intranet, rather than public URLs on the Internet.

H.5.3 URL Tests and Proxies

If your portal installation resides behind a firewall it may be necessary to configure Oracle Text to use a proxy before it can fetch URLs that reside beyond the firewall.

If you run `TEXTTEST` in these circumstances without setting proxies, the URL indexing tests fail. In this case you have three choices:

- Remove the failing URLs from the test data set. Simply remove the line from the URL data file.
- Mark the offending tests as expected to fail. Do this by placing the URL followed by the expected error message in the URL data file.
- Specify a proxy to use. See Section H.5.4, "Specifying Proxies for Use with URL Indexing Tests" for details.

H.5.4 Specifying Proxies for Use with URL Indexing Tests

You can specify a proxy to use in two locations:

- In the file `Config.pm` that contains separate settings for *ftp_proxy* and *http_proxy*.
- Using the `-p` parameter for the `TEXTTEST` script. In this case, the same proxy is used for both HTTP and FTP proxies.

In both cases the form of the proxy should be `<hostname>.<domain>:<port>`. The port is optional. For example,

```
www-proxy.us.abc.com:80
emeacache.abc.com
```

The `-n` command line argument and the no_proxy `Config.pm` setting can both be used to specify a list of domains for which the proxy should not be used. The list should be comma separated. For example,

```
uk.abc.com,us.abc.com,abc.com
```

H.6 Descriptions of TEXTTEST Tests

This section describes each of the tests that `TEXTTEST` performs and outlines some of the common causes for failure of each test.

H.6.1 Connect to Database as User sys

Description:

Connects to the database as the privileged user used to create the test schema. This is referred to the *sys* user or the *sys* schema. However, it does not have to be the user *sys*, any sufficiently privileged user will suffice.

Possible cause of failure:

- Incorrect schema name or password.
- If the user, such as *sys*, needs to connect with a specific role then the roles must be specified in the in the connect string in the usual format, that is, `sys/change_on_install as sysdba`.

When this test fails, it causes other tests to fail.

H.6.2 Create textcase Schema

Description:

Creates the schema into which test objects are installed. By default, this schema is called `textcase` and it is referred to as the *test* schema.

Possible cause of failure:

- The user with which `TEXTTEST` is connected, does not have privileges to create other users.
- There are several other reasons why it might not be possible to create a new schema, for example, there may be insufficient space in the database.

When this test fails, it causes other tests to fail.

H.6.3 Grant DBA Role to textcase Schema

Description:

Grants the DBA role to the `test` schema. This allows it to directly create and remove objects from the `ctxsys` schema.

Possible cause of failure:

- The user with which `TEXTTEST` is connected, does not have the necessary privileges to grant the DBA role to another user. It must have the DBA role itself to do this.

H.6.4 Grant CTXAPP Role to textcase Schema

Description:

Grants the CTXAPP role to the *test* schema. This is required when using Oracle Text features.

Possible cause of failure:

- The user with which `TEXTTEST` is connected, does not have the necessary privileges to grant CTXAPP to another user. It must have the DBA role itself to do this.

- The CTXAPP role is missing. This indicates an incomplete, corrupt or missing Oracle Text installation.

H.6.5 Disconnect From sys

Description:

`TEXTTEST` disconnects from the *sys* schema to reconnect to the *test* schema.

Possible cause of failure:

- No obvious cause of failure.

H.6.6 Connect to textcase Schema

Description:

`TEXTTEST` reconnects to the *test* schema to begin creating schema objects and running Oracle Text tests.

Possible cause of failure:

- No obvious cause of failure.

H.6.7 Create textcase Item Related Tables

Description:

Creates the tables used for testing item indexing with a user datastore.

Possible causes of failure:

- No obvious cause of failure.
- General database problems such as insufficient free tablespace to complete the operation.

H.6.8 Populate Item Tables

Description:

Populates the tables used for item indexing tests. They are populated using data held within the `TEXTTEST` script itself.

Possible cause of failure:

- No obvious cause of failure.

H.6.9 Create Document Table

Description:

Creates the table used for document filtering and indexing tests.

Possible cause of failure:

- No obvious cause of failure.

H.6.10 Populate Document Table

Description:

Populates the document table from a specified document directory.

Possible cause of failure:

- The specified document directory cannot be found or is not readable. The files within the document directory must be readable.
- Insufficient memory on the computer where `TEXTTEST` is running to hold any one of the documents in memory.

H.6.11 Create URL Table

Description:

Creates the table used for URL indexing tests.

Possible cause of failure:

- No obvious cause of failure.

H.6.12 Populate URL Table

Description:

Populates the tables used for URL indexing tests. They are populated from the URL data file. See Section H.6.11, "Create URL Table" for details.

Possible cause of failure:

- The URL indexing data file cannot be found, or is not readable.
- Data within the URL data file is in an incorrect format.

H.6.13 Create Oracle Text Datastore Procedure

Description:

Creates a datastore procedure in the `ctxsys` schema. The *test* user has DBA privileges and this procedure is created or replaced, so if the `ctxsys` schema is installed, there should not be a problem.

Possible cause of failure:

- The `ctxsys` schema is not present, which also implies that Oracle Text is not installed in the database.

H.6.14 Create Oracle Text Preferences

Description:

Creates the Oracle Text preferences (not including the Lexer preferences). Any existing preferences are dropped to avoid clashes.

Possible cause of failure:

- Problems with the Oracle Text installation.

- Problems with the compatibility of the preferences that TEXTTEST is attempting to create with this Oracle Text version, that is, preference version is not as expected.

H.6.15 Create Lexer Preferences

Description:

Creates the Oracle Text lexer preferences. Any existing preferences are dropped to avoid clashes.

Possible cause of failure:

- Problems with the Oracle Text installation.
- Problems with the compatibility of the preferences that TEXTTEST is attempting to create with this Oracle Text version, that is, preference version is not as expected.

H.6.16 Create Section Group and Zone Sections

Description:

Creates the section groups and zone sections for the item indexing tests.

Possible cause of failure:

- No obvious cause of failure.
- Possibly a problem with the Oracle Text installation, or one of the previous test having failed.

H.6.17 Create Oracle Text Item Index

Description:

Creates the Oracle Text index for testing item indexing with a user datastore. This test does **not** populate the index.

Possible cause of failure:

- No obvious cause of failure.
- Possibly a problem with the Oracle Text installation, or one of the previous test having failed.

H.6.18 Create Oracle Text Document Index

Description:

Creates the Oracle Text index for testing document indexing. This test does **not** populate the index.

Possible cause of failure:

- No obvious cause of failure.
- Possibly a problem with the Oracle Text installation, or one of the previous test having failed.

H.6.19 Create Oracle Text URL Index

Description:

Creates the Oracle Text index for testing URL indexing. This test does **not** populate the index.

Possible cause of failure:

- No obvious cause of failure.
- Possibly a problem with the Oracle Text installation, or one of the previous test having failed.

H.6.20 Touch All Item Content So That Pending

Description:

Updates all of the rows in the items test table so that they are placed in the Oracle Text pending queue.

Possible cause of failure:

- No obvious cause of failure.
- Possibly a problem with the Oracle Text installation, or one of the previous test having failed.

H.6.21 Touch All Document Content So That Pending

Description:

Updates all of the rows in the document test table so that they are placed in the Oracle Text pending queue.

Possible cause of failure:

- No obvious cause of failure.
- Possibly a problem with the Oracle Text installation, or one of the previous test having failed.

H.6.22 Touch All URL Content So That Pending

Description:

Updates all of the rows in the URL test table so that they are placed in the Oracle Text pending queue.

Possible cause of failure:

- No obvious cause of failure.
- Possibly a problem with the Oracle Text installation, or one of the previous test having failed.

H.6.23 Synchronize Item Index

Description:

Synchronizes the Oracle Text index on the item indexing test tables. This causes the content to be indexed.

Because the data set used for the item indexing is controlled and internal to the `TEXTTEST` script, this test is always expected to pass.

Possible cause of failure:

- A previous test has failed.
- Possibly a problem with the Oracle Text installation. Verify the Oracle Text installation and reinstall if necessary. Ensure that you complete *all* manual steps for any database upgrades, as these often contain Oracle Text related steps.

H.6.24 Synchronize Document Index

Description:

Synchronizes the Oracle Text index on the document indexing test table. This causes the content to be indexed.

Possible cause of failure:

- One of the documents uploaded for the test could not be filtered. This is not necessarily a problem as the document might not be in one of the formats that are filterable by Oracle Text.

  Consult the *Oracle Text Reference* (see chapter on supported formats). Either remove the document, or mark it as an expected failure (see Section H.5.1, "Configuring Document Tests" for details).

- An unexpected indexing failure, either caused by a bug in the filtering software or by incorrect configuration.

  Consult the *Oracle Text Reference* and Chapter 8, "Configuring the Search Features in OracleAS Portal" for more information. If the Oracle Text installation is configured correctly and the document format is a supported one but it still cannot be filtered, contact Oracle Support Services.

H.6.25 Synchronize URL Index

Description:

Synchronizes the Oracle Text index on the URL indexing test tables. This causes the content to be indexed.

Possible cause of failure:

- One of the URLs specified in the URL indexing test data may not be returning HTML or plain text that can be indexed by Oracle Text. This can happen for a number of reasons. The URL may be incorrect or the site might be unavailable.
- If the database instance is behind a firewall and the URL is beyond the firewall, then it might be necessary to configure the tests to use a proxy server. See Section H.5.2, "Configuring URL Tests" for more information. If the URL is expected to fail, you can marked it as such in the URL test data so that this test will pass.

H.6.26 Drop Datastore Procedure from ctxsys

Description:

Drops the datastore procedure created in the `ctxsys` schema.

This test is not carried out if the `-k` option is used to keep the *test* schema once the tests are completed. See Section H.3, "Running TEXTTEST" for more information.

Possible cause of failure:

- No obvious cause of failure.

H.6.27 Disconnect From textcase Schema

Description:

Disconnects from the *test* schema.

Possible cause of failure:

- No obvious cause of failure.

H.6.28 Connect As User sys

Description:

Reconnects to the `sys` schema to drop the test schema.

This test is not carried out if the `-k` option is used to keep the *test* schema once the tests are completed. See Section H.3, "Running TEXTTEST" for more information.

Possible cause of failure:

- No obvious cause of failure.

H.6.29 Drop textcase Schema

Description:

Drops the *test* schema.

This test is not carried out if the `-k` option is used to keep the *test* schema once the tests are completed. See Section H.3, "Running TEXTTEST" for more information.

Possible cause of failure:

- No obvious cause of failure.

H.6.30 Disconnect From Database

Description:

Disconnects from the `sys` schema.

Possible cause of failure:

- No obvious cause of failure.

Configuring the Portal Tools Providers

Portal Tools includes two Web providers, Web Clipping and OmniPortlet, that enable page designers and portlet developers to build portlets declaratively. With the Web Clipping portlet, you can publish content from remote Web sites as portlets on a portal page. With OmniPortlet, you can publish data from various data sources, such as Web Services, XML, or a database, and display the data in various layouts, such as a table, a chart, or HTML that they define.

This appendix covers the following topics:

- Configuring Web Clipping
- Configuring OmniPortlet

I.1 Configuring Web Clipping

Web Clipping is a browser-based declarative tool that enables you to integrate any Web application with OracleAS Portal. It is designed to give you quick integration by leveraging the Web application's existing user interface. Web Clipping has been implemented as a Web provider using the Java Portal Developers Kit, which is a component of OracleAS Portal.

With Web Clipping, you can collect Web content into portlets in a single centralized Web page. You can use Web Clipping to consolidate content from Web sites scattered throughout a large organization.

Before you use Web Clipping, you must perform some administrative tasks, including:

- Configuring the Web Clipping Repository
- Registering the Web Clipping Provider (PDK Only)
- Configuring HTTP or HTTPS Proxy Settings
- Configuring Caching
- Adding Certificates for Trusted Sites

  Section 6.1.9.1 in Chapter 6, "Securing OracleAS Portal" describes how to configure or extend the trusted certificate file. A trusted server certificate file, `ca-bundle.crt`, generated from Oracle Wallet Manager is shipped with OracleAS Portal. This file contains an initial list of trusted server certificates that might be used for navigating to some secure servers using HTTPS. However, because it is not a complete list of all possible server certificates that exist on the Web, this file must be configured or extended to recognize any additional trusted server certificates for any new trusted sites that are visited.

- Configuring Oracle Advanced Security for the Web Clipping Provider

Section 6.1.9.2 in Chapter 6, "Securing OracleAS Portal" describes configuring Oracle Advanced Security Option (ASO) to secure and encrypt the channel between itself (on the middle tier) and the database, which hosts the Web Clipping Repository.

I.1.1 Configuring the Web Clipping Repository

Web clippings have definitions that must be stored persistently in the Web Clipping Repository hosted by an Oracle Database.

As portal administrator, you can configure the Web Clipping Repository using the Web Clipping Test page at:

```
http://<host>:<port>/portalTools/webClipping/providers/webClipping
```

The Web Clipping Test page automatically detects whether or not the Web Clipping provider is configured with a valid database to host the Web Clipping repository. If it is not, the **Status** column for the Web Clipping Repository displays **Not Configured**. The **Edit** link in the **Actions** column enables you to configure the connection parameters for the repository database in the Edit Provider Page. Figure I–1 shows the Web Clipping Test page:

Figure I–1 Web Clipping - Provider Test Page

By default, the Web Clipping provider is configured to use the Oracle Application Server Infrastructure database as the Web Clipping Repository. To change the repository configuration, you need to obtain a database user account from your database administrator before you begin configuring.

To change the repository settings:

1. In the **Provider Configuration** section, the **Setting** column contains the field **Web Clipping Repository.** Click its corresponding **Edit** link in the **Actions** column.
2. In the **Repository Settings** section of the **Edit Provider: webClipping** page, you specify the database connection information for the Web Clipping provider. Select one of the following choices for the **Repository Target** database:

- **OracleAS Infrastructure Database (default):** If you select this option, no other connection parameters need be specified.

  When you select this option, the Web Clipping Repository is stored in the PORTAL user schema of the OracleAS Infrastructure database.

- **Other Oracle9*i* (or later) Database**: If you select this option, you must first create a database user in which to store the Web Clipping Repository. Execute the following SQL*Plus commands as a user with SYSDBA privileges:

  ```
  ORACLE_HOME/bin/sqlplus /nolog
  SQL> connect sys/password as sysdba
  SQL>  CREATE USER username IDENTIFIED BY password;
  SQL> GRANT connect, restricted session, resource TO username;
  ```

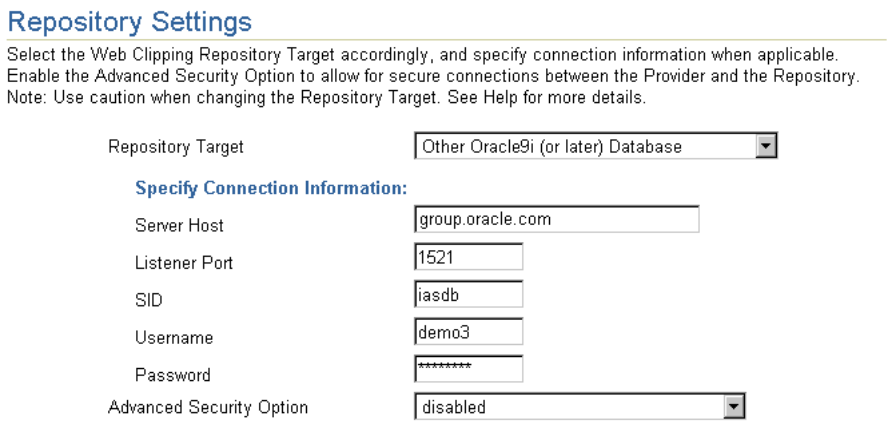
  Then, in the **Other Oracle9*i* (or later) Database** field, specify the following connection parameters:

  - **Server Host:** The name of the database server
  - **Listener Port:** The listener port for the database server
  - **SID:** The database SID
  - **Username:** The database user name
  - **Password:** The password for the database user

3. If you require a secure database connection, in the **Advanced Security Option** field, select **enabled (secure database connection).** See Section 6.1.9.2, "Configuring Oracle Advanced Security for the Web Clipping Provider" for more information about configuring the Advanced Security Option.

   Figure I–2 shows the **Repository Settings** section of the Edit Provider page:

Figure I–2 Web Clipping - Repository Settings



4. Click **OK** to save the settings and return to the Web Clipping Test page.

If you have specified the same database for your Repository Target as the one used for a PDK 9.0.2.4.0 installation, as part of your upgrade, you will be notified that a Repository Upgrade also needs to take place. The Web Clipping Test page will contain a **Upgrade (from 9.0.2.4.0)** link that enables you to do a one-click upgrade for installing new tables, and migrating existing Clipping Definitions to the latest versions.

Note: After the upgrade, the Clipping Definitions stored in the Web Clipping Repository will no longer work with PDK 9.0.2.4.0.

In the **Proxy Settings** section of this page, you specify proxy information, if necessary. See Section I.1.3, "Configuring HTTP or HTTPS Proxy Settings" for more information.

I.1.2 Registering the Web Clipping Provider (PDK Only)

Perform this task only if you have downloaded and installed the Web Clipping provider as part of OracleAS PDK.

Note: If you installed OracleAS Portal as part of the Oracle Application Server installation, the Web Clipping provider is registered by default under the Portlet Builder folder in the Portlet Repository.

After you configure the Web Clipping provider, you must register it as a portlet provider in the OracleAS Portal instance. You can then add portlets to a portal page.

To register the Web Clipping provider, perform the following steps:

1. Log on to OracleAS Portal.
2. Navigate to the Build tab on the OracleAS Portal Home Page. By default, the Providers portlet is available on this page. If it is not available here, then use the Portal Search feature to locate the Providers portlet.
3. In the Providers portlet, click **Register a Portlet Provider**.
4. Follow the instructions in the registration wizard, and specify the Web Clipping provider registration settings as shown in Table I–1.

Note: To distinguish this Web Clipping provider from the seeded Web Clipping provider under the Portlet Builder folder in the Portlet Repository, you must give it a distinguishable name, for example, `Web Clipping provider on host ABC`.

Table I–2 lists the settings that you must specify.

Table I–1 The Web Clipping Provider Registration Settings

Field	Value
Name	`WebClipping_ABC`
Display Name	`Web Clipping provider on host ABC`
Timeout	`200 seconds`
Timeout Message	`Web Clipping provider on host ABC timed out`
Implementation Style	`Web`

Table I–1 (Cont.) The Web Clipping Provider Registration Settings

Field	Value

URL	`http://<host>:<port>/portalTools/webClipping/providers/webClipping`
	If you want the portlet content to be cached, then specify the Web Cache URL host name and port number to point to the OracleAS Web Cache instance. For example:
	`http://<cache_instance_name>:<cache_port>/portalTools/webClipping/providers/webClipping`

The user has the same identity in the Web providers application as in the single sign-on identity	Select this option.
Select User to send user-specific information to the provider	Select this option.
Login Frequency	Once Per User Session
Require Proxy	`No` (if no proxy is required to contact the Provider Adapter)

5. Click **Finish**.

You can now use the Web Clipping provider to add portlets to a portal page. The Web Clipping provider is registered, by default, under the Portlet Staging Area folder in the Portlet Repository.

I.1.3 Configuring HTTP or HTTPS Proxy Settings

Your HTTP or HTTPS proxy settings must be set to allow the Web Clipping Studio to connect to Web sites outside of your firewall. You can specify the settings in the following ways:

- Using the Web Clipping Test page (see Section I.1.3.1)
- Manually editing the `provider.xml` file (see Section I.1.3.2)

I.1.3.1 Configuring Proxy Settings Using the Web Clipping Test Page

As portal administrator, you can configure proxy settings using the Web Clipping Test page at:

`http://<host>:<port>/portalTools/webClipping/providers/webClipping`

To configure proxy settings using the test page:

1. In the **Provider Configuration** section, the **Setting** column contains the field **HTTP Proxy.** Click its corresponding **Edit** link in the **Actions** column.
2. In the **Proxy Settings** section of the Edit Provider: webClipping page, enter the proxy settings for the Web Clipping provider:
   - **HTTP Proxy:** Enter the name of a proxy server if one is required for the provider to access the Web site being clipped. A proxy server provides access to content from servers outside a firewall.
   - **Port:** Enter the port number for the HTTP Proxy server. If you specify an HTTP Proxy server, but not a port, the HTTP Proxy port is set to default port number `80`.

- **No Proxy for:** Enter the name of any domain to which you can directly connect, bypassing a proxy server. Domain names are the part of a URL that contain the names of a business, or organization, or government agency, for example:

  ```
  .company.com, .companycorp.com, .us.company.com
  ```

  You can also use this field to specify a list of host names to restrict content, as described in Section I.1.3.3.

- **Requires Authentication:** Select this check box for proxy servers that require authentication before access. Then, enter information in the following fields:

  – **Type:** Choose the type of proxy server this provider will use: **Basic** or **Digest.**

  – **Realm:** Enter the name of the realm of the proxy server that will be accessed by the user. If you do not know the name of the realm, contact the administrator of the proxy server.

  – **Login Scheme:** Choose one of the following options:

    Use login below for all users: By choosing this option, the login information you enter in the **Username** or **Password** fields will be used for all users. Users will not see the Proxy Authentication section on the Edit Defaults and the Personalize pages.

    Require login for all users: By choosing this option, you do not enter any information in the **Username** or **Password** fields. Page designers must enter authentication information for the proxy server on the Edit Defaults page when they define the portlet. They only need to enter the information once for the Portlet instance. End users must enter authentication information for the proxy server on the Personalize page.

    Require login for all users - use login below for public users: By choosing this option, the login information you enter in the **Username** or **Password** fields will be used for all public users only. If the user is not public, they must enter authentication information for the proxy server on the Personalize page.

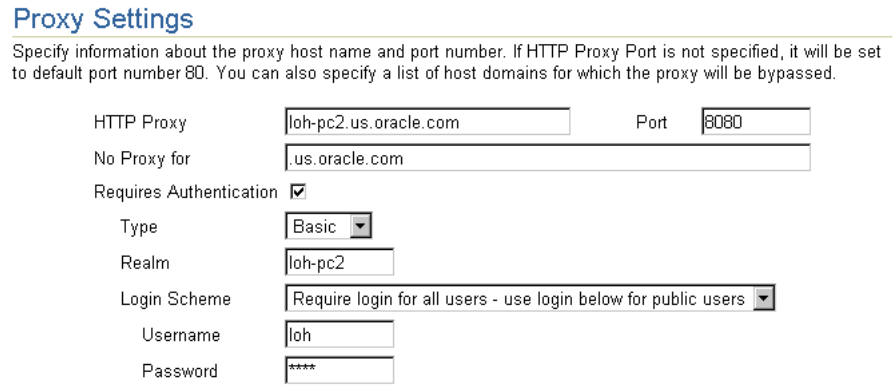
  – **Username:** Enter the user name to log in to the proxy server.

  – **Password:** Enter the password for the specified user name.

  If you select the **Requires Authentication** check box and require users to log in (that is, you choose either **Require login for all users** or **Require login for all users - use login below for public users**), you must configure the Web Clipping repository. To configure the repository, see Section I.1.1.

Figure I–3 shows the **Proxy Settings** section of the Edit Provider page:

Figure I–3 Web Clipping - Proxy Settings



3. Click **OK** to save the settings and return to the Web Clipping Test page.

I.1.3.2 Setting Proxy Settings Manually

As the portal administrator, you can set proxy settings manually according to your HTTP or HTTPS configuration. Edit the appropriate entries in the `provider.xml` file located in the following directory:

On UNIX:

ORACLE_HOME`/j2ee/OC4J_Portal/applications/`
`portalTools/webClipping/WEB-INF/providers/webClipping`

On Windows:

ORACLE_HOME`\j2ee\OC4J_Portal\applications\`
`portalTools\webClipping\WEB-INF\providers\webClipping`

The following example shows the relevant portion of `provider.xml`:

```
<proxyInfo class="oracle.portal.provider.v2.ProxyInformation">
<httpProxyHost>proxy_hostname</httpProxyHost>
<httpProxyPort>proxy_portnum</httpProxyPort>
<dontProxyFor>list_of_proxies</dontProxyFor>
<proxyUser>proxy_username</proxyUser>
<proxyPassword>proxy_password</proxyPassword>
<proxyType>basic_or_digest</proxyType>
<proxyUseAuth>true_or_false</proxyUseAuth>
<proxyUseGlobal>true_or_false</proxyUseGlobal>
<proxyRealm>realm_name</proxyRealm>
</proxyInfo>
```

The elements have the following meanings and values:

- `httpProxyHost`: The name of a proxy server if one is required to make a URL connection with the provider, for example, when OracleAS Portal requests one of the provider's portlets. A proxy server provides access to content from servers outside a firewall.
- `httpProxyPort`: The port number for the proxy server. If you specify an httpProxyHost, but not a port, httpProxyPort is set to default port number `80`.
- `dontProxyFor`: The name of any domain to which you can directly connect, bypassing a proxy server. Domain names are the part of a URL that contain the names of a business, or organization, or government agency, for example:

```
.company.com, .companycorp.com, .us.company.com
```

You can also use this element to specify a list of host names to restrict content, as described in Section I.1.3.3.

- `proxyUser`: The user name to log in to the proxy server.
- `proxyPassword`: The password for the specified user name.
- `proxyType`: The type of proxy server this provider will use. Valid values are `Basic` or `Digest`.
- `proxyUseAuth`: Whether or not login information is required to access this proxy server. A value of `true` specifies that login information is required. A value of `false` indicates that login information is not required.
- `proxyUseGlobal`: Whether or not the user name and password are required for public users or all users and how the user name and password are used. This element is only relevant if `proxyUseAuth` is set to `true`.
  - If the value of `proxyUseGlobal` is `true`, the *`proxy_username`* specified in `proxyUser` and the *`proxy_password`* specified in `proxyPassword` are used for all users. The `proxyUser` and `proxyPassword` elements must be present in the file and must contain a user name and password. Users will not see the Proxy Authentication section on the Edit Defaults and Personalize pages.
  - If the value is `false` and the `proxyUser` and `proxyPassword` elements are present in the file, the specified *`proxy_username`* and *`proxy_password`* are used for public users only. If the user is not public, they must enter authentication information for the proxy server on the Personalize page.
  - If the value is `false` and the `proxyUser` and `proxyPassword` elements are *not* present in the file, all users must log in. Page designers must log in using the Proxy Authentication section on the Edit Defaults page when they define the portlet. End users must enter authentication information for the proxy server on the Personalize page.

  If the value is `false`, regardless of whether or not the `proxyUser` and `proxyPassword` elements are present in the file, you must configure the Web Clipping repository. To configure the repository, see Section I.1.1.
- `proxyRealm`: The name of the realm of the proxy server that will be accessed by the user. If you do not know the name of the realm, contact the administrator of the proxy server.

I.1.3.3 Restricting Users from Clipping Content from Unauthorized External Web Sites

To restrict users from clipping content from unauthorized external Web sites, Web Clipping uses the proxy exception list. This is available only for environments that utilize a proxy server to reach external Web sites. (Usually, you use the proxy exception list to specify any domain to which you can directly connect, bypassing a proxy server.)

To add an external Web site to the proxy exception list:

1. Go to the Edit Provider: webClipping page, as described in Section I.1.3.1.
2. In the **Proxy Settings** section, for the **No proxy for** field, enter the Web sites which you want to restrict.
3. Click **OK** to save the settings and return to the Web Clipping Test page.

Users attempting to reach a Web site in one of the listed domains, from the Web Clipping Studio, will receive an HTTP timeout error.

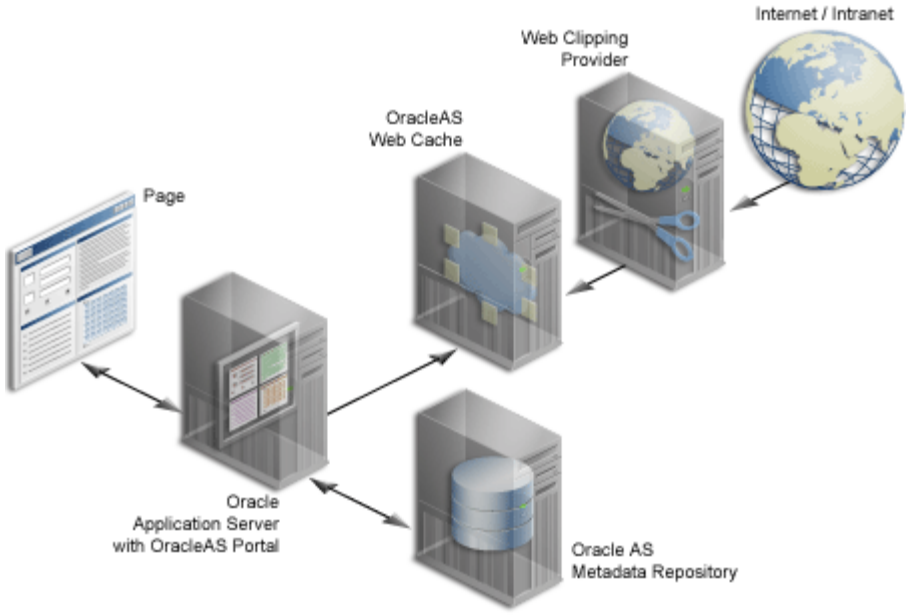
I.1.4 Configuring Caching

By default, validation-based caching is used through OracleAS Portal for all Web Clipping portlets. With **validation-based caching**, the Parallel Page Engine (PPE) contacts the OracleAS Portal provider to determine if the cached item is still valid.

If you have Oracle Application Server Web Cache installed, you can elect to use **invalidation-based caching** through OracleAS Web Cache. Note that each type of caching is mutually exclusive; that is, you can use only one or the other, but not both.

With invalidation-based caching, an item remains in the cache until the cache receives notification that the item needs to be refreshed. For example, if the Web Clipping portlet contains content that is updated on a regular basis, the cache will be invalidated. Invalidation-based caching, as shown in Figure I–4, decreases the number of requests the Web Clipping provider must entertain while maintaining the same network traffic for each round trip involving PPE. Depending on your deployment scenario, you may prefer using one caching method over the other.

Figure I–4 Invalidation-Based Caching Provided by OracleAS Web Cache



See Section 1.3, "Understanding Caching in OracleAS Portal" and Section 5.8, "Managing OracleAS Portal Content Cached in OracleAS Web Cache" for more information about caching.

By default, the Web Clipping provider uses portal caching (validation-based caching). To use OracleAS Web Cache (invalidation-based caching), see one of the following sections to configure caching:

- Section I.1.4.1, "Configuring Caching Using the Web Clipping Test Page"
- Section I.1.4.2, "Configuring Caching Manually"

If you decide to use OracleAS Web Cache to cache Web clipping content, as a final step, you must use the Portal Navigator and change the connect string for the provider URL to point to a URL with the OracleAS Web Cache port. Usually, the OracleAS Web

Cache port is `7778`. Check the Application Server Control Console Ports page to verify this value. For example:

`http://`*`host`*`:`*`webcacheport`*`/portalTools/webClipping/providers/webClipping`

In this configuration, OracleAS Web Cache caches Web clipping content between the OracleAS Portal instance and the Web Clipping provider.

I.1.4.1 Configuring Caching Using the Web Clipping Test Page

As portal administrator, you can configure caching using the Web Clipping Test Page at:

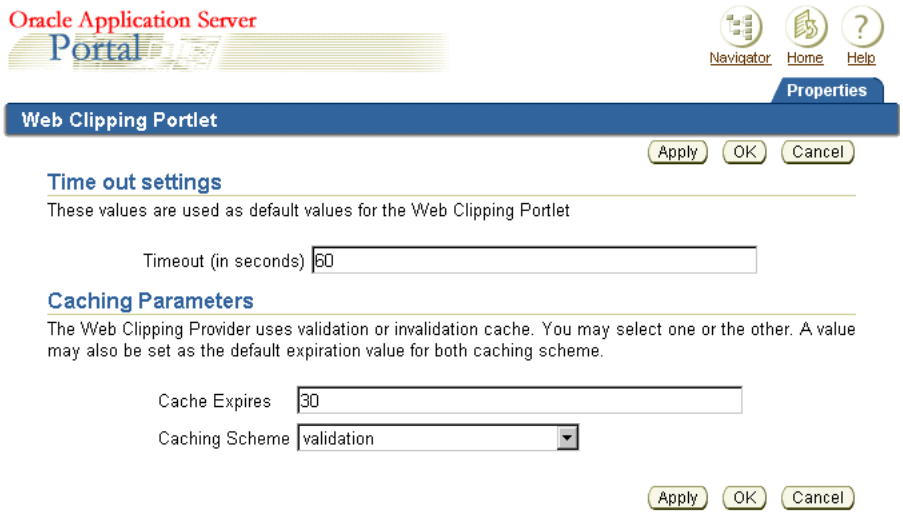
`http://<host>:<port>/portalTools/webClipping/providers/webClipping`

To configure caching:

1. In the **Provider Configuration** section, the **Setting** column contains the field **Portlet Caching**. Click its corresponding **Edit** link in the **Actions** column.
2. In the **Time out settings** section of the Web Clipping Portlet page, enter the number of seconds OracleAS Portal should try to connect to Web Clipping portlets before displaying a timeout message.
3. In the **Caching Parameters** section, specify a cache expires value in the **Cache Expires** field. The default value is 30 minutes.
4. In the **Caching Scheme** field, select the caching scheme (either **validation** or **invalidation (requires OracleAS Web Cache).**)

   Figure I–5 shows the Web Clipping Portlet page:

Figure I–5 Web Clipping Portlet Page



5. Click **OK** to save the settings and return to Web Clipping Test page.

I.1.4.2 Configuring Caching Manually

To manually enable caching with OracleAS Web Cache, take the following steps:

1. Check the `cache.xml` file in the following directory to verify the accurate values of the invalidation host and port number:

   On UNIX:

ORACLE_HOME/portal/conf

On Windows:

ORACLE_HOME\portal\conf

2. Edit the `provider.xml` file located in the following directory:

   On UNIX:

   ORACLE_HOME/j2ee/OC4J_Portal/applications/
   portalTools/webClipping/WEB-INF/providers/webClipping

   On Windows:

   ORACLE_HOME\j2ee\OC4J_Portal\applications\
   portalTools\webClipping\WEB-INF\providers\webClipping

   In the `provider.xml` file:

   a. Search for the `useInvalidationCaching` tag and set its value to `true` to enable OracleAS Web Cache invalidation-based caching.

   b. Search for the `cacheExpires` tag and set a default value if you wish to modify that value. This value is in minutes.

I.2 Configuring OmniPortlet

OmniPortlet is a subcomponent of OracleAS Portal that enables page designers and developers to easily publish data from various data sources using a variety of layouts. You can base OmniPortlet on almost any kind of data source, such as Web Services, spreadsheet (character-separated values), XML, and even application data from an existing Web page. OmniPortlet enables page designers and content contributors to do the following:

- Access secured data
- Format data using a variety of layouts including charts, forms, and reports
- Accept page parameters
- Pass parameters
- Raise events
- Expose personalizable settings to page viewers

Before you use OmniPortlet, you must perform a few administrative tasks, including:

- Section I.2.1, "Configuring the OmniPortlet Provider"
- Section I.2.2, "Performing Optional OmniPortlet Configurations"
- Section I.2.3, "Registering the OmniPortlet Provider (PDK Only)"
- Section I.2.4, "Configuring the OmniPortlet Provider to Access Other Relational Databases Using DataDirect JDBC Drivers"

I.2.1 Configuring the OmniPortlet Provider

Before you use OmniPortlet, you must perform certain administrative tasks depending on how you installed OracleAS Portal. You must perform most of these tasks only if you installed the Oracle Application Server Portal Developer Kit on a standalone Oracle Containers for J2EE (OC4J) instance, or if you downloaded the preconfigured

standalone OC4J with OracleAS PDK. This manual refers to these two installations as PDK-Only installations. If you installed OracleAS Portal as part of the Oracle Application Server release, then most of the configurations are already done.

You can perform these administrative tasks using the OmniPortlet Provider Test page. The test page detects whether the OmniPortlet provider is properly configured or not. You can access the test page by clicking **OmniPortlet Provider** on the Portal Tools Welcome Page, located at

`http://<host>:<port>/portalTools`

or, by directly accessing the following URL:

`http://<host>:<port>/portalTools/omniPortlet/providers/omniPortlet`

The administrative tasks that you must perform are as follows:

- Section I.2.1.1, "Configuring HTTP or HTTPS Proxy Settings"
- Section I.2.1.2, "Configuring the Secured Data Repository (PDK only)"
- Section I.2.1.3, "Configuring Caching (PDK Only)"
- Section I.2.1.4, "Configuring OmniPortlet to Access HTTPS URLs"

I.2.1.1 Configuring HTTP or HTTPS Proxy Settings

If a proxy server is required for the provider to make a URL connection to a data source outside the firewall, then you must set up the HTTP or HTTPS Proxy.

> **Note:** Setting up the HTTP or HTTPS Proxy is applicable only to URL-based data sources such as XML, CSV, Web Services, and Web Page data sources.

If you have configured the proxy settings for Web Clipping provider already using the Web Clipping Test page, then you need not configure them again for the OmniPortlet provider. If you configured the proxy settings for the Web Clipping provider manually, that is by editing the Web Clipping `provider.xml` file, then you need to perform the same steps for the OmniPortlet provider by editing the OmniPortlet `provider.xml` file located in the following directory:

ORACLE_HOME`/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet`

To configure the proxy settings, click **Edit** next to HTTP Proxy on the OmniPortlet Provider Test page. An Edit Provider page is displayed, on which you can specify the HTTP Proxy Host, Proxy Port, and other proxy information. For an installation where the proxy server access requires authentication, you can specify the user name and password at the following levels:

- Global: Login information is shared for all users. Specify the global login information in the Edit Provider page.
- For each user: Users must set their own login information on an OmniPortlet instance as follows:
  - For page viewers, set the login information in the Personalize page.
  - For page designers, in the Edit Defaults mode, click the **Source** tab, then set the login information.

You do not need to restart OC4J for the new settings to take effect. Refer to the Web Clipping provider section, Section I.1.3, "Configuring HTTP or HTTPS Proxy Settings" for more information about configuring proxy settings.

I.2.1.2 Configuring the Secured Data Repository (PDK only)

OmniPortlet uses the Web Clipping repository to store credentials needed to access secured data. You must configure the repository if you intend to use the Web Page data source or work with secured data (for example, a SQL database or a URL-based data source with HTTP basic authentication). If you have configured the Web Clipping Repository already, you do not need to configure the Secured Data Repository again because they are the same repository.

To configure the repository, click **Edit** next to Secured Data Repository on the OmniPortlet Provider Test page. The Edit Provider page is displayed, on which you can enter the repository information. Refer to the Web Clipping provider section, Section I.1.1, "Configuring the Web Clipping Repository" for more information.

I.2.1.3 Configuring Caching (PDK Only)

If you want the portlet content cached using invalidation-based caching, then an OracleAS Web Cache instance must be configured as a front end to the provider.

You must use the URL host name and port number to point to the OracleAS Web Cache instance when registering the provider as shown in the following example:

```
http://<cache_instance_name>:<cache_
port>/portalTools/omniPortlet/providers/omniPortlet
```

This task must be performed when registering the OmniPortlet provider. Refer to Section I.2.3, "Registering the OmniPortlet Provider (PDK Only)" for details about registering the OmniPortlet provider.

When an OmniPortlet definition is altered either through the Edit Defaults or Personalize page, the provider generates a request that invalidates and removes the portlet content from the cache. The invalidation request is sent to the invalidation port of the OracleAS Web Cache instance. Information about the OracleAS Web Cache instance is maintained in the `cache.xml` file in the *ORACLE_HOME*`/portal/conf` directory. If the Web Cache invalidation settings change, then you must update this file. The following example shows sample entries in the `cache.xml` file:

```
<?xml version="1.0"?>
<webcache>
    <invalidation
        host="<cache_instance_name>"
        port="<cache_invalidation_port>"
        authorization="<obfuscated_username_password>"/>
</webcache>
```

Where:

- *<cache_instance_name>* is the host name of the Web Cache instance.
- *<cache_invalidation_port>* is the Web Cache invalidation port.
- *<obfuscated_username_password>* is the invalidator user name and password.

  For information about obfuscating the invalidator user name and password, refer to the *Oracle Application Server Portal Developer's Guide*.

I.2.1.4 Configuring OmniPortlet to Access HTTPS URLs

You can configure OmniPortlet to access data through HTTPS URLs by doing the following:

- Adding Certificates for Trusted Sites
- Library For HTTPS Access (PDK Only)

Adding Certificates for Trusted Sites

Perform this task only once, either for Web Clipping or for OmniPortlet.

The trusted server certificate file, `ca-bundle.crt`, generated from Oracle Wallet Manager is shipped with OracleAS Portal. This file contains an initial list of trusted server certificates that may be used for navigating to some secure servers using HTTPS. However, because the file does not contain a complete list of all possible server certificates that exist on the Web, this file must be configured or extended to recognize any additional trusted server certificates for any new trusted sites that are visited. Refer to Section 6.1.9.1, "Adding Certificates for Trusted Sites" for details about configuring or extending the trusted certificate file.

Copying the Library for HTTPS Access (PDK Only)

To access HTTPS URLs, OmniPortlet needs access to the files, `njssl10.dll` (for Windows) or `libnjssl10.so` (for UNIX).

For providers running on Windows, the `njssl10.dll` file must be present in a folder defined in the `PATH` environment variable. If it is not available, then you can copy this library from the *`ORACLE_HOME`*`/bin` directory.

For providers running on UNIX, the `libnjssl10.so` file must be present in the folder defined in `LD_LIBRARY_PATH` environment variable. If it is not available, then you can copy this library from the *`ORACLE_HOME`*`/lib` directory.

After copying the library, you must restart the OC4J instance.

I.2.2 Performing Optional OmniPortlet Configurations

The following configuration tasks are optional.

Setting the LocalePersonalizationLevel

The default setting for the `LocalePersonalizationLevel` of OmniPortlet and Simple Parameter Form is `none`. This mode indicates that, when you edit the portlet defaults by using the Edit Defaults mode, the changes apply to all users, regardless of the current portal session language or the locale of the browser. If you do not want the changes made using the Edit Defaults mode to apply to all users, then you can modify the `provider.xml` file for the OmniPortlet provider by setting the `LocalePersonalizationLevel` tag to `language` or `locale`. For more information about these settings, refer to the PDK-Java Release Notes available at the following location:

`ORACLE_HOME``/portal/pdkjava/v2/pdkjava.v2.releasenotes.html`

Reverting to Using Graph Class to Render Chart Layout Style

OmniPortlet now uses ThinGraph class to render the chart layout style. This provides better multilingual support and removes middle-tier font dependence. The chart style produced in OracleAS Portal 10.1.2.0.2 may be different from that of earlier versions where Graph class was used. To display the old chart style, you must revert to using the old Graph class.

To do this, perform the following steps:

1. Navigate to the `provider.xml` file located in the directory *ORACLE_HOME*`/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet`.
2. Edit the `useThinGraph` tag as follows:

   ```
   <useThinGraph>false</useThinGraph>
   ```

3. Save the `provider.xml` file.

I.2.3 Registering the OmniPortlet Provider (PDK Only)

Perform this task only if you have downloaded and installed the OmniPortlet provider as part of OracleAS PDK.

> **Note:** If you installed OracleAS Portal as part of the Oracle Application Server installation, the OmniPortlet provider is registered by default under the Portlet Builder folder in the Portlet Repository.

After you configure the OmniPortlet provider, you must register it as a portlet provider in the OracleAS Portal instance. You can then add portlets to a portal page.

To register the OmniPortlet provider, perform the following steps:

1. Log on to OracleAS Portal.
2. Navigate to the Build tab on the OracleAS Portal Home Page. By default, the Providers portlet is available on this page. If it is not available here, then use the Portal Search feature to locate the Providers portlet.
3. In the Providers portlet, click **Register a Portlet Provider**.
4. Follow the instructions in the registration wizard, and specify the OmniPortlet provider registration settings.

> **Note:** To distinguish this OmniPortlet provider from the seeded OmniPortlet provider under the Portlet Builder folder in the Portlet Repository, you must give it a distinguishable name, for example, `OmniPortlet provider on host ABC`.

Table I–2 lists the settings that you must specify.

Table I–2 The OmniPortlet Provider Registration Settings

Field	Value
Name	`OmniPortlet_ABC`
Display Name	`OmniPortlet provider on host ABC`
Timeout	`200 seconds`
Timeout Message	`OmniPortlet provider on host ABC timed out`
Implementation Style	`Web`

Table I–2 (Cont.) The OmniPortlet Provider Registration Settings

Field	Value

URL	`http://<host>:<port>/portalTools/omniPortlet/providers/omniPortlet`
	If you want the portlet content to be cached, then specify the Web Cache URL name and port number to point to the OracleAS Web Cache instance. For example:
	`http://<cache_instance_name>:<cache_port>/portalTools/omniPortlet/providers/omniPortlet`

The user has the same identity in the Web providers application as in the single sign-on identity	Select this option.
Select User to send user-specific information to the provider	Select this option.
Login Frequency	`Never`
Require Proxy	`No` (if no proxy is required to contact the Provider Adapter)

5. Click **Finish**.

You can now use the OmniPortlet provider to add portlets to a portal page. The OmniPortlet provider is registered, by default, under the Portlet Staging Area folder in the Portlet Repository.

I.2.4 Configuring the OmniPortlet Provider to Access Other Relational Databases Using DataDirect JDBC Drivers

The OmniPortlet SQL data source is preconfigured to access Oracle Databases using the Oracle JDBC driver, and ODBC data sources using the JDBC-ODBC driver from Sun Microsystems. Perform this step if you want to access other relational databases with OmniPortlet using the DataDirect JDBC drivers, which are the preferred drivers.

You can configure the OmniPortlet SQL data source to access other relational databases by using DataDirect JDBC drivers. To do this, perform the following steps:

- Installing DataDirect JDBC Drivers
- Registering DataDirect Drivers in OmniPortlet

> **See Also:** For a list of supported databases, Certification Matrix for Oracle Application Server and DataDirect JDBC available at
>
> `http://www.oracle.com/technology/tech/java/oc4j/htdocs/datadirect-jdbc-certification.html`

I.2.4.1 Installing DataDirect JDBC Drivers

DataDirect JDBC drivers are packaged in a single ZIP file containing the different drivers used to access supported databases. Download the ZIP file from the following location:

`http://www.oracle.com/technology/software/products/ias/htdocs/utilsoft.html`

To install DataDirect JDBC drivers, perform the following steps:

1. Unzip the contents of the ZIP file into a temporary directory, for example `/temp/datadirect`.
2. Create the *ORACLE_HOME*`/j2ee/<`*OC4J_INSTANCE_HOME*`>/applib` directory if it does not already exist.
3. From the `/temp/datadirect/lib` directory, copy the DataDirect JDBC drivers to the *ORACLE_HOME*`/j2ee/<`*OC4J_INSTANCE_HOME*`>/applib` directory.
4. Check the configuration of the OC4J_Portal instance to ensure that the DataDirect libraries are loaded. To do this, perform the following steps:
   a. Open the *ORACLE_HOME*`/j2ee/<`*OC4J_INSTANCE_HOME*`>/config/application.xml` file. This file is used to configure all applications in this instance.
   b. Add the XML entry, `<library path="../applib"/>`, to the file if it does not already exist.

I.2.4.2 Registering DataDirect Drivers in OmniPortlet

OmniPortlet is implemented as a Web provider and all the configuration properties are stored in the `provider.xml` file. To use DataDirect JDBC drivers with OmniPortlet, you must register these drivers in the `provider.xml` file.

To register the new DataDirect JDBC drivers, perform the following steps:

1. Back up the file, *ORACLE_HOME*`/j2ee/<`*OC4J_INSTANCE_HOME*`>/applications/portalTools/omniPortlet/WEB-INF/providers/omniPortlet/provider.xml`, and then open the file.
2. Add the drivers that you want to use for the SQL data source configuration entry. To do this, perform the following:
   a. Search for the XML tag, `driverInfo`.
   b. Add a new entry after the last `driverInfo` tag.

   Following is an example showing a Microsoft SQL Server entry:

   - For OmniPortlet version 9.0.4.1 or later:

```
<!-- registration of DataDirect Connect for JDBC SQL Server driver -->
<driverInfo class="oracle.webdb.reformlet.data.jdbc.JDBCDriverInfo">
  <name>Microsoft SQL Server</name>
  <sourceDataBase>other</sourceDataBase>
  <subProtocol>sqlserver</subProtocol>
  <connectString>mainProtocol:subProtocol://databaseName</connectString>
  <driverClassName>com.oracle.ias.jdbc.sqlserver.SQLServerDriver
  </driverClassName>
  <dataSourceClassName>com.oracle.ias.jdbcx.sqlserver.SQLServerDataSource
  </dataSourceClassName>
  <connHandlerClass>oracle.webdb.reformlet.data.jdbc.JDBCConnectionHandler
  </connHandlerClass>
  <connPoolSize>5</connPoolSize>
  <loginTimeOut>30</loginTimeOut>
</driverInfo>
```

   - For OmniPortlet versions before 9.0.4.1:

```
<!-- registration of DataDirect Connect for JDBC SQL Server driver -->
<driverInfo class="oracle.webdb.reformlet.data.jdbc.JDBCDriverInfo">
  <name>Microsoft SQL Server</name>
  <sourceDataBase>other</sourceDataBase>
```

```
  <subProtocol>sqlserver</subProtocol>
  <connectString>mainProtocol:subProtocol://databaseName</connectString>
  <driverClassName>com.oracle.ias.jdbc.sqlserver.SQLServerDriver
  </driverClassName>
  <connHandlerClass>
  oracle.webdb.reformlet.data.jdbc.JDBCODBCConnectionHandler
  </connHandlerClass>
  <connPoolSize>5</connPoolSize>
  <loginTimeOut>30</loginTimeOut>
</driverInfo>
```

Table I–3 describes the parameters in the `driverInfo` property.

Table I–3 Parameters in the driverInfo Property

Parameter	Description
`name`	Name of the database you want to use. This name will be used on the Source tab of the OmniPortlet wizard.
`sourceDataBase`	Internal value. Set the value to `other`.
`subProtocol`	JDBC subprotocol name used by OmniPortlet to create the connection string, for example `sqlserver`, `sybase`, or `db2`. To get the list of subprotocol names, refer to the DataDirect JDBC driver documentation using the links provided at the end of this section.
`connectString`	Description of the connect string format. For DataDirect drivers, the format is: `mainProtocol:subProtocol://databaseName`
`driverClassName`	Name of the driver class. To get the different values, refer to the DataDirect JDBC driver documentation using the links provided at the end of this section.
`dataSourceClassName`	Name of the data source class that implements connection pooling. This parameter is only available in OmniPortlet version 9.0.4.1 or later. Refer to Table I–4 for the right data source class name for your driver.

`connHandlerClass`	Class used by OmniPortlet to manage the driver and connection pooling. The value is either of the following:
	■ For OmniPortlet version 9.0.4.1 or later:
	`oracle.webdb.reformlet.data.jdbc.JDBCConnectionHandler`
	■ For OmniPortlet versions before 9.0.4.1:
	`oracle.webdb.reformlet.data.jdbc.JDBCODBCConnectionHandler`

`connPoolSize`	Minimum number of connections that are opened by the connection pool.
`loginTimeOut`	Maximum time, in seconds, that this data source will wait while attempting to connect to a database.

Table I–4 lists the values for the `driverClassName` and `dataSourceClassName` properties for specific DataDirect JDBC drivers.

Table I–4 Parameters and Values for driverClassName and dataSourceClassName

DataDirect Drivers Supported	Properties
Microsoft SQL Server	■ Parameter: `driverClassName` Value: `com.oracle.ias.jdbc.sqlserver.SQLServerDriver` ■ Parameter: `dataSourceClassName` Value: `com.oracle.ias.jdbcx.sqlserver.SQLServerDataSource`
Sybase	■ Parameter: `driverClassName` Value: `com.oracle.ias.jdbc.sybase.SybaseDriver` ■ Parameter: `dataSourceClassName` Value: `com.oracle.ias.jdbcx.sybase.SybaseDataSource`
DB2	■ Parameter: `driverClassName` Value: `com.oracle.ias.jdbc.db2.DB2Driver` ■ Parameter: `dataSourceClassName` Value: `com.oracle.ias.jdbcx.db2.DB2DataSource`
Informix	■ Parameter: `driverClassName` Value: `com.oracle.ias.jdbc.informix.InformixDriver` ■ Parameter: `dataSourceClassName` Value: `com.oracle.ias.jdbcx.informix.InformixDataSource`

3. Save the `provider.xml` file.
4. Stop and start the Oracle Application Server instance.

Note: If you are using OmniPortlet in a multiple nodes configuration, for example, in a clustering or load-balancing environment, then you must manually copy the `provider.xml` file on each node.

See Also: For more information on DataDirect JDBC drivers, refer to the following documentation:

- The white paper titled "*How to Use DataDirect JDBC Drivers with OmniPortlet*" on the Portlet Development page on Portal Center at

  `http://www.oracle.com/technology/products/ias/portal/portlet_development_10g1014.html`

- Certification Matrix for Oracle Application Server and DataDirect JDBC available at

  `http://www.oracle.com/technology/tech/java/oc4j/htdocs/datadirect-jdbc-certification.html`

- The OC4J page on the Oracle Technology Network (OTN) available at

  `http://www.oracle.com/technology/software/products/ias/htdocs/utilsoft.html`

- How to use DataDirect JDBC drivers in OmniPortlet in

  Oracle Application Server Portal Developer's Guide

Troubleshooting Information

If you encounter errors or problems when configuring or using the OmniPortlet provider, refer to Appendix K, "Troubleshooting OracleAS Portal" for troubleshooting information.

J

Setting Up and Maintaining a Virtual Private Portal

This appendix walks you through the steps for setting up and maintaining a virtual private portal (VPP). It works through a case study to demonstrate the various tasks involved in setting up and maintaining a virtual private portal (hosted portal).

The following topics are covered in this appendix:

J.1 Overview of Hosting

Before reviewing the tasks, let us look at why hosting features are beneficial and what some of the known limitations are.

J.1.1 Why Use Hosting?

Consider an Application Service Provider (ASP), Acme, that wants to provide portal services for its customers. Acme wants to give its customers the flexibility to build and customize cost-effective and secure portals. They want customers to create and manage their own users, information, and portal pages securely.

Dedicated portal or database instances for each customer would provide the security they require. Traditionally, implementing fully isolated portal environments for multiple organizations within a company required a dedicated database instance for each organization. This proved expensive in terms of hardware and manpower resources, especially when the number of organizations was large. Manpower and hardware costs fast increased as their customer base grew. A single shared instance is obviously more manageable, but will not provide the level of isolation required to host multiple organizations securely.

A single instance is cheaper and easier to manage, but a traditional portal solution requires complex security rules to be written into the application. What Acme needs is

the best of both worlds. VPP provides a platform for ASPs a more manageable way for large Enterprise IT departments to host departmental intranet or extranet portal sites. Oracle Application Server Portal introduces a more cost-effective and manageable solution for hosting multiple organizations and provides the benefits of a shared instance model with complete security. When using VPP you are required to add subscribers. A Subscriber is a company that signs up with an ASP (Application Service Provider) and receives a stripe on a hosted Oracle Application Server Portal.

J.1.2 Known Limitations

Although a shared instance model has many benefits, there are several things to consider before implementing a VPP environment.

Hosted technology will completely isolate each subscriber or identity realm. The VPP will prompt each user to enter their company ID and name, or set a particular context before portal retrieves any content. The scope of the content and data is limited to the subscriber's context. The portal is secured at the subscriber level and does not allow sharing of any data between one subscriber and another. Sharing of data is not allowed for security reasons. For example, VPP should not be used if Company A and Company B need to share documents.

Making repetitive changes to all subscribers is also more complex. From an administrative perspective, user interface manipulation of the portal must be done for each subscriber.

Example J–1 Scenario 1 - Administering Many Subscribers

Company A, Company B, and Company C have identical portal pages 1, 2, and 3. When an administrator logs in to Company A to change the layout of page 1, it only affects that particular subscriber. To change page 1 on Company B, the administrator for Company B would need to perform the same changes using the portal user interface. Logging in to each subscriber is easy, as long as the number of subscribers is small. When administering lots of subscribers, the best way to manage many portal sites is to update the pages by using portal APIs, or through an automated testing tool to make the changes on each site. This makes managing a large number of subscribers very complex.

Example J–2 Scenario 2 - Upgrading

Another area of consideration is upgrading. When performing portal repository upgrades, every subscriber's data must be upgraded. If Acme hosts 1000 subscribers, the portal repository upgrade must go through every subscriber's data before successful completion.

Assume that an average single repository upgrade takes 10 minutes. As it is not possible to split the upgrade process on a single instance, VPP portal repository upgrade will loop through all existing subscribers. So in this example, it would take 10 minutes for a single upgrade multiplied by the number of subscribers: 10 times 1000 will be 10000 minutes. This has huge downtime implications.

Therefore, small manageable deployments of VPP with about 50 subscribers for each instance is recommended. In cases where you must exceed the recommended maximum number, consider deploying multiple VPP instances. To choose a reasonable set of downtime windows to apply changes and upgrades, it is also recommended that you segment on a time zone basis. You can configure multiple portal repositories that could be upgraded individually. So, you can upgrade 50 subscribers on an instance without affecting all the 1000 subscribers at the same time.

Note: For clarity, the terms Subscribers and Identity Realm are used interchangeably in this document.

J.2 Overview of Steps to Perform for Virtual Private Portals

The following subsections outline the tasks involved in setting up and managing your hosted installation.

- Enabling Hosting
- Setting Up Users and Groups
- Adding Subscribers
- Removing Subscribers
- Advanced Features
- Pre-Installation Checklist
- Using Oracle Directory Manager

J.2.1 Enabling Hosting

- Enable hosting on OracleAS Portal and the OracleAS Single Sign-On (SSO) server.
- Create a basic structure on Oracle Internet Directory for ASP user/group support.

J.2.2 Setting Up Users and Groups

- Set up the virtual private portal with a support and administration infrastructure and users. The ASP uses these to administer the virtual private portal on behalf of their customers.

J.2.3 Adding Subscribers

- Create a new subscriber stripe in the OracleAS Portal and SSO schemas. This step includes copying objects like page groups, pages, portlet and providers information so that the default environment and pages can be pre-defined.
- Create a new Oracle Internet Directory subscriber tree, and establish required portal entries in Oracle Internet Directory (for example, seeded groups, users, and privileges).
- Copy ASP groups/users for the new subscriber in Oracle Internet Directory (for example, creating mirror entries, assigning privileges, and so on).

J.2.4 Removing Subscribers

- Remove a subscriber's data in OracleAS Portal and SSO schema.
- Delete the whole subscriber sub tree in Oracle Internet Directory.

J.2.5 Advanced Features

- WebDAV enables you to use a URL address as a transparent read and write medium where content can be checked out, edited, and checked in.

- Oracle Ultra Search provides uniform search-and-locate capabilities over multiple repositories, such as Oracle Databases, IMAP servers, Web pages, disk files, and portal page groups.

J.2.6 Pre-Installation Checklist

Before running the scripts to enable virtual private portals, first gather the information to run them. Table J–1 lists and describes the parameters.

Table J–1 Parameters

Parameters	Description
-pc	Database connect string for portal schema, in format of `<host>:<port>:<sid>`, where `<host>` is a fully qualified domain name. This is a mandatory parameter.
-ps	OracleAS Portal schema name. By default, it is `portal`.
-pw	OracleAS Portal schema password. By default it is the value of `-ps` parameter. See Section J.2.7, "Using Oracle Directory Manager" for help with this parameter.
-sc	Database connect string for SSO schema, in format of `<host>:<port>:<sid>`, where <host> is a fully qualified domain name. By default, it is the value of `-pc` parameter.
-ss	SSO schema name. By default, it is the `orasso`.
-sw	SSO schema password. By default is the value of `-ss` parameter. See Section J.2.7, "Using Oracle Directory Manager" for help with this parameter.
-h	Oracle Internet Directory server host name. This is a mandatory parameter.
-p	Oracle Internet Directory server port number. By default, it is `389` or `4032`.
-d	Oracle Internet Directory bind DN. By default, it is cn=orcladmin. This DN must have Oracle Internet Directory administrative privilege, for example, privilege to create new subscribers.
-w	Password for Oracle Internet Directory bind DN. By default, it is `welcome1`.

J.2.7 Using Oracle Directory Manager

To begin the process, use the Oracle Directory Manager (ODM). The ODM is a GUI tool to help you administer Oracle Internet Directory. To obtain the passwords for portal and orasso users:

1. Launch the Oracle Directory Manager.
   - In the first field, provide the Oracle Internet Directory bind DN (parameter `-d`). By default, it is `cn=orcladmin`. This DN must have Oracle Internet Directory admin privilege, for example, privilege to create new subscribers.
   - In the second field, provide the password for Oracle Internet Directory bind DN (parameter `-w`). By default, it is `welcome1`.
   - In the third field, select your Oracle Internet Directory instance. If you have not defined your Oracle Internet Directory instance, click the icon to right of the field and give the server host name (parameter `-h`) and port number (parameter `-p`) that Oracle Internet Directory is running on. By default, the port is `389` or `4032`.

2. Once you have logged into Oracle Internet Directory, navigate through the menu tree. Entry Management > cn=OracleContext > cn=Products > cn=IAS.

   Cn=IAS Infrastructure Databases > orclReferenceName=name of Oracle Internet Directory database.

3. Continue to navigate the tree.
4. Click the orasso user name.
5. In the right pane, find the section called **orclpasswordattribute**. This is the password for the orasso user (parameter –sw).
6. Click the portal user name.
7. In the right pane, there is a section called **orclpasswordattribute**. This is the password for the portal user (parameter –pw).

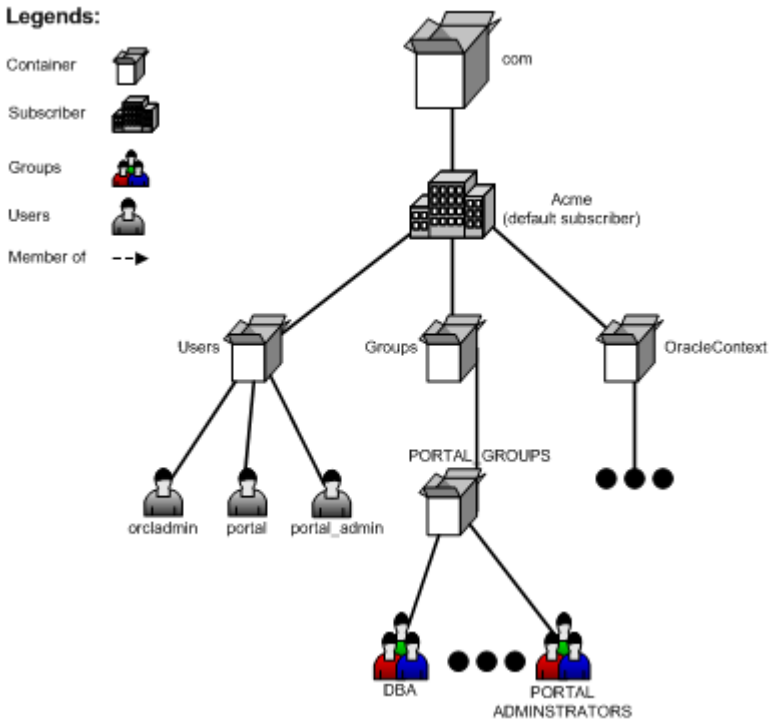
J.3 Enabling Hosting on an Out-of-the-Box Portal

To begin an out-of-the-box OracleAS Portal installation, enable hosting on the portal. A C-shell script is provided, that:

- Enables hosting on OracleAS Portal and the OracleAS Single Sign-On server.
- Creates a basic structure on Oracle Internet Directory for ASP user/group support

To illustrate how the script works, here is what the Oracle Internet Directory tree looks like before running the script:

Figure J–1 Oracle Internet Directory Tree Before Running the Script



To run the script, type the following at the UNIX command line:

```
cd ORACLE_HOME/portal/admin/plsql/wwhost
./enblhstg.csh -pc portaldb.acme.com:1521:portaldb -ps portal -pw ky8T5sr3 -sc
```

```
portaldb.acme.com:1521:portaldb -ss orasso -sw hA6fHjE2 -h oid.acme.com -p 389 -d
"cn=orcladmin" -w welcome1
```

Update the sample login page with the multiple-realm version of the page, by editing the `login.jsp` page located at *ORACLE_HOME*`/j2ee/OC4J_SECURITY/applications/sso/web/jsp`.

> **Note:** In a distributed deployment, this file is located on the Single Sign-On middle tier.

After making a backup copy of the file, remove comments from this section:

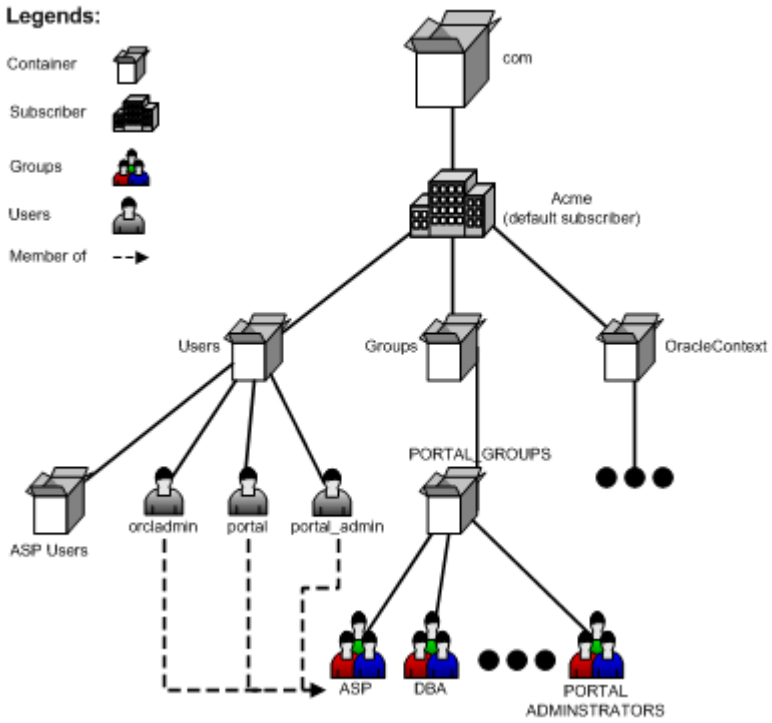
```
<!-- UNCOMMENT TO ENABLE MULTIPLE REALM SUPPORT
<tr>
<label>
<th id="c6"><font
class="OraFieldText"><%=msgBundle.getString(ServerMsgID.COMPANY_
LBL)%></font></th>
<td headers="c6"> <INPUT TYPE="text"  SIZE="30" MAXLENGTH="50"
NAME="subscribername" value=""></td>
</label>
</tr>
-->
```

Stop and start the Single Sign-On middle tier.

Refer to Section J.8, "Parameters for the Scripts" for a detailed explanation of parameters.

After running the script, the Oracle Internet Directory tree looks like Figure J–2:

Figure J–2 Oracle Internet Directory Tree After Running the Script



Now the portal instance is hosting enabled. If you go to the portal login screen, you see three input fields (Username, Password, and Company). To login as the default subscriber, you can type acme in the Company field, or leave it blank.

The default subscriber is reserved for the ASP for administrative purposes. For each of its customers, a new subscriber must be created before people can login and use it.

J.4 ASP Users And Groups

Because Acme is the ASP it needs to have a support and administration infrastructure that administers the virtual private portal on behalf of the customers. The virtual private portal provides support for ASP users and groups such that administration of multiple subscriber portals is simple.

These ASP users and groups can have different levels of administrative access into the virtual private portals of the subscribers (customers) of Acme. ASP users can be split into groups according to the privileges they need. For example, Alice needs privileges to manage user accounts; Bob and Joe need privileges to manage page contents. These privilege groups are ASP groups.

These ASP users and groups allow an ASP user to log in to multiple subscribers using a single set of credentials (user name and password), and have the same set of pre-defined privileges in all subscribers. This is achieved by creating mirror entries of ASP users and groups across all subscribers. The user and group entries can then be kept in sync through pre-supplied scripts (see ASP Sync Script section). Note: The synchronization (script or automatic) only synchronizes the users and groups, not the portal privileges.

The following sections show how to set up the virtual private portal with ASP user/group support for Acme, and some other tasks you may want to perform:

- Setting Up ASP Users and Groups
- Restrictions

J.4.1 Setting Up ASP Users and Groups

The master entry for ASP Users and groups resides under the default subscriber. Because these users and groups will have additional access (not all users in the default subscriber can log in to all subscribers) you must set up ASP users/groups explicitly.

When you enabled hosting on your portal, the script creates a group called ASP under the default subscriber's Oracle Internet Directory sub-tree, which is a placeholder for ASP user/group support. You need this to set up ASP users/groups. From now on, this placeholder group will be referred to as the ASP group. Let's look at some examples where Acme could use ASP users and groups:

- Alice needs to manage user accounts for all subscribers.
- Bob and Joe need to manage pages for all subscribers.
- Tom needs to log in to all subscribers but only have normal authenticated user privileges.

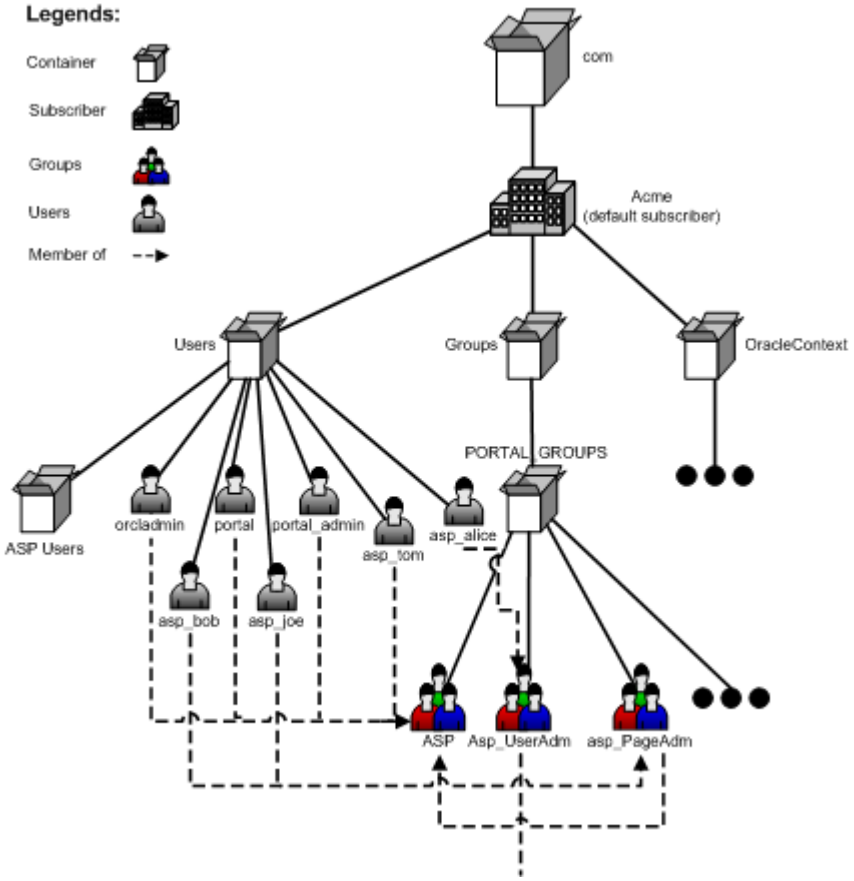
To accomplish this, do the following:

- Create users asp_alice, asp_bob, asp_joe and asp_tom in default subscriber.
- Create group asp_UserAdm in default subscriber and assign it privileges to manage user accounts; and also create group asp_PageAdm in default subscriber and assign it privileges to manage pages.

- Add asp_alice as member of asp_UserAdm group.
- Add asp_bob and asp_joe as members of asp_PageAdm group.
- Add asp_UserAdm and asp_PageAdm as members of the ASP group.
- Add user asp_tom as member of the ASP group.

Now you have set up ASP users and groups. The Oracle Internet Directory tree looks like Figure J–3:

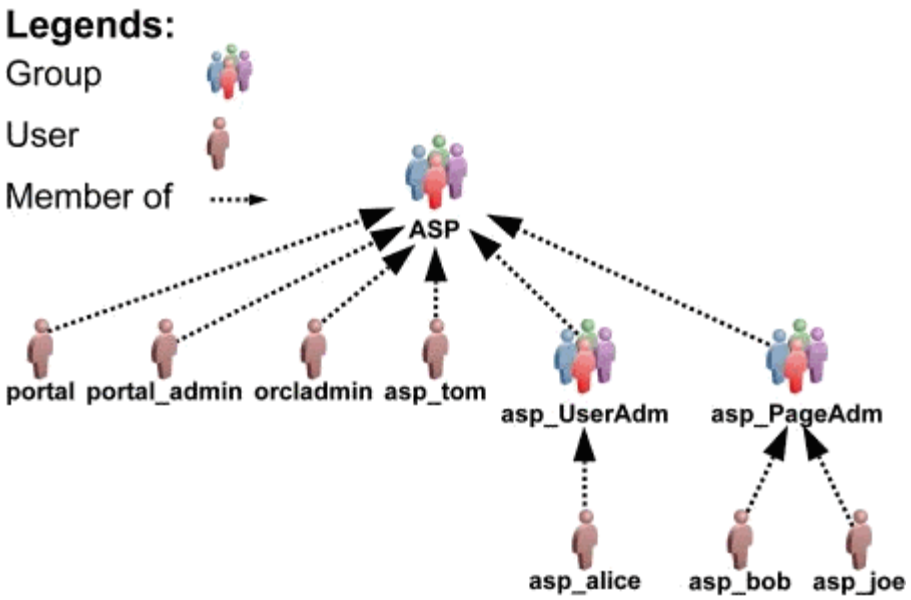
Figure J–3 Oracle Internet Directory Tree with Users and Groups



More precisely, ASP users/groups are defined as follows:

- ASP groups are either the ASP group itself or its direct group members.
- ASP users that are a direct user member of any ASP group.

Figure J–4 Membership Structure of Acme Users and Groups



By default, the portal bootstrap users are members of the ASP group, which means that they are by default ASP users. For more information on portal bootstrap users portal, portal_admin, and orcladmin, see the *Oracle Application Server Administrator's Guide* and the *Oracle Application Server Portal User's Guide*.

When you add a new subscriber, the portal Add Subscriber script automatically creates mirror entries for those ASP users/groups in the new subscriber. Then those users can login and have the corresponding privileges.

J.4.2 Restrictions

There are some restrictions on ASP users/groups set up:

- ASP users and groups can be no more than two levels deep. That is, groups that are not direct members of the ASP group or users that are not direct members of any ASP groups are ignored during mirror entry creation.
- Oracle Internet Directory mandates that user names must be unique (case insensitive) within each subscriber, including those of ASP users. For example, you cannot have two users in subscriber CompanyA called Bob or bob. Because ASP users have mirror entries in every subscriber, use special names for ASP users to prevent user name collisions. This is reflected throughout this document with names such as asp_bob, asp_joe, and the like.
- For similar reasons, use special names for ASP groups, for example, asp_ PageAdmin, asp_UserAdmin, and the like. Because hosting scripts handle ASP groups dynamically, do not make a portal seeded group into an ASP group. If you need an ASP group with similar privileges, create a new group and make it a member of the seeded group.
- Manage nondefault subscribers' ASP users and groups only with hosting scripts. Do not manually modify those users, or groups, or both.
- The ASP group is only a placeholder for all ASP users and groups, and is not designed for privilege purposes. Do not assign privileges to the ASP group. Those privileges are not propagated to other subscribers.

J.5 Adding Subscribers

Acme has now set up its ASP users and groups and has enabled the portal for hosting. The next step is to add the customers as subscribers of the virtual private portal. For each of Acme's customers (CompanyA, CompanyB), you will create a new subscriber in the portal. A C-shell script is provided, that:

- Creates a new stripe in the OracleAS Portal and SSO schemas. This step copies objects like page groups, pages, portlet and providers information, and the like.
- Creates a new Oracle Internet Directory subscriber tree and establishes required portal entries in Oracle Internet Directory (for example, seeded groups, users, and privileges).
- Copies ASP groups/users to the new subscriber in Oracle Internet Directory (for example, creating mirror entries, assigning privileges, and so on).

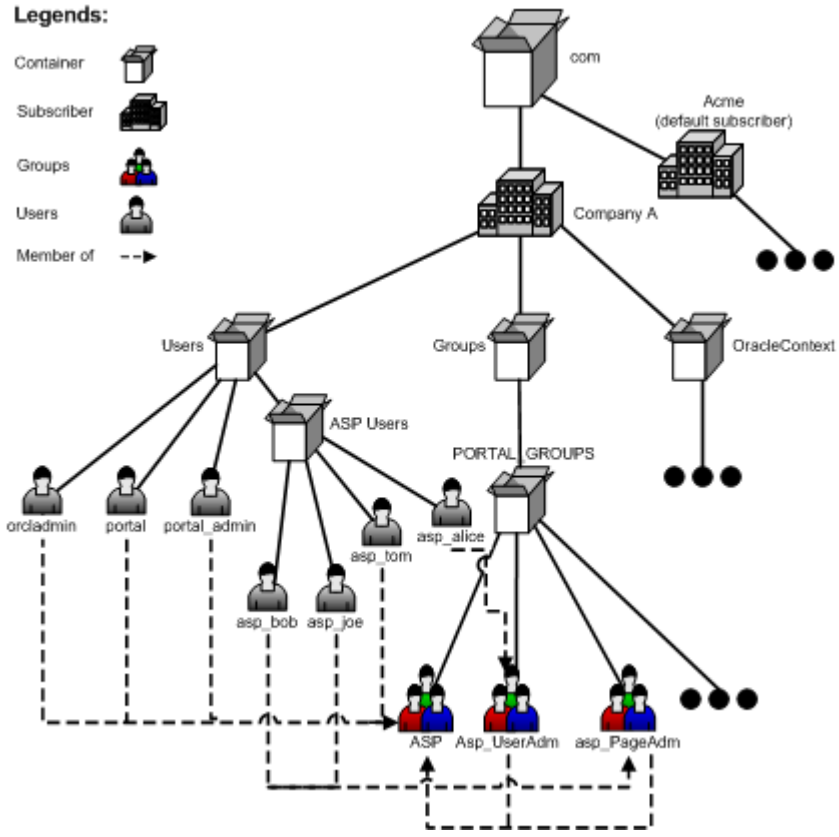
To add subscriber CompanyA, enter the following commands at the UNIX command line:

```
> cd ORACLE_HOME/portal/admin/plsql/wwhost
>./addsub.csh -name CompanyA -id 1001 -type all -pc
 portaldb.acme.com:1521:portaldb -pp change_on_install -ps portal -pw ky8T5sr3 -sc
 portaldb.acme.com:1521:portaldb -sp change_on_install -ss orasso -sw hA6fHjE2 -a
 portal.portaldb.portaldb.acme.com -h oid.acme.com -p 389 -d "cn=orcladmin" -w
 welcome1 -rc "cn=OracleContext" -sd acme -tp ORACLE_HOME/ldap/schema/oid/
```

Refer to Section J.8, "Parameters for the Scripts" for an explanation of parameters.

Check the output, and contact Oracle technical support if there is any error. After running the script, subscriber CompanyA exists in both OracleAS Portal and Oracle Internet Directory. The Oracle Internet Directory tree looks like Figure J–5:

Figure J–5 CompanyA in Both Portal and Oracle Internet Directory



Run the same script to create subscriber CompanyB.

Now you have set up a virtual private portal with two subscribers. To try the ASP users, log in to CompanyA as user asp_alice using the same password as when you created it in default subscriber. Alice should have privileges to do user management tasks.

J.6 Advanced Operations on a Virtual Private Portal

Specific topics covered in this section include:

- Managing ASP Users and Groups
- Removing Subscribers
- Using WebDAV in the Virtual Private Portal
- Using Oracle Ultra Search with the Virtual Private Portal
- Setting Up Directory Integration Platform for the Virtual Private Portal
- Partially Prepare (Pre-Cook) Subscribers

J.6.1 Managing ASP Users and Groups

After you have set up all the subscribers, there could be several types of changes to the ASP users/groups structure. For example:

- Bob changed his password in default subscriber, and you must synchronize the new password in all other subscribers.

- Joe left Acme and should no longer be able to login as an ASP user.
- The service contract changed and the ASP is no longer responsible for user account problems. So, the asp_UserAdm group is no longer needed.
- When ASP users/groups are changed in the default subscriber, you must use a provided script to synchronize the changes in all other subscribers.

The synchronization script has three options:

- Password Sync
- Delta (Structure Changes) Sync
- Complete Sync

J.6.1.1 Password Sync

If you use password sync, the script updates passwords for all the ASP user's mirror entries using the password in the default subscriber.

For the first example in the preceding text, you can synchronize Bob's new password using the following commands at the UNIX command line:

```
> cd ORACLE_HOME/portal/admin/plsql/wwhost

>./syncasp.csh -pc portaldb.acme.com:1521:portaldb -ps portal -pw ky8T5sr3 -h
oid.acme.com -p 389 -d "cn=orcladmin" -w welcome1 -mode pwd -u asp_bob
```

Alternatively, if you have enabled the Directory Integration Platform, it synchronizes ASP user password changes automatically so that you do not need to run this script.

J.6.1.2 Delta (Structure Changes) Sync

If you use delta sync, the script searches for users/groups that have been changed in the default subscriber and applies the changes to all other subscribers.

For departing employees or service contract changes, you can synchronize the new ASP structure using the following commands at the UNIX command line:

```
> cd ORACLE_HOME/portal/admin/plsql/wwhost

>./syncasp.csh -pc portaldb.acme.com:1521:portaldb -ps portal -pw ky8T5sr3 -h
oid.acme.com -p 389 -d "cn=orcladmin" -w welcome1 -mode dif
```

Delta sync assumes consistency and integrity of old ASP structures. That is, if the old ASP structure in each subscriber is consistent and correct, then delta sync does the job correctly. Otherwise, you could use the Complete Sync option, which is slower than the delta sync.

J.6.1.3 Complete Sync

The script takes the ASP structure of default subscriber and overwrites the structures of all other subscribers. If delta sync failed to synchronize the ASP structure, consider using this option.

For departing employees or service contract changes, you can synchronize the new ASP structure using the following commands at the UNIX command line:

```
> cd ORACLE_HOME/portal/admin/plsql/wwhost

>./syncasp.csh -pc portaldb.acme.com:1521:portaldb -ps portal -pw ky8T5sr3 -h
oid.acme.com -p 389 -d "cn=orcladmin" -w welcome1 -mode all
```

Complete sync is slower than delta sync, so use only when necessary.

J.6.2 Removing Subscribers

If a subscriber in a portal is no longer needed, or errors occurred during the subscriber creation, you can permanently remove a subscriber using a provided script. The script does the following:

- Removes the subscriber's data in OracleAS Portal and SSO schema.
- Deletes the whole subscriber sub tree in Oracle Internet Directory.

For example, to remove a subscriber called *nowhere*, type the following command at the UNIX command line. However, once you remove a subscriber, there is no way to restore it except from any backups taken of the Oracle Database on which the virtual private portal instance has been installed.

```
> cd ORACLE_HOME/portal/admin/plsql/wwhost

>./rmsub.csh -name nowhere -pc portaldb.acme.com:1521:portaldb -pp change_on_
install -ps portal -sc portaldb.acme.com:1521:portaldb -sp change_on_install -ss
 orasso -a portal.portaldb.portaldb.acme.com -h oid.acme.com -p 389 -d
 "cn=orcladmin" -w welcome1 -cs 1000
```

See Section J.8, "Parameters for the Scripts" for an explanation of parameters.

J.6.3 Using WebDAV in the Virtual Private Portal

WebDAV is a protocol that supports distributed authoring and versioning. With WebDAV the Internet becomes a transparent read and write medium where content can be checked out, edited, and checked in to a URL address. For details about how WebDAV works with OracleAS Portal and how to set up WebDAV, see the *Oracle Application Server Portal User's Guide*.

Setting up WebDAV in a virtual private portal is the same as setting up WebDAV in an out-of-the-box portal.

Connecting to WebDAV in a virtual private portal is similar to that in an out-of-the-box portal. The only difference is that, when connecting to WebDAV in a virtual private portal, you use:

```
"<username>@<subscriber_name>" as the username, instead of just ...
"<username>" as required in an out-of-the-box portal.
```

For example, to connect to WebDAV using user Joe in subscriber CompanyA, use `joe@CompanyA` as the user name and Joe's password as the password.

When different subscribers use the same URL for WebDAV connection, the client side operating system may cache the connection. For example, if you connected to WebDAV using user portal_admin@acme on a Windows 2000 PC, you may not be able to connect to WebDAV in subscriber CompanyA as user joe@CompanyA because of the operating system cache. For details about how to clear an operating system cache and stored user name and password, see your operating system documents.

J.6.4 Using Oracle Ultra Search with the Virtual Private Portal

Oracle Ultra Search provides uniform search-and-locate capabilities over multiple repositories (Oracle Databases, IMAP servers, Web pages, disk files, and portal page groups). To use Oracle Ultra Search in a virtual private portal, do the following:

- Set up branded URL for different subscribers.

- Enable hosting on your Oracle Ultra Search instance.

To enable hosting on your Oracle Ultra Search instance, run the following commands in the UNIX command line:

```
> cd ORACLE_HOME/ultrasearch/admin
> sqlplus /nolog @wk0host.sql [schema_name] [schema_password] [connect_string] E
```

where:

[schema_name] - Oracle Ultra Search schema name

[schema_password] - Oracle Ultra Search schema password

[connect_string] - Database connect string of your Oracle Ultra Search instance

E – Enables hosting for an Oracle Ultra Search instance

Currently, Oracle Ultra Search does not support ASP users/groups.

J.6.5 Setting Up Directory Integration Platform for the Virtual Private Portal

The Directory Integration Platform is a comprehensive framework that performs synchronization between various directories and directory-enabled applications. One of the services it provides is Provisioning Integration, which can send notifications about directory events to Directory Enabled Applications.

See Also: *Oracle Internet Directory Administrator's Guide*

In an out-of-the-box OracleAS Portal installation, the Directory Integration Platform is enabled. If you have disabled Directory Integration Platform for a virtual private portal, do the following to re-enable Directory Integration Platform:

1. Run the provided script that enables Directory Integration Platform on existing subscribers.

   For example, for UNIX:

   ```
   enbldip.csh -pc portaldb.acme.com:1521:portaldb -pp change_on_install -ps
   portal -h oid.acme.com -p 389 -d "cn=orcladmin" -w welcome -enable
   ```

2. Uncomment the calls to oidprovtool in the addsub.csh and rmsub.csh, so that those two scripts take care of Directory Integration Platform profile entries when you add/remove subscribers.

   To do this:

   a. Open the two files in your editor.

   b. Search for lines with the oidprovtool string.

   c. Uncomment those lines.

Also, you can do the following to disable Directory Integration Platform on all subscribers in your portal:

1. Run the provided script in at the UNIX command line as follows:

   ```
   enbldip.csh -pc portaldb.acme.com:1521:portaldb -pp change_on_install -ps
   portal -h oid.acme.com -p 389 -d "cn=orcladmin" -w welcome -disable
   ```

2. Comment out the calls to oidprovtool in addsub.csh and rmsub.csh, so that those two scripts ignore Directory Integration Platform profile entries when you add or remove subscribers.

To do this:

a. Open the two files in your editor.

b. Search for lines with `oidprovtool`.

c. Comment these lines out.

J.6.6 Partially Prepare (Pre-Cook) Subscribers

Creating a new subscriber by running the `addsub.csh` script can take a few minutes based on how the computer where OracleAS Portal, Oracle Internet Directory, and OracleAS Single Sign-On are installed is configured. Along with the data operations that occur when a new subscriber is created, most ASPs have some administrative provisioning and subscriber-specific customizations that they perform when a subscriber is created.

To expedite subscriber registration, the virtual private portal allows ASPs to partially prepare the subscribers. This is done so that when an ASP is registered, the subscriber need only perform post registration customizations and directly assign a virtual private portal stripe to that subscriber. The virtual private portal provides a database-only mode in the `addsub.csh` script where the data copying is performed on the portal and SSO databases. When the ASP is ready to assign a stripe to a subscriber, it can complete the subscriber creation by running the `addsub.csh` script using the LDAP mode.

To partially prepare a subscriber in portal and SSO databases, use the `-type` parameter in `addsub.csh`. For example, type the following at the UNIX command line:

```
> cd ORACLE_HOME/portal/admin/plsql/wwhost
>./addsub.csh -name TEMP_COMPANY -id 1003 -type db -pc
 portaldb.acme.com:1521:portaldb -pp change_on_install -ps portal -pw ky8T5sr3 -sc
 portaldb.acme.com:1521:portaldb -sp change_on_install -ss orasso -sw hA6fHjE2 -h
 oid.acme.com -p 389 -d "cn=orcladmin" -w welcome1 -rc "cn=OracleContext" -sd acme
```

You can use a temporary name for company name, like (TEMP_COMPANY) as used in the preceding example. Later, when a customer (example, CompanyC) comes, you can run the following command at the UNIX command line:

```
> cd ORACLE_HOME/portal/admin/plsql/wwhost
>./addsub.csh -name CompanyC -id 1003 -type ldap -pc
 portaldb.acme.com:1521:portaldb -pp change_on_install -ps portal -pw ky8T5sr3 -sc
 portaldb.acme.com:1521:portaldb -sp change_on_install -ss orasso -sw hA6fHjE2 -a
 portal.portaldb.portaldb.acme.com -h oid.acme.com -p 389 -d "cn=orcladmin" -w
 welcome1 -rc "cn=OracleContext" -sd acme -tp ORACLE_HOME/ldap/schema/oid/
```

You must use the same subscriber ID when you partially prepare the subscriber, and give the real name of your customer (CompanyC). The new name will replace the old name (TEMP_COMPANY in the preceding example). The script will create an Oracle Internet Directory subscriber tree for CompanyC and synchronize the Oracle Internet Directory settings to portal schema, which takes less time than creating the subscriber from scratch.

You do have to partially prepare (using `-type db` option) the subscriber before you can use run `addsub.csh` with `-type ldap` option.

OracleAS Portal Middle-Tier Installation on the Virtual Private Portal

Follow the steps in Chapter 3, "Installing OracleAS Portal" to run the OracleAS Portal middle-tier installation. Then, you have to use the `ptlconfig` tool to reconfigure

your OracleAS Portal middle-tier settings. See Appendix A, "Using the Portal Dependency Settings Tool and File" for details about using the `ptlconfig` tool.

The OracleAS Portal middle-tier installation can be run against a virtual private portal.

J.7 Restrictions

The following subsections provide summaries of the restrictions on the different virtual private portal scripts and operations:

- Scripts
- ASP Users/Groups Support
- Add Subscriber
- Remove Subscriber
- Upgrade

J.7.1 Scripts

The virtual private portal configuration and provisioning scripts currently only run on a UNIX C-shell environment.

J.7.2 ASP Users/Groups Support

- The top level ASP group must not have any Oracle Internet Directory privileges assigned to it. Privileges are not copies or synchronized across subscribers. Privileges of the sub-groups of the ASP group are synchronized and copied.
- Any modifications to the ASP user and group structure in Oracle Internet Directory that are performed on any other subscriber other than the default subscriber are not preserved when the subscriber synchronization scripts are run.
- Portal seeded groups should not be designated as ASP groups.

J.7.3 Add Subscriber

Names of new subscribers must be unique within Oracle Internet Directory.

J.7.4 Remove Subscriber

This script cannot be used to remove the default subscriber. To do that, use the Portal Dependency Settings Tool, `ptlconfig`. See Section A.1, "Portal Dependency Settings Tool" for details about using the `ptlconfig` tool.

J.7.5 Upgrade

When performing an upgrade from OracleAS Portal release 9.0.2.*x* to 9.0.4.*x*, the Oracle Text indexes need to be re-created. See Section 8.3.4.1, "Creating All Oracle Text Indexes Using ctxcrind.sql" for information on running the `ctxcrind.sql` script to re-create all the Oracle Text indexes.

J.8 Parameters for the Scripts

Table J–2 through Table J–6 list and describe all the parameters for the scripts provided for administering a virtual private portal. These scripts can be found in the *`ORACLE_HOME`*`/portal/admin/plsql/wwhost` directory.

Note: To produce a list of the parameters for any of the scripts, run the script in your UNIX shell without any parameters. If you want the output of the scripts to be saved to a log file, type `|& tee <log_filename>` at the end of the command, replacing <log_filename> with the name of your log file.

Table J–2 enblhstg.csh

Parameter	Description
`-pc`	Database connect string for OracleAS Portal schema, in format of `<host>:<port>:<sid>`, where `<host>` is a fully qualified domain name. This is a mandatory parameter.
`-ps`	OracleAS Portal schema name. By default, it is `portal`.
`-pw`	OracleAS Portal schema password. By default it is the value of `-ps` parameter.
`-sc`	Database connect string for SSO schema, in format of `<host>:<port>:<sid>`, where `<host>` is a fully qualified domain name. By default, it is the value of `-pc` parameter.
`-ss`	SSO schema name. By default, it is the orasso
`-sw`	SSO schema password. By default, it is the value of `-ss` parameter.
`-h`	Oracle Internet Directory server host name. This is a mandatory parameter.
`-p`	Oracle Internet Directory server port number. By default, it is `389`.
`-d`	Oracle Internet Directory bind DN. By default, it is `cn=orcladmin`. This DN should have Oracle Internet Directory admin privilege, for example, privilege to create new subscribers.
`-w`	Password for Oracle Internet Directory bind DN. By default, it is `welcome1`.

Table J–3 addsub.csh

Parameter	Description
`-name`	Oracle Internet Directory nickname of the new subscriber. This is a mandatory parameter. This name must not have been used by any other subscriber
`-id`	Internal ID for the new subscriber, which is used within OracleAS Portal and OracleAS Single Sign-On. This is a mandatory parameter. It should not have been used by any other subscriber in OracleAS Portal or OracleAS Single Sign-On schema.
`-type`	Valid values are: ■ `db` – only copy seed data in OracleAS Portal and OracleAS Single Sign-On schemas. ■ `ldap` – create Oracle Internet Directory entries for OracleAS Portal and OracleAS Single Sign-On. You can run the script only using `-type ldap` option after you add temporary subscriber using `-type db` option. ■ `all` – default value, do both `db` and `ldap` types jobs.

Table J–3 (Cont.) addsub.csh

Parameter	Description
-pc	Database connect string for OracleAS Portal schema, in format of <host>:<port>:<sid>, where <host> is a fully qualified domain name. This is a mandatory parameter.
-pp	SYS user password of portal instance. By default, change_on_install.
-ps	OracleAS Portal schema name. By default, portal.
-pw	OracleAS Portal schema password. By default it is the value of -ps parameter.
-sc	Database connect string for SSO schema, in format of <host>:<port>:<sid>, where <host> is a fully qualified domain name. By default, it is the value of -pc parameter.
-sp	SYS user password of SSO instance. By default, if OracleAS Single Sign-On and OracleAS Portal are on different database instances, it is change_on_install; if OracleAS Single Sign-On and OracleAS Portal use the same database instance, it is the value of -pp parameter.
-ss	SSO schema name. By default, it is orasso.
-sw	SSO schema password. By default is the value of -ss parameter.
-a	Portal Application name. By default, it is <portal_schema>.<sid>.<dbhost>
-h	Oracle Internet Directory server host name. This is a mandatory parameter.
-p	Oracle Internet Directory server port number. By default, it is 389.
-d	Oracle Internet Directory bind DN. By default, it is cn=orcladmin. This DN must have Oracle Internet Directory admin privilege, for example, privilege to create new subscribers.
-w	Password for Oracle Internet Directory bind DN. By default, it is welcome1.
-rc	Oracle Internet Directory root context DN. By default, it is cn=OracleContext
-sd	Oracle Internet Directory nickname of the template subscriber. By default, it is the nickname of the portal default subscriber.
-tp	File system path of template files for Oracle Internet Directory subscriber creation. By default, it is *ORACLE_HOME*/ldap/schema/oid/.

Table J–4 rmsub.csh

Parameter	Description
-name	Oracle Internet Directory nickname of an existing nondefault subscriber to be removed. This is a mandatory parameter. Default subscriber cannot be removed using this script, use the ptlconfig tool instead. See Section A.1, "Portal Dependency Settings Tool" for more information.
-pc	Database connect string for OracleAS Portal schema, in format of <host>:<port>:<sid>, where <host> is a fully qualified domain name. This is a mandatory parameter.

Table J–4 (Cont.) rmsub.csh

Parameter	Description
-pp	SYS user password of portal instance. By default, it is change_on_install.
-ps	OracleAS Portal schema name. By default, it is portal.
-sc	Database connect string for SSO schema, in format of <host>:<port>:<sid>, where <host> is a fully qualified domain name. By default, it is the value of -pc parameter.
-sp	SYS user password of OracleAS Single Sign-On instance. By default, if OracleAS Single Sign-On and OracleAS Portal are on different database instances, it is change_on_install; if OracleAS Single Sign-On and OracleAS Portal use the same database instance, it is the value of -pp parameter.
-ss	SSO schema name. By default, it is orasso.
-a	Portal Application name. By default, it is <portal_schema>.<sid>.<dbhost>.
-h	Oracle Internet Directory server host name. This is a mandatory parameter.
-p	Oracle Internet Directory server port number. By default, it is 389.
-d	Oracle Internet Directory bind DN. By default, it is cn=orcladmin. This DN must have Oracle Internet Directory admin privilege, for example, privilege to create new subscribers.
-w	Password for Oracle Internet Directory bind DN. By default, it is welcome1.
-cs	Commit size, specifying the number of rows that can be deleted before a mandatory database commit. By default, it is 1000.

Table J–5 syncasp.csh

Parameter	Description
-mode	This is a mandatory parameter. Valid values are: ■ pwd – Synchronize password for one ASP user. ■ dif – Synchronize ASP structure changes since last synchronization. ■ all – Do a complete synchronization of ASP structure.
-pc	Database connect string for OracleAS Portal schema, in format of <host>:<port>:<sid>, where <host> is a fully qualified domain name. This is a mandatory parameter.
-ps	OracleAS Portal schema name. By default, portal.
-pw	OracleAS Portal schema password. By default it is the value of -ps parameter.
-h	Oracle Internet Directory server host name. This is a mandatory parameter.
-p	Oracle Internet Directory server port number. By default, it is 389.

Table J–5 (Cont.) syncasp.csh

Parameter	Description
-d	Oracle Internet Directory bind DN. By default, it is cn=orcladmin. This DN must have Oracle Internet Directory admin privilege, for example, privilege to create new subscribers.
-w	Password for Oracle Internet Directory bind DN. By default, it is welcome1.
-u	This parameter is used with the password sync mode (pwd) to specify the user name whose password must be synchronized.
-l	Log file name.

Table J–6 embldip.csh

Parameter	Description
-pc	Database connect string for OracleAS Portal schema, in the format of <host>:<port>:<sid>, where <host> is a fully qualified domain name. This is a mandatory parameter.
-pp	SYS user password of portal instance. By default, it is change_on_install.
-ps	OracleAS Portal schema name. By default, it is portal.
-h	Oracle Internet Directory server host name. This is a mandatory parameter.
-p	Oracle Internet Directory server port number. By default, it is 389.
-d	Oracle Internet Directory bind DN. By default, it is cn=orcladmin. This DN must have Oracle Internet Directory admin privilege, for example, privilege to create new subscribers.
-w	Password for Oracle Internet Directory bind DN. By default, it is welcome.
-enable	Enables Directory Integration Platform on all subscribers in portal. This parameter precedes the -disable parameter.
-disable	Disables Directory Integration Platform on all subscribers in portal.

K

Troubleshooting OracleAS Portal

This appendix describes common problems that you may encounter when using Oracle Application Server Portal and explains how to solve them. It also gives detailed instructions on how to diagnose OracleAS Portal problems. It contains the following topics:

K.1 Problems and Solutions

This section describes common problems and solutions. It contains the following topics:

K.1.1 Unable to Access OracleAS Portal

You cannot access your portal instance. For example, pages are not displayed, you get an "HTTP 503 Service Unavailable" error, or an "An error occurred while processing the request. Try refreshing your browser. If the problem persists contact the site administrator" error when you try to access OracleAS Portal.

OracleAS Portal requires Oracle Application Server components, such as Oracle HTTP Server, OracleAS Web Cache, Portal Services, OracleAS Metadata Repository, and OC4J_Portal to be available (up) and running. One or more of these components may be unavailable (down).

Problem 1

Oracle HTTP Server is down.

Solution 1

Display the OracleAS Portal home page in Application Server Control Console. See Section 7.3, "Using Application Server Control Console to Monitor and Administer OracleAS Portal" for more information.

Check if Oracle HTTP Server is up. The Oracle HTTP Server status is displayed in the **Component Status** table on the OracleAS Portal home page.

- If the status is '**Up**', then continue to the next step.
- If the status is '**Down**', then start Oracle HTTP Server using Application Server Control Console.

To access Oracle HTTP Server monitoring and administration pages in Application Server Control Console, click **HTTP_Server** in either of the following:

- Portal Component Status table
- Application Server System Components table

If Oracle HTTP Server starts successfully, check whether your portal is accessible.

If Oracle HTTP Server fails to start, then investigate the Oracle HTTP Server error log files and try to determine the problem. See Section K.2.4, "Using Application Server Control Console Log Viewer" for more information. If you are not using Log Viewer, then check the relevant error log files in the following directories:

- *ORACLE_HOME*`/opmn/logs`
- *ORACLE_HOME*`/Apache/Apache/logs/error_log`

Problem 2

OracleAS Web Cache is down.

Solution 2

Navigate to the Oracle Enterprise Manager 10*g* Application Server Control Console of the Oracle home directory that is running the OracleAS Web Cache process. For details, refer to the section on Starting and Stopping Oracle Application Server Web Cache in *Oracle Enterprise Manager Advanced Configuration*.

Check if OracleAS Web Cache is up. The OracleAS Web Cache status is displayed in the Application Server System Components table.

- If the status is '**Up**', then continue to the next step.
- If the status is '**Down**', then start OracleAS Web Cache using the Application Server Control Console.

To access OracleAS Web Cache monitoring and administration pages in the Application Server Control Console, click **Web Cache** in the Application Server System Components table.

If OracleAS Web Cache starts successfully, check whether your portal is now accessible.

If OracleAS Web Cache fails to start, then investigate the OracleAS Web Cache error log files and try to determine the problem. See Section K.2.4, "Using Application Server Control Console Log Viewer" for more information.

If you are not using Log Viewer, then check the relevant error log files in the *ORACLE_HOME*`/opmn/logs` and *ORACLE_HOME*`/webcache/logs` directories.

Problem 3

OracleAS Portal is down due to incorrect Portal DAD configuration.

Solution 3

Check the status and configuration of the OracleAS Portal DAD. Navigate to the Oracle HTTP Server home page in Application Server Control Console. See Section 7.3, "Using Application Server Control Console to Monitor and Administer OracleAS Portal" for more information. Click **HTTP_Server** in either of the following:

- Portal Component Status table
- Application Server System Components table

From the Oracle HTTP Server home page, click **Administration**. and then **PL/SQL Properties**. Check the DADs table to see if the DAD configured for your portal is up.

- If the status is '**Up**', then continue to the next step.
- If the status is '**Down**', then click the name of the DAD in the DADs table and verify that all properties are set correctly. Save any changes and then restart both OC4J_Portal and Oracle HTTP Server for any change to take effect. See Section 4.5.3, "Configuring a Portal DAD" for information about configuring the DAD from the portal's home page.

> **Note:** You can verify the connection details for a DAD using SQL*Plus - in the Oracle home directory associated with the Oracle Application Server for your portal. The DAD Settings page displays the password in an encrypted form and forces you to reenter the password, to ensure that password validity is not the problem.

Check if your portal is accessible now.

Problem 4

OracleAS Metadata Repository is down.

Solution 4

Display the OracleAS Portal home page in the Application Server Control Console. See Section 7.3, "Using Application Server Control Console to Monitor and Administer OracleAS Portal" for more information.

Look under the **OracleAS Metadata Repository Used by Portal** section.

- If the status is '**Up**', then continue to the next step.
- If the status is '**Down**', then start the database.

If the database starts successfully, check whether your portal is accessible now.

Problem 5

The OC4J_Portal service is down.

Solution 5

Display the Application Server home page (for your OracleAS Portal instance), in the Application Server Control Console. See Section 7.2, "Using the Application Server Control Console" for more information.

The OC4J_Portal status is displayed in the System Components table.

- If the status is '**Up**', then continue to the next step.
- If the status is '**Down**', then start OC4J_Portal using the Application Server Control Console.

To access OC4J_Portal monitoring and administration pages in the Application Server Control Console, click **OC4J_Portal** in either of the following:

- Parallel Page Engine Services page (available from the Component Status table on the OracleAS Portal home page)
- Application Server System Components table

If OC4J_Portal starts successfully, check whether your portal is now accessible.

If OC4J_Portal fails to start, then investigate OC4J_Portal error log files and try to determine the problem. See Section K.2.4, "Using Application Server Control Console Log Viewer" for more information.

Problem 6

SQL* Net listener is down, or misconfigured.

Solution 6

Check that the SQL*Net TNS listener is up and running on the host where the metadata repository is installed. Log in to the computer containing the database, change to the *ORACLE_HOME*`/bin` directory if you are currently not in the `$PATH` directory, and use the following command to determine the status of the TNS listener:

```
lsnrctl status
```

If the service is not running, then start it by using the following command:

```
lsnrctl start
```

If the service is already up and running, then refer to the *Oracle Database Net Services Administrator's Guide* in the Oracle Database 10*g* documentation library, for information on how to troubleshoot Oracle Net Services.

Problem 7

OPMN or the Portal Services is reporting some other error.

Solution 7

Perform the following steps to resolve the problem:

1. Navigate to *ORACLE_HOME*/opmn/bin and issue the `opmnctl status` command. Ensure that key services show an **alive** status. If not, scan the files under *ORACLE_HOME*/opmn/logs for more details.
2. Navigate to *ORACLE_HOME*/j2ee/OC4J_Portal/application-deployments/portal/OC4J_Portal_default_island_1 and scan the file `application.log` for details on how to proceed with resolving the issue.
3. Navigate to *ORACLE_HOME*/webcache/logs. Scan the `event_log` file for any pointers to the problem.

K.1.2 Unable to Log In to OracleAS Portal

You can access the public home page but are unable to log in. Common symptoms of this problem are the following:

- The login page does not appear after you click **Login**.
- You get an error after you enter your credentials on the OracleAS Single Sign-On login page.
- You get errors on OracleAS Portal pages after you have been authenticated.

Problem 1

You may not be able to log in to OracleAS Portal due to problems encountered during the process of logging in to OracleAS Portal.

The OracleAS Portal login process can be logically broken down into three parts:

- Communication between OracleAS Portal and OracleAS Single Sign-On
- Communication between OracleAS Portal and Oracle Internet Directory
- Assignment of the Home Page

Solution 1

To help diagnose the cause of this problem, look at the solutions focused on each part of the login process.

Verify Communication Between OracleAS Portal and OracleAS Single Sign-On

To understand the first part of the login process, assume that OracleAS Portal is accessed at:

```
http://www.company.com/portal/pls/portal/
```

When you click **Login** on the public home page you get redirected to the OracleAS Single Sign-On page. For example, the URL changes to:

```
http://login.company.com:4443/pls/orasso
```

If you enter the user name and password provided by your administrator and click **Login**, then OracleAS Single Sign-On sends the user information back to OracleAS Portal.

To diagnose the cause of a problem encountered in this part of the login process, perform the following steps:

1. **Display the OracleAS Single Sign-On home page in the Oracle Enterprise Manager 10*g*** Application Server Control Console.

   The OracleAS Single Sign-On home page is available from the home page for the Infrastructure home directory instance.

   For details, refer to the section on Interpreting and Using the Home Page on the Standalone Console in the *Oracle Application Server Single Sign-On Administrator's Guide*.

2. **Check if Oracle HTTP Server is Up**.

   Click **HTTP_Server** displayed in the **Related Links** section.

   - If the status is '**Up**', then continue to the next step.
   - If the status is '**Down**', then start Oracle HTTP Server using the Application Server Control Console.

   To access Oracle HTTP Server monitoring and administration pages in the Application Server Control Console, click **HTTP_Server** in either of the following:

   - OracleAS Single Sign-On home page
   - Application Server System Components table

   If Oracle HTTP Server starts successfully, check whether you can log in now.

   If Oracle HTTP Server fails to start, then investigate the Oracle HTTP Server error log files and try to determine the problem. See Section K.2.4, "Using Application Server Control Console Log Viewer" for more information. If you are not using Log Viewer, then check the relevant error log files in the following directories:

   - *ORACLE_HOME*`/opmn/logs`
   - *ORACLE_HOME*`/Apache/Apache/logs/error_log`

3. **Check the status and configuration of the OracleAS Single Sign-On DAD**.

   From the OracleAS Single Sign-On home page, perform the following steps:

   a. In the Related Links section of the OracleAS Single Sign-On home page, click **HTTP_Server**.

   b. Click **Administration**.

   c. Click **PL/SQL Properties** and review the DADs section:

   - If the status is '**Up**', then continue to the next step.
   - If the status is '**Down**', then click the name of the DAD in the DAD table and verify that all properties are set correctly. Save any changes and restart Oracle HTTP Server and OC4J_Portal for any change to take effect.

   Check if you can log in now.

4. **Check if the database containing the OracleAS Single Sign-On schema is running**.

Database information is displayed on the OracleAS Single Sign-On home page in the Oracle Enterprise Manager 10*g* Application Server Control Console. Drill down for further information.

- If the status is '**Up**', then continue to the next step.
- If the status is '**Down**', then start the database.

If the database starts successfully, check whether your OracleAS Single Sign-On schema is accessible now.

5. **Check the status and configuration of the SQL* Net Listener**.

   Check that the SQL*Net TNS listener is up and running on the host where the Identity Management repository is installed. Log in to the computer containing the database. Change to the *ORACLE_HOME*`/bin` directory if you are currently not in the `$PATH` directory, and use the following to determine the status of the TNS listener:

   ```
   lsnrctl status
   ```

   If the service is not running, then start it by using the following:

   ```
   lsnrctl start
   ```

   If the service is already up and running, then refer to the *Oracle Database Net Services Administrator's Guide* in the Oracle Database 10*g* documentation library, for the specific error number returned, and then take suitable action.

6. **Check if the OC4J_Security service is up**.

   The OC4J_Security status is displayed in the Application Server Control Console.

   - If the status is '**Up**', then continue to the next step.
   - If the status is '**Down**', then start OC4J_Security using the Application Server Control Console.

   To access OC4J_Security monitoring and administration pages in the Application Server Control Console, click **OC4J_Security** in the System Components table on the home page for the Infrastructure home directory instance.

   If OC4J_Security starts successfully, then check if your OracleAS Single Sign-On schema is accessible.

   If OC4J_Security fails to start, then investigate OC4J_Security error log files and try to determine the problem. See Section K.2.4, "Using Application Server Control Console Log Viewer" for more information.

7. **Check the OracleAS Portal and OracleAS Single Sign-On connection configuration**.

   Check the connection configuration of OracleAS Portal and OracleAS Single Sign-On. There may be a problem with the connection configuration if you see any of the following error messages:

   - "`WWC-41453: The cookie version specified in the authentication message is not supported by this configuration". Please notify your administrator.`"
   - "`WWC-41454: The decryption of the authentication information was unsuccessful". This may be caused by corruption of the data, an incorrect encryption key in this application's configuration, or an illegal access attempt. Please notify your administrator.`

- "WWC-41439: You cannot login because there is no configuration information stored in the enabler configuration table."

If you find there is a problem with the OracleAS Portal and OracleAS Single Sign-On connection configuration, then fix the `Host` parameter in the `IASInstance` element and the `ListenPort` parameter in the `WebCacheComponent` element in the `iasconfig.xml` file and run the `ptlconfig` tool as follows:

```
ptlconfig -dad <dad name> -site
```

Verify Communication Between OracleAS Portal and Oracle Internet Directory

In the second part of the OracleAS Portal login process, the credentials provided by OracleAS Single Sign-On are used by OracleAS Portal to get Group membership information from Oracle Internet Directory.

To help diagnose the causes of problems encountered in this part of the login process, you need to check if the Oracle Internet Directory service is *Up*. The Oracle Internet Directory status is displayed on the Application Server page.

- If the status is '**Up**', then continue to the next step.
- If the status is '**Down**', then start Oracle Internet Directory using the Application Server Control Console.

To access Oracle Internet Directory monitoring and administration pages in the Application Server Control Console, click **OID** in the System Components table on the home page for the Infrastructure home directory instance.

If Oracle Internet Directory starts successfully, then check if you can log in.

If Oracle Internet Directory fails to start, then investigate the Oracle Internet Directory error log files and try to determine the problem. See Section K.2.4, "Using Application Server Control Console Log Viewer" for more information. It is also likely that this is a problem with OracleAS Portal and Oracle Internet Directory connection configuration. The solution is to fix the values in the `OIDComponent` element in the `iasconfig.xml` file and run the `ptlconfig` tool as follows:

```
ptlconfig -dad <dad> -oid
```

Verify Assignment of the Home Page

In the final part of the OracleAS Portal login process, you are redirected to the appropriate OracleAS Portal home page based on your Group membership. The home page preference can be specified at the System, Group, or User level.

If a home page has been specified for you, then it is displayed when you log in. If no home page has been specified for you, but you belong a default group, and a home page has been specified for your default group, then that page is displayed. If a home page has not been specified for you and you either do not belong to a default group or a home page has not been specified for the default group, then the System level default home page is displayed.

To help diagnose the cause of the problem, check if you have `View` privileges for the home page, at the User, Group, or System level.

When a home page is being displayed, you must have privilege to view the page. The privilege can be granted to one of the following:

- User
- User Group
- Public

If you do not have privileges to view the page at any of the levels in the preceding list, then you receive the "`WWC-44131: You do not have permission to perform this operation`" error.

At the Group or the System level, verify with an administrator that the group you are part of has correct privileges to view the page.

A portal administrator must perform the following steps to identify a user home page:

- Edit the user profile to find out the default home page and the default group for the user.
- If a default home page is already specified for the user, then stop here. Otherwise, edit the group profile for the default group, and check if a default home page is specified.
- If a default home page has been specified for the default group, then stop here. Otherwise check the default home page from the Global Settings page.

Once the home page is established, the next step is to find out about the privileges granted on the page. Edit the page, and click **Access**. Check whether or not the page can be viewed by public. In addition, look at the list of grantees. Check if the user or any group that the user belongs to has been given `View` or higher privileges on the page. Grant the appropriate privileges if needed. If the privilege has been granted to a group that the user is a member of, then ensure that the name of the user appears in the list of members.

Problem 2

OPMN is reporting issues.

Solution 2

Navigate to *`ORACLE_HOME`*`/opmn/bin` and issue the `opmnctl status` command. Ensure that key services show an **alive** status. If not, scan the files under *`ORACLE_HOME`*`/opmn/logs` for more details.

K.1.3 Problems with Oracle Application Server Integration Configuration

Oracle Enterprise Manager, authentication, caching, or URLs do not work with OracleAS Portal.

Problem

OracleAS Portal is not correctly configured to connect to certain other Oracle Application Server components. The `iasconfig.xml` file does not have the correct information for connecting to other Oracle Application Server components.

Solution

See Section K.2.6, "Verifying the Portal Dependency Settings File" for details on verifying the contents of the Portal Dependency Settings file, `iasconfig.xml`.

K.1.4 Problems Creating Category or Perspective Pages

When you create category or perspective pages, you may encounter the following errors:

- `WWS-32022:The category has been created but it was not possible to place the search portlets onto the category page. The category page will not show the items or pages in the category.`
- `WWS-32023:The perspective has been created but it was not possible to place the search portlets onto the perspective page. The perspective page will not show the items or pages in the perspective.`

Problem

When you create a category in a page group, a category page is created based on the category template. Similarly, when you create a perspective, a perspective page is created based on the perspective template. If changes are made to the underlying category or perspective templates, then you may see one of the preceding messages when you create a new category or perspective.

Solution

If either of these errors is displayed, you must first delete the current category or perspective template, and then run scripts to do the following:

- Replace the current category or perspective template with the original version.
- Re-create category or perspective pages that are based on the current template. You can do this either across all page groups, or for specific page groups.

This ensures that all new category or perspective pages are created without errors and that all existing category or perspective pages display their associated items and pages as expected.

See Section C.10, "Using the Category and Perspective Scripts" for more information about where to look for and run these scripts.

K.1.5 Problems with Network Address Translation (NAT) Setup

After following the steps in Section 5.3, "Configuring Multiple Middle Tiers with a Load Balancing Router", you encounter the following error:

```
Timeout occurred while retrieving page metadata
```

Problem

If a NAT bounceback rule is not correctly set up on the LBR when configuring multiple middle tiers, then the response to loopback requests is deleted, causing OracleAS Portal pages to time out.

Solution

NAT bounceback rule is set up differently on individual LBRs. Consult your LBR configuration guide for detailed information. Refer to Section 5.3, "Configuring Multiple Middle Tiers with a Load Balancing Router" for a detailed description on why the LBR needs additional configuration to make loopback communication successful.

K.1.6 User and Group Information in OracleAS Portal and Oracle Internet Directory Does Not Match

User and group information in OracleAS Portal is not synchronized with the information in Oracle Internet Directory.

Problem

Changes from Oracle Internet Directory are not propagated to OracleAS Portal. OracleAS Portal uses a provisioning profile to receive notifications when user or group privilege information changes. This enables OracleAS Portal to keep its authorization information synchronized with the information stored in Oracle Internet Directory. By default, this provisioning profile is enabled.

Solution

Perform the following steps to help diagnose the cause of this problem:

1. **Check if provisioning is enabled.**

   Perform the following steps to check if provisioning is enabled:

   a. Log in to OracleAS Portal. Click the **Administer** tab. On the **Portal Builder** page, click **Global Settings** under Services.

   b. Click the **SSO/OID** tab, and scroll down to the Directory Synchronization section. This section enables you to specify whether or not directory synchronization should be enabled. **Enable Directory Synchronization** should be selected, and by default, **Send event notifications every n seconds** must be set to 300.

   c. If the **Directory Synchronization** section is not visible or the check box for **Enable Directory Synchronization** is not checked, then the provisioning profile is not enabled. Enable the provisioning profile by selecting the **Enable Directory Synchronization**, check box and then clicking **OK** or **Apply**.

   If you encounter an error, you must re-create the provisioning profile. The solution is to run the `ptlconfig` tool as follows:

   ```
   ptlconfig -dad <dad> -dipreg
   ```

2. **Check if Oracle Directory Integration Platform is up and running.**

   Perform the following steps to do this:

   a. Display the Oracle Enterprise Manager 10*g* Application Server Control Console. See Section 7.2, "Using the Application Server Control Console" for more information. Navigate to the Application Server Control Console of the Infrastructure home directory associated with your portal.

   b. Oracle Internet Directory status is displayed on the Application Server page.

      – If the status is '**Up**', then continue to the next step.

      – If the status is '**Down**', then start Oracle Internet Directory using Application Server Control Console.

      To access the Oracle Internet Directory monitoring and administration pages in the Application Server Control Console, click **OID** in the Application Server System Components table.

      If Oracle Internet Directory starts successfully, then continue to the next step.

      If Oracle Internet Directory fails to start, then check the Oracle Internet Directory error log files and try to determine the problem. See Section K.2.4, "Using Application Server Control Console Log Viewer" for more information.

   c. Check if Oracle Directory Integration Platform is up.

      Click **OID** in the Application Server System Components table. On the page that follows, click **Directory Integration** in the Status section.

– If the status is '**Up**', then continue to the next step.

– If the status is '**Down**', then start the Oracle Directory Integration Platform server using Oracle Enterprise Manager 10*g* Application Server Control Console.

See Also:

- Managing the Oracle Directory Integration and Provisioning Server in the *Oracle Identity Management Integration Guide*
- Diagnosing Oracle Directory Integration and Provisioning Server Problems in the *Oracle Identity Management Integration Guide*

3. Check the contents of the trace and audit log files.

When changes are propagated, results are written to trace and audit files. Checking the contents of these files can give you additional information about propagation failures. Perform the following steps to check the contents of these files:

a. Log in to the computer that has the Oracle Directory Integration Platform server running.

Typically, the computer that has Oracle Internet Directory installed has the Oracle Directory Integration Platform server.

b. Check for the trace and audit Log Files.

Navigate to the *ORACLE_HOME*`/ldap/odi/log/` directory.

For each provisioning profile, there are two associated files in the log directory: `*.trc` (trace) and `*.aud` (audit) log files.

By default, the trace log file contains entries that are generated every 300 seconds, because the default value of **Send event notifications every n seconds** is `300`. This file contains the log of recent information logged by Oracle Directory Integration Platform and also contains any errors that may have been encountered. The trace log file gets recycled after some time.

The audit log file contains a history of all the changes that have been propagated to the provisioning profile. The following is an example of a message found in the audit log file:

```
Tue Jun 19 17:07:30 GMT 2004  -  Audit Log Start
--------------------------------------------------
Group Exists Check - DN : cn=super_users,cn=ASDB.COMPANY.US.COM,cn=Database
Instances,cn=UltraSearch,cn=Portal,cn=Products,cn=OracleContext,dc=us,dc=co
mpany,dc=com ,GUID (CE0D473B93B521FAE0340003BA109AC2) - Response :
=============Event ID : 2320 - (GROUP_MODIFY)=============
Source     : orclapplicationcommonname= ASDB.COMPANY.COM,cn=database
 instances,cn=ultrasearch,cn=portal,cn=products,cn=oraclecontext
Time       : 20031209170036z
Object Name: super_users
Object GUID: CE0D473B93B521FAE0340003BA109AC2
Object DN  : cn=super_users,cn= ASDB.COMPANY.COM,cn=Database
Instances,cn=UltraSearch,cn=Portal,cn=Products,cn=OracleContext,dc=us,dc=co
mpany,dc=com
AttrName     -      OpType     -     Value
-----------------------------------------
uniquemember    -     ADD    -
 cn=portal,cn=users,dc=us,dc=company,dc=com
EVENT_NTFY Response : 1
```

```
2320 : Success : 2 : cn=super_users,cn= ASDB.COMPANY.COM,cn=Database
Instances,cn=UltraSearch,cn=Portal,cn=Products,cn=OracleContext,dc=us,dc=co
mpany,dc=com
--------------------------------------------------
```

The propagation information gets written to the trace log files, and is periodically added to the audit log files. If changes are propagated properly, then the time stamp in the trace log file will be updated:

- If changes are propagated properly, but are not reflected in OracleAS Portal, then continue to the "Are Changes Propagated Properly?" section.
- If you do not find the trace and audit log files, then check if a provisioning profile exists by performing the following steps:

1. Log in to OracleAS Portal. Click the **Administer** tab. On the **Portal Builder** page, click **Global Settings** under Services.
2. Click the **SSO/OID** tab and scroll down to the Directory Synchronization section. This section lets you indicate whether or not directory synchronization is enabled. **Enable Directory Synchronization** must be selected, and **Send event notifications every [ ] seconds** must be set to 300 by default.
3. If these values are not set, then you must create a provisioning profile as detailed in the following section.

If you do not find the trace and audit log files in the *ORACLE_HOME*`/ldap/odi/log/` directory, then chances are that the provisioning profile has been deleted. To re-create a provisioning profile, run the `ptlconfig` tool as follows:

```
ptlconfig -dad <dad> -dipreg
```

Are Changes Propagated Properly?

To help diagnose whether or not changes are propagated properly, create, delete, and re-create a user through OracleAS Portal (in Oracle Internet Directory). To do this, perform the following steps:

1. Click the **Administer** tab.
2. Under User, click **Create New Users**.
3. Edit the profile of the user.
4. Delete the user.
5. Re-create the user with the same name.

Wait for the interval period (the time specified for **Send event notification** of messages). To minimize your wait time, you can reset this value to less than 300 seconds. After this, log in as the newly created user. If you receive the following error while logging in, then information is not propagating from Oracle Internet Directory to OracleAS Portal:

```
Error "WWC-41742: There is a conflict with your assigned user
name. There is a user entry with this name, but with a
different globally unique identifier (GUID), which must be
resolved before you can log on with this name. Please inform
your administrator."
```

If no information is propagated, then OracleAS Portal throws this error, because it has the same user name stored with a different GUID.

If changes are not propagated properly, then it is likely that there is a problem in either Oracle Directory Integration Platform or in the configuration of OracleAS Portal with Oracle Directory Integration Platform. If this is an OracleAS Portal configuration problem, then run the `ptlconfig` tool as follows:

```
ptlconfig -dad <dad> -dipreg
```

K.1.7 Problems with OracleAS Portal Performance

You may experience performance issues with OracleAS Portal for example, pages may load slowly.

There could be multiple reasons why your portal is slow. Some of these problems are described here.

Problem 1

Caching is disabled.

Solution 1

- Set the `PlsqlCacheEnable` parameter to `On` in the *`ORACLE_HOME`*`/Apache/modplsql/conf/cache.conf` file. See Section 4.5.4, "Configuring the Portal Cache" for more information.
- Ensure that the `PlsqlCacheDirectory` parameter is correctly configured in the *`ORACLE_HOME`*`/Apache/modplsql/conf/cache.conf` file. See Table 7–2, " Portal Cache Settings" for more information.

See Also: *Oracle HTTP Server Administrator's Guide*

Problem 2

Page metadata is not cached in OracleAS Web Cache. The initial, one-time call from the middle tier to the OracleAS Portal schema to determine the OracleAS Portal version may have failed.

Solution 2

To resolve this problem, perform the following tasks:

1. Confirm that page metadata is not cached in OracleAS Web Cache. To do this, perform either of the following steps:
   - Append a page URL with `&_debug=1`, refresh the browser, and verify that the OracleAS Web Cache page metadata cache status is `MISS, NON-CACHEABLE`.
   - Access the Popular Requests screen in OracleAS Web Cache Manager and verify that page metadata is not cached.
2. Restart all OC4J_Portal instances.
3. Perform Step 1 again to determine if page metadata is now cached in OracleAS Web Cache.

If the caching problem has not been resolved, then perform the diagnostic steps described in the *Oracle Application Server Portal Error Messages Guide* for the following error:

```
WWC-40018: General invalidation message processing exception: %1
```

Problem 3

Oracle Containers for J2EE (OC4J) is unable to handle the load.

Solution 3

If you get repeated error messages such as the following, then it is possible that the load on by OC4J_Portal exceeds the capacity of either your hardware or configuration settings:

```
application.log file  contains [example]
04/28/04 5:47 PM portal: Fetcher content-fetcher12 shut down.
04/28/04 5:47 PM portal: UncaughtException in thread name=content-fetcher12,
java.lang.RuntimeException: Fetcher
at oracle.webdb.page.ContentFetcher.run(Unknown Source)
```

Upgrade your hardware to handle a heavy load by doing any or all of the following:

- **Adding hardware or an LBR**: For details, refer to the white paper titled "*How to Effectively Size Hardware for Your Portal Implementation*" located on Oracle Technology Network (OTN) at: http://www.oracle.com/technology/products/ias/portal/administration_10g1014.html.
- **Removing `-xingc`**: This is a default setting in Oracle9*i*AS Portal (9.0.2), required for the Wireless element of OC4J_Portal. `-xingc` is the flag that turns on incremental garbage collection for the JVM. Garbage collection in OracleAS Portal occurs when maximum heap size (-mx) is reached, and does not necessarily require `-xingc`. By removing `-xingc` , you will not see periodic CPU spikes that occur during the garbage collection process. The CPU spiking for incremental garbage collection can be a problem on a CPU-bound system. If `-xingc` is removed, then garbage collection will occur only when that `-mx` is reached.
- **Increasing `-ms` and `-mx`**: These are initial (`ms`) and maximum (`mx`) heap sizes specified for each instance of OC4J_Portal. These are usually specified in Megabits.
- **Increasing `numProcs`**: `numProcs` is the number of default instances of an OC4J that will be instantiated at startup. The default value of `numProcs` is `1`. Refer to subsection "Option 1: Create a New OC4J Instance to Create Another Set of PPE Threads" under Section 9.3, "Setting the Number of PPE Fetchers", for information about increasing the number of OC4J_Portal instances. Before OracleAS Portal (9.0.4), this could also be set by editing the application configuration file where the default values of the `numProcs` setting are stored.

  The `numprocs` setting provides an element of scalability and redundancy. The Parallel Page Engine (PPE) has an internal concept of fetcher threads, which are used to respond to requests for content. By default, each PPE has 25 fetcher threads available for use. When a thread is busy, the available number is reduced. At a high load when no threads are available, the incoming requests are queued. This can be alleviated by increasing the fetcher threads number in the PPE configuration file. While this provides you with an increased number of available threads, there is no redundancy.

  By setting `numProcs` to `2`, you have two instances of OC4J_Portal and therefore twice the number of fetcher threads, that is 50. If any instance should fail for any reason, then you still have 25 threads while OPMN restarts the instance that has died.

Problem 4

Low or no reuse of connection pool.

Solution 4

Set the `PlsqlMaxRequestsPerSession` parameter to `1000` in the `dads.conf` file.

- On UNIX systems, this file is in the *`ORACLE_HOME`*`/Apache/modplsql/conf` directory.
- On Windows systems, this file is in the *`ORACLE_HOME`*`\Apache\modplsql\conf` directory.

Ensure that the `PlsqlMaxRequestsPerSession` parameter is *not* set to `1`. Doing this disables connection pooling. For information about the `PlsqlMaxRequestsPerSession` parameter, refer to the *Oracle HTTP Server Administrator's Guide*.

Note: It is recommended that you edit the `dads.conf` file using Application Server Control Console.

If you manually edit the `dads.conf` file, then you must add the necessary mod_rewrite and mod_oc4j directives to the `httpd.conf` and `mod_oc4j.conf` files respectively. To do this, perform the steps mentioned in Section E.2, "DAD Configuration File (dads.conf)" using the Application Server Control Console.

Problem 5

Processes start up or shut down frequently.

Solution 5

Tune the Oracle HTTP Server parameters, `MaxSpareServers` and `MinSpareServers`, in the *`ORACLE_HOME`*`/Apache/Apache/conf/httpd.conf` file. The default value for the `MaxSpareServers` parameter is `10`, and the default value for the `MinSpareServers` parameter is `5`.

Note: Tuning of both the `MaxSpareServers` and `MinSpareServers` parameters should be necessary only on very busy sites. Setting this parameter to a large number is a bad idea. For more information refer to the *Oracle HTTP Server Administrator's Guide*.

Problem 6

Connection pool is cleaned up too frequently.

Solution 6

Tune the `PlsqlIdleSessionCleanupInterval` parameter in the *`ORACLE_HOME`*`/Apache/modplsql/conf/plsql.conf` file. Increasing the value of this parameter allows pooled database connections to remain available in the pool for the specified time. The default value is `15` (minutes).

Problem 7

Disk input or output not distributed.

Solution 7

Many components, such as the following, access disks all the time:

- Oracle HTTP Server access and error logs

- Portal cached content
- Web content service
- Other local applications

All these components compete for the resources of the file system. To reduce input or output bottlenecks, ensure that you have a good distribution across physical disks.

Problem 8

Too many network hops. Typical problems can be any of the following:

- PPE loopbacks are not configured on clustered Oracle HTTP Server environments.
- Servlet engines (Oracle Containers for J2EE) run on a computer other than Oracle HTTP Server or mod_oc4j.
- Infrastructure components run across wide networks with multiple routers.

Solution 8

Try to reduce the number of network hops by avoiding or working around the listed problems.

Problem 9

Use of the HTTPS protocol for serving content.

You may have configured your portal to use HTTPS for ordinary content that does not need to be secure.

Solution 9

Avoid the unnecessary use of HTTPS. HTTP works well in most cases. If you really need a secure environment, then use reverse proxy hardware that will manage HTTPS and Secure Socket Layer (SSL). See Section 5.6, "Configuring Reverse Proxy Servers" for more information.

See Also:

- Section 6.3.2.1, "Configuring SSL for OracleAS Portal"
- *Oracle Application Server Enterprise Deployment Guide*

Problem 10

After performing all the tasks in the solutions provided, OracleAS Portal is still slow.

Solution 10

- **Review metric information for OracleAS Portal, its host, and other relevant components.**

  If all the components required by OracleAS Portal are up and running as expected, then the next step is to review metric information in Oracle Enterprise Manager 10*g* Grid Control Console, or Application Server Control Console.Reviewing this information can help you identify the problem.

  Click **All Metrics** in the portal home page to review metric information. Repeat this on home pages for other relevant components (OracleAS Web Cache, Oracle HTTP Server, OC4J, and so on).

- **Run OracleAS Portal Diagnostics Assistant.**

You can diagnose portal-related issues by reviewing the report generated by using OracleAS Portal Diagnostics Assistant. See Section K.2.5, "Using OracleAS Portal Diagnostics Assistant" for more information.

See Also:

- For more information, refer to the Performance page on the OracleAS Portal section of OTN, `http://www.oracle.com/technology/products/ias/portal/performance_10g1014.html`
- The Performance Monitoring Scripts zip file on OTN, `http://www.oracle.com/technology/products/ias/portal/files/portal_performance.zip`
- *Oracle Application Server Performance Guide*
- *Oracle HTTP Server Administrator's Guide*

Note: The Performance Monitoring Scripts zip file is also available as part of the Oracle Application Server installation.

K.1.8 Error When Creating Web Folders

When you try to create Web folders in OracleAS Portal, you get an ORA-20504 error, in the Web server error log file.

Problem

The `wwdav$path` and `wwdav$asl` tables are corrupt.

Solution

To repopulate the tables, you have to run the DAV Loader (`wwdav_loader`) utility. You can run the DAV Loader utility by executing the following procedure from SQL*Plus:

```
set serveroutput on size 1000000
begin
    wwdav_loader.create_dav_content;
end;
```

This re-creates all the DAV data. To get more debugging information, you can also use:

```
set serveroutput on size 1000000
begin
    wwdav_loader.create_dav_content(
        p_debug_mode => true);
end;
```

Running the DAV Loader removes any temporary documents, and any locks on documents, from the DAV tables. Items submitted for approval no longer appear in the DAV Loader until they are accepted or rejected.

In future, to examine whether there are any data corruptions in the DAV schema, you can run the DAV Report utility. To run this utility, perform the following steps:

1. Change the directory to *ORACLE_HOME*`/portal/admin/plsql/wws`, where the `davreprt.sql` file is available.

2. Log in to SQL*Plus as the PORTAL schema user.
3. Run the DAV Report Utility as follows:

```
davreprt.sql
```

This will run through a series of tests. If all tests pass, then no known data corruptions can be found in the DAV schema. If any test fails, then the DAV Loader must be run to correct the data corruption.

K.1.9 Create New Users and Create New Groups Portlets Do Not Appear

The Create New Users and Create New Groups portlets are displayed based on user privileges. The portlets may not appear for a variety of reasons.

Problem 1

You do not have sufficient privileges.

Solution 1

Use the Delegated Administration Service Self-Service Console to verify if you can administer users and groups. If you do not have the required privileges, then request the administrator to grant you the required privileges. However, if you can successfully perform these operations from the Self-Service Console, then it is most likely related to the next two problems. Inform the administrator about the issue.

Problem 2

Oracle Internet Directory is down or the group information in Oracle Internet Directory is incorrect.

Solution 2

If the Group membership information in Oracle Internet Directory is incorrect or if Oracle Internet Directory is not up and running, then perform the following steps to help diagnose the cause of this problem:

1. Display the portal home page in Oracle Enterprise Manager 10*g* Application Server Control Console. See Section 7.2, "Using the Application Server Control Console" for more information. Navigate to Application Server Control Console of the Infrastructure home directory associated with your portal.

   Check if Oracle Internet Directory is up. Oracle Internet Directory status is displayed in the application Server page.

   - If the status is '**Up**', then continue to the next step.
   - If the status is '**Down**', then start Oracle Internet Directory using Application Server Control Console or the command line.

     To access Oracle Internet Directory monitoring and administration pages in Application Server Control Console, click **Oracle Internet Directory** in the Application Server System Components table.

     If Oracle Internet Directory starts successfully, then check if the Create New Users and Create New Groups portlets are displayed.

     If Oracle Internet Directory fails to start, then investigate Oracle Internet Directory error log files and try to determine the problem. See Section K.2.4, "Using Application Server Control Console Log Viewer" for more information.

Problem 3

OracleAS Portal and Oracle Internet Directory connection configuration is incorrect.

Solution 3

Fix the values in the `OIDComponent` element in the `iasconfig.xml` file and use the `ptlconfig` tool to reconfigure your Oracle Internet Directory and OracleAS Portal configuration as follows:

```
ptlconfig -dad <dad> -oid
```

Note: When you run the `ptlconfig -dad <dad> -oid` command, it configures Oracle Internet Directory with OracleAS Portal with the following types of seeded entries:

- Application Entry
- Group Container Entry
- Groups in the Container
- User Entries
- Membership of Privileged Groups

If you have previously removed any of these seeded entries, then you must remember that they will be re-created when you reconfigure OracleAS Portal with Oracle Internet Directory.

K.1.10 ORA-2000x Errors in the error_log File

When you attempt to log in to OracleAS Portal, you see one of the following errors in the error_log file located in the *MID_TIER_ORACLE_HOME*`/Apache/Apache/logs` directory:

- **ORA-20000**: "An attempt was made to access the session context without a valid session"

  This error denotes that the OracleAS Portal session associated with that particular browser session is broken or lost, or that the session cookie itself is missing.

- **ORA-20001**: "The session cookie is corrupt - unable to obtain session information. Please close your browser and reconnect."

  This error indicates a corrupt or otherwise invalid session cookie.

- **ORA-20005**: "The session context could not be restored because the session is marked as inactive"

  This error is raised when the session cookie points to an inactive session. The cookie sent by the browser matches the cookie stored in the session, but the session is not active.

- **ORA-20006**: "The session context could not be restored because the cookie value does not match the value stored in the session repository"

  This error indicates that there is a mismatch between the cookie sent by the browser and the cookie stored in the session.

Note: It is important to understand that some of these errors are expected. Report the problem to Oracle Support Services only if an unusual number of exceptions is encountered. See Section K.3, "Need More Help?"for more information.

These errors are discussed in detail in the following subsections.

ORA-20000 "An attempt was made to access the session context without a valid session"

This error can be caused by any of the following problems:

Session Row Is Missing

Each session cookie has a corresponding session stored in the portal schema that contains information about the session cookie corresponding session. The data stored in the session includes the session ID, user name, session start time, information about whether the user is logged on or not, what time the user logged on, whether the session is active or marked for cleanup. The ORA-20000 error is raised if the session ID specified in the cookie does not exist in the sessions stored in the portal schema in the OracleAS Metadata Repository.

Session Is Cleaned Up

A background job runs frequently to clean up old sessions from the portal schema in the OracleAS Metadata Repository. By default, this job is configured to clean up sessions that are older than seven days. An attempt to access a session that has been cleaned up by the background job will result in an ORA-20000 error. See Section C.6, "Managing the Session Cleanup Job" for details.

Session Cookie Is Missing

If more than one DAD is configured for use with the portal schema in an OracleAS Metadata Repository, and the cookie name specified in these DADs is not the same, then it will result in the `cookie_name` value in the `wwctx_cookie_info$` table switching values every time a new session creation request is received through one of the DADs. This will result in an ORA-20000 error.

ORA-20001 "The session cookie is corrupt - unable to obtain session information. Please close your browser and reconnect."

The ORA-20001 error can be caused by any of the following problems:

Cookie Is Truncated

If you click **Stop** on the browser during the transmission of a request, then the cookie may be truncated. The next time you access the browser, the server is unable to properly decrypt the cookie.

Cookie from Another Server Is Received

If you have recently accessed another OracleAS Portal that is configured with a domainwide cookie scope, then an ORA-20001 error may be raised. If the cookie name of that portal is the same as your portal cookie name, then OracleAS Portal tries to use that cookie. However, because each portal cookie is encrypted using portal-specific keys, OracleAS Portal will not be able to decrypt the cookie, and will raise an ORA-20001 error assuming that the cookie is corrupted.

To avoid this namespace collision issue, you must determine the source of these cookies. Closing the browser will clear all the session cookies. You can debug the problem by starting up the browser with cookie warnings turned on, to see where the cookies are obtained from.

Cookie Encryption Key Is Changed

The cookies are encrypted using **DES3 encryption**. The encryption key is stored in the portal schema in the OracleAS Metadata Repository. Its value is typically set during OracleAS Portal installation and does not change thereafter. If this value is changed after installation, then it is not possible to decrypt any of the outstanding session cookies. Also, any other values that have been encrypted with this key cannot be decrypted. Note that this value should not be changed.

ORA-20005 "The session context could not be restored because the session is marked as inactive."

The ORA-20005 error results when the session cookie points to an inactive session. The cookie sent by the browser matches the cookie stored in the session, but the session is not active. It indicates that you made a logout request, but another request (for example, a user makes a request to change the language from the Language portlet) was sent before the cookie was reset in the browser.

Session Is Marked Inactive

When the user logs out, it is possible that the session stored in the portal schema gets updated to an inactive state. However, if you click **Stop** in the browser, then the cookie does not get cleared from the users browser. If this happens, then the browser sends the old cookie, causing OracleAS Portal to try to locate the inactive session. When this happens, an ORA-20000 error is raised.

ORA-20006 "The session context could not be restored because the cookie value does not match the value stored in the session repository."

The ORA-20006 error indicates that there is a mismatch between the cookie sent by the browser and the cookie that is stored in the session. This could happen if the cookie changes based on one request, and the user sends another request before the cookie is actually updated in the browser. For example, the user makes a request to change the language from the Language portlet, but sends another request before the first request is complete. This is similar to the ORA-20005 error, with the difference that the cookie itself contains a mismatch between the client and the server.

Time Stamp Does Not Match

The cookie may be decrypted properly, but if the time stamp in the cookie does not match the time stamp in the associated session row, it is considered to be corrupt. This mismatch in time stamp may occur if the user invokes the login twice, if there are network configuration issues, or bugs in the session creation logic, or because of a malicious session attack.

K.1.11 Remote Web Providers Time Out in a Dynamic DNS Environment

A remote Web provider that is located on a computer different from the OracleAS Portal middle tier, works when the OC4J_Portal service is first started, but stops working after some time. After a long timeout interval, the `Error: the portlet could not be contacted` message is shown in the place of each portlet from the same provider. Portlet timeout interval errors are also found in the OC4J_Portal `application.log` file. After restarting OC4J_Portal, the Web provider works again, but only for a limited period of time.

Problem

The possible cause for this problem can be that the Web provider is using dynamic DNS (DDNS) for its *Domain Name to IP Address* mapping. This means that the IP address that the Web provider domain name resolves to changes over time. Java default caching policy caches IP addresses forever, once it has resolved them. This means the Java cache stores an outdated IP address of the Web provider if the IP address of the Web provider changes, because of DDNS.

Solution

To resolve this problem, you need to perform additional configuration in OC4J_Portal to prevent remote Web providers from timing out. You must change the `sun.net.inetaddr.ttl` system property for OC4J_Portal. On JDK 1.3 and later, you can use the `sun.net.inetaddr.ttl` system property to specify the "*time to live*" (TTL) in seconds for cached IP addresses.

> **Note:** This system property is passed as a command-line option to Oracle Containers for J2EE (OC4J). Setting the property in the `oc4j.properties` file does not help, because the system property is read first before OC4J reads this file. Therefore, it is best to modify the `<java-option>` line in the OC4J_Portal section of the *ORACLE_HOME*`/opmn/conf/opmn.xml` directory.

Example

1. Edit the `opmn.xml` file as follows:

```
<java-option value="-server -Xincgc -Xnoclassgc -Xms256m -Xmx512m
-Dsun.net.inetaddr.ttl=120"/>
```

2. Shut down `opmn` and all its subprocesses, and restart it for the latest configuration changes to take effect.

   To do this, run the following commands:

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

K.1.12 Problems Related to Memory-Intense Operations

You see the error "`ORA-04031: unable to allocate 30192 bytes of shared memory.`"

Problem

By default, the `shared_pool_size` value in Oracle Application Server is 32 megabytes. This can cause problems if you are performing memory-intense operations such as the following:

- Export or Import
- Creating Portal Forms or Reports

Solution

To facilitate memory-intense operations, you must increase the value of the `shared_pool_size` parameter.

If you are unfamiliar with the steps involved in updating a database initialization parameter, refer to the section, "Managing Initialization Parameters Using a Server

Parameter File", in the *Oracle Database Administrator's Guide* in the Oracle Database 10*g* documentation library.

> **Note:** As an optional step, it is suggested that you run OracleAS Portal Diagnostics Assistant to view a report of the existing and recommended values of the database. See "Running OracleAS Portal Diagnostics Assistant" for details.

K.1.13 Problems with Oracle Text Installation

You are facing issues with the installation of Oracle Text.

Problem

You encounter Oracle Text-related problems.

Solution

Use the `TEXTTEST` utility to check that Oracle Text is installed and set up correctly. See Appendix H, "Using TEXTTEST to Check Oracle Text Installation" for more information.

K.1.14 Unable to Create Oracle Text Indexes

When trying to create Oracle Text indexes, you may encounter the following errors:

- "Cannot grant CTXAPP role to portal"
- "ERROR: Creating data store procedures in CTXSYS"
- "ERROR: Setting up Oracle Text data stores"
- "An unexpected error has occurred (WWS-32100)"

Problem

You face problems when trying to create Oracle Text indexes.

Solution

Oracle Text must be installed in the same Oracle Database home directory as the portal schema. See Section 8.3.2, "Oracle Text Prerequisites" for details.

Choose one of the following options to resolve this issue:

- Access the database and log in as the OracleAS Portal schema owner. Start SQL*Plus and run the `inctxgrn.sql` script. This script is located in the *`ORACLE_HOME`*`\portal\admin\plsql\wws` directory. Running this script creates the Oracle Text data store procedures and also grants the `CTXAPP` role to the OracleAS Portal schema.
- If you have access to the database, but you do not have a copy of the `inctxgrn.sql` script, use SQL*Plus to connect to the database as the schema owner and run the following commands:

```
set serveroutput on size 10000
begin
   wwv_context_util.grantCtxRole(user);
end;
@@sbrimtlx
```

Replace `(user)` with the OracleAS Portal schema owner, for example, `portal`.

Refer to Appendix H, "Using TEXTTEST to Check Oracle Text Installation" for more information.

K.1.15 Problems with MultiLanguage Support for Help

Only a subset of the online Help appears to be translated in OracleAS Portal.

Problem

In OracleAS Portal, there is multi-language support for the online Help. However, only a subset of the context-sensitive Help topics is translated for languages other than Japanese.

Solution

This is expected activity.

K.1.16 Stale Style-Sheet Data Is Displayed on Portal Pages

When editing style sheets, you see stale style-sheet data when previewing or viewing the style sheet in the context of a portal page.

Problem

Changes to style sheets are not reflected on portal pages. This is because the Greenwich Meridian Time (GMT) is appended to the numeric value, which generates the Last-Modified header without correcting the time zone. If the time zone of the original server precedes GMT, then the generated Last-Modified header is actually a future date.

Solution

Perform the diagnostic steps described in the *Oracle Application Server Portal Error Messages Guide* for the following errors:

```
WWC-40018: General invalidation message processing exception: %1
WWC-40019: Could not open web cache connection
```

If the problem is not resolved by these steps, then verify that the date, time, and time zone have been set to the current values on the OracleAS Web Cache hosts and the database host. Also, verify that the database time zone has been set to match the database host time zone. The database time zone can be determined by executing the following query:

```
SQL> SELECT DBTIMEZONE FROM DUAL;
```

If the database time zone differs from the database host time zone, then set the database time zone to the database host time zone using the `ALTER DATABASE SET TIME_ZONE` command, and then restart the database.

For example:

```
SQL> ALTER DATABASE SET TIME_ZONE = '-05:00';
```

The change will not take effect until the database is restarted.

K.1.17 Stale Content Is Displayed on Portal Pages

The content on your portal pages is not getting refreshed and stale content is displayed.

Problem

Your browser cache settings may be incorrect.

Solution

Ensure that the browser cache setting is *not* set to **Never**.

To verify this setting, refer to the "Browser Recommendations" section in the Preface of the *Oracle Application Server Portal User's Guide*.

K.1.18 Images Are Not Displayed on Portal Pages

When using OracleAS Portal, you may face either of the following problems:

- Images are not displayed.
- After logging out of portal, you cannot log in unless you close the browser and open it again.

Problem

Your browser image settings may be incorrect.

Solution

Ensure that images are automatically loaded. To verify this setting, refer to the "Browser Recommendations" section in the Preface of the *Oracle Application Server Portal User's Guide*.

> **Note:** It is recommended that this setting is always enabled.

K.1.19 Unhandled Exception Errors

When accessing or using OracleAS Portal, you may encounter an unhandled exception error. For example, "Error 30526: An Unhandled Exception has occurred."

Problem

OracleAS Portal encounters a database error from which it cannot recover.

Solution

In case of unhandled exception errors, the actual cause of the error is not clear. To gather more information about the possible cause of the error, generate a trace file.

After you have turned tracing on, you can find the generated trace files in the directory specified in the database parameter `user_dump_dest`. To find out the name of the directory, use either of the following commands:

```
select value from v$parameter where name = 'user_dump_dest';

show parameter user_dump_dest;
```

Refer to Section K.2.2, "Generating Trace Files" for the procedure to generate trace files. These trace files are not formatted. Use the `tkprof` utility to format them.

K.1.20 Problems in Configuring the OmniPortlet Provider

When configuring the OmniPortlet provider to build portlets, you may encounter a number of problems. To resolve many of the problems, you may need to view the

OmniPortlet provider application log file, `application.log`. This log file is available at either of the following locations:

- *OC4J_HOME*`/j2ee/home/application-deployments/portalTools/` (for PDK Only installations)
- *ORACLE_HOME*`/j2ee/OC4J_Portal/application-deployments/portalTools/OC4J_Portal_default_island_1/`

Problem

The required SSL library is not in the library path.

If you installed the OracleAS PDK on a standalone OC4J instance, or if you downloaded the preconfigured standalone OC4J with OracleAS PDK, then you may encounter the following error:

```
"java.lang.NoClassDefFoundError: at
oracle.security.ssl.OracleSSLCipherSuite.isSSLLibDomestic when accessing HTTPS
site with certificate"
```

Solution

Ensure that the SSL library is in the library path. Refer to "Copying the Library for HTTPS Access (PDK Only)" for more information.

K.1.21 Problems in Configuring OracleAS Web Cache for the OmniPortlet Provider

Stale portlet content displays on the portal page, and it does not reflect the portlet definition. This problem may occur because of the following errors in OracleAS Web Cache configuration:

Problem 1

The port value is not specified properly in the `cache.xml` file. You may encounter the following error:

```
CONFIGURATION: Encountered a Cache Invalidation Exception.
oracle.net.http.HttpConfigurationException: Bad "port" value in configuration
element "invalidation"
```

Solution 1

Set the correct port number in the `cache.xml` file.

A template copy of the `cache.xml` file can be found in the *ORACLE_HOME*`/portal/conf` directory. To specify the port, modify the configuration file as indicated by the italicized entry in the following example:

```
<?xml version="1.0"?>
<webcache>
    <invalidation
        host="cache.us.oracle.com"
        port="4001"
authorization="0510198d5df8efd5779406342be2528aa0cccb179ea6b77baf49f019f5075a3a11"
/>
</webcache>
```

Problem 2

The authorization value is not encrypted in the `cache.xml` file. You may encounter the following error:

```
CONFIGURATION: Encountered a Cache Invalidation Exception.
oracle.net.http.HttpConfigurationException: Bad "authorization" value in
configuration element "invalidation." String un-obfuscation error
```

Solution 2

Information about the OracleAS Web Cache instance is maintained in the `cache.xml` file in the *ORACLE_HOME*`/portal/conf` directory. If the Web Cache invalidation settings change, then you must update this file. Refer to Section I.2.1.3, "Configuring Caching (PDK Only)" for more information.

Problem 3

The `oracle.http.configfile` system property is not defined. This means that the configuration file for Web Cache Invalidation is not defined. You may encounter the following error when you start the OC4J instance:

```
Error: CONFIGURATION: Provider Test Page: Web Cache Invalidation config file not
defined by "oracle.http.configfile"
```

Solution 3

Add the `oracle.http.configfile` system property as a new line in the *ORACLE_HOME*`/j2ee/OC4J_Portal/config/oc4j.properties` file as shown in the following example:

`oracle.http.configfile=<`*fully_qualified_filename*`>`

Problem 4

The configuration file defined by the `oracle.http.configfile` system property does not exist. You may encounter the following error:

```
Error: CONFIGURATION: Provider Test Page: Web Cache Invalidation config file
defined by "oracle.http.configfile" does not exist.
```

Solution 4

Ensure that you have specified a valid file name in the `oracle.http.configfile` system property in the *ORACLE_HOME*`/j2ee/OC4J_Portal/config/oc4j.properties` file.

K.1.22 Problems in Accessing OracleAS Portal from a Mobile Device

Mobile devices do not provide good interfaces for displaying detailed error information when compared with standard desktop browsers. Because of this, a lot of error information is logged in the Oracle Application Server Wireless log file. You can access this log file using a Web-based monitoring tool known as the Activity Logger. Refer to the *Oracle Application Server Wireless Administrator's Guide* for details about using the Activity Logger.

When you access OracleAS Portal through OracleAS Wireless, you may encounter either of the following errors:

- Service Error
- Temporary Error

Service Error A service error is generated by the OracleAS Wireless server when the wireless server has a problem accessing the back-end server. A service error is displayed as follows:

```
Service Error
```

A service error may be generated for any of the following reasons:

- A document that is not of `text` or `vnd.oracle.mobilexml` type has been returned to the OracleAS Wireless server.
- A document of `text` or `vnd.oracle.mobilexml` type has been returned to the OracleAS Wireless server, but the content is not valid XML.
- A document of `text` or `vnd.oracle.mobilexml` type has been returned to the OracleAS Wireless server, but the content is not valid OracleAS Wireless XML.
- An error status has been returned to the OracleAS Wireless server, but there is no attached document that can be returned to the user.

Temporary Error A temporary error is a message generated by Parallel Page Engine (PPE) if there is a problem in rendering error documents for a mobile device. A temporary error is displayed as follows:

```
A temporary error has prevented Oracle Portal from servicing your request.
(id=<nnnnn>)
```

The value <*nnnn*> is the log error ID.

When rendering error documents for standard desktop browsers, PPE takes the error document that resulted from the metadata call to the database, and passes it to the user. This cannot be done for mobile devices because the documents rendered for mobile requests must be in OracleAS Wireless XML.

If PPE is servicing a mobile request and the database renders an error document that is not valid OracleAS Wireless XML, then PPE performs the following tasks:

1. Writes the document into the servlet error log file, `application.log`, located in the *`ORACLE_HOME`*`/j2ee/OC4J_Portal/application-deployments/portal/OC4J_Portal_default_island_1` directory.
2. Assigns a unique ID to the error.
3. Passes a standard error template to the user in the following format:

   ```
   A temporary error has prevented Oracle Portal from servicing your request.
   (id=<nnnnn>)
   ```

   where <*nnnn*> is the log error ID.

Problems and Workarounds for Service and Temporary Errors The different problems and workarounds for service errors and temporary errors are discussed in the following subsections.

> **Note:** After performing a suggested workaround, clear the cache, close the browser, and open it again. This must be done because the error page may be cached and you may encounter the service error again when you try to access the back-end service.

Problem 1

Oracle Application Server is not configured correctly. You encounter a service error.

Solution 1

Verify the OracleAS Portal and OracleAS Wireless configurations in Oracle Application Server. For details about verifying these settings, refer to the *Oracle Application Server Administrator's Guide*.

Problem 2

You are not authenticated to access OracleAS Portal from a mobile device or simulator. This could be because the comparison of IP addresses failed when validating the session cookie during logon. You encounter a service error.

Solution 2

Change the state of IP checking in cookie validation. Refer to Section C.2, "Configuring for IP Check During Session Cookie Validation" for details.

Access OracleAS Portal from a mobile device or simulator and check if you still get a service error.

Problem 3

The `xml.validation.mode` parameter is set to `True`. If this parameter is set to `True`, then OracleAS Wireless tries to validate the error message file, which is not in valid XML format. You encounter a service error.

Solution 3

In the OracleAS Wireless instance, ensure that the `xml.validation.mode` parameter is set to `False` in the `web.xml` file located at:

ORACLE_HOME`/j2ee/OC4J_Wireless/applications/wdk/wdk-web/WEB-INF`

Access OracleAS Portal from a mobile device or simulator and check if you still get a service error.

Problem 4

There are changes in OracleAS Wireless actions. You encounter a service error.

Solution 4

Check if any service can be run on the OracleAS Wireless server. For example, check if you can run OracleAS Wireless examples from a mobile device or simulator. For details, refer to the *Oracle Application Server Wireless Administrator's Guide*.

- If the example services do not work, then you may have to install and configure OracleAS Wireless again. For details, refer to OracleAS Wireless documentation.
- If the example services work properly, then check the OracleAS Wireless server log file for troubleshooting information. The log file stores information about the response from the portal service that is causing problems. Using this information, you can check if portal is returning an error status or invalid OracleAS Wireless XML.

  Access the OracleAS Wireless server log file by clicking **View Log** at the bottom of the OracleAS Wireless Activity Logger. The last 500 lines in the log file are displayed. The wireless server log file is available in the *ORACLE_HOME*`/wireless/logs/` directory or the `/var/tmp/` directory.

Based on the information in the log file, perform corrective steps or contact Oracle Support Services.

> **Note:** You can change the number of lines displayed while viewing the log file, but if you are searching for specific information in a large log file, then it is recommended to view the file using an operating system command, for example, `vi`, `emacs`, or `more`.

Problem 5

There is an error in retrieving metadata. You encounter a temporary error.

Solution 5

Access the servlet error log file, `application.log`, from the *`ORACLE_HOME`*`/j2ee/OC4J_Portal/application-deployments/portal/OC4J_Portal_default_island_1` directory to view troubleshooting information. The servlet error log file records the original error document and its headers, and therefore contains as much information as is available to a standard desktop browser when there is a problem. Use the information in the error log file to perform standard OracleAS Portal troubleshooting analysis.

K.1.23 Error During Export and Import After Upgrading from OracleAS Portal 3.0.9 or 9.0.4

If you run the OracleAS Portal Export and Import, after upgrading from OracleAS Portal 3.0.9 or 9.0.4, you may encounter unexpected errors.

Problem 1

If your transport set includes categories or perspectives, the error may be due to category and perspective templates with incorrect page type IDs; that is, the page type ID is 1 instead of 11.

Solution 1

Check your transport set. If categories or perspectives are included, you can fix this issue by running the following script before running the OracleAS Portal Export and Import utility:

```
SQL> @pstpgcre.sql
```

This script is located at `ORACLE_HOME/portal/admin/plsql/wws/pstpgcre.sql`. This script drops and re-creates the category and perspective templates and their associated pages.

Refer to Section C.10, "Using the Category and Perspective Scripts" for information on how to use the category and perspective scripts.

Problem 2

When you upgrade from OracleAS Portal 3.0.9, category and perspective names are appended with `pageid` and `siteid` and this impacts export and import between portals. For example, if you upgrade a category named GENERAL from version 3.0.9 to version 9.0.4, and then upgrade to version 10.1.2, the name of the upgraded category may be GENERAL_12345_0, where 12345 is the `pageid` and 0 is the `siteid`.

When you export and import between portals, search portlets that are customized to search for categories or perspectives will lose the category and perspective search

criteria, if the source and target portals have different names for the same categories and perspectives.

Solution 2

Ensure that category and perspective names in the source and target portals are exactly the same. For example:

- Change category and perspective names to pre-upgrade names by removing the ID that is appended to the name. For example, change GENERAL_12345_0 back to GENERAL.
- Alternatively, specify new category and perspective names. If there is an existing category or perspective with the name you specify, then you are prompted for a different name.

Note: Make the same changes in the both the source and target portals.

K.2 Diagnosing OracleAS Portal Problems

OracleAS Portal consists of middle and database tiers, each of which consists of numerous components. Components can be distributed across many computers, and they can also simultaneously handle a large number of requests.

You can also use diagnostic tools on OracleAS Portal to analyze and resolve issues about how OracleAS Portal works.

This section contains the following topics:

K.2.1 Enabling ECID Logging

To facilitate problem diagnosis, components can record information related to the requests they receive in log files. This section details how to configure and use various log files to diagnose problems, and how an individual request can be traced from start to finish by using the Execution Context Identifier (ECID).

Execution Context Identifier

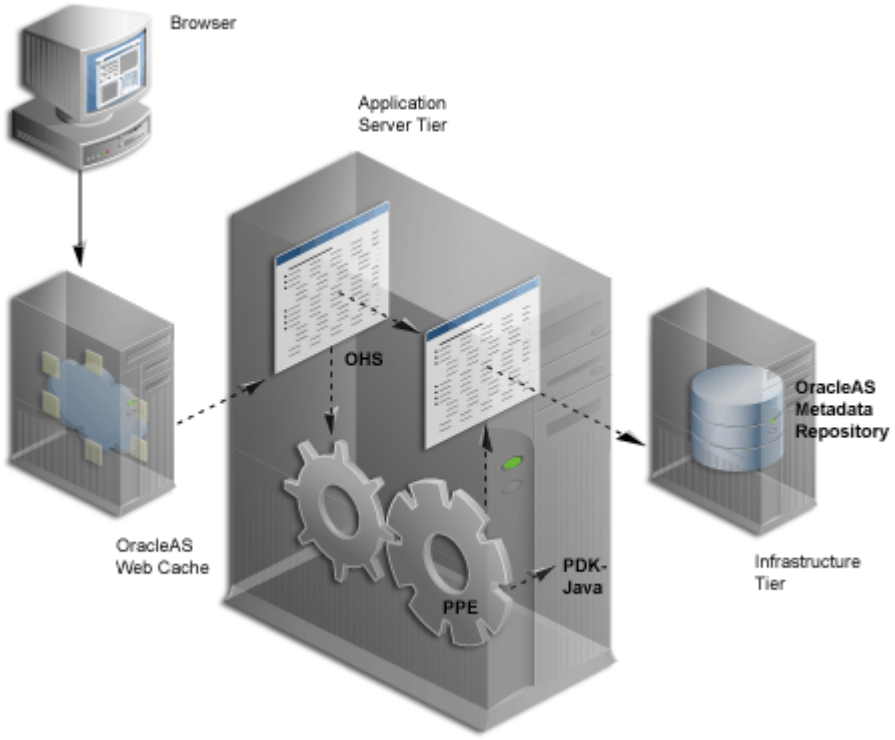
Because OracleAS Portal can satisfy a large number of requests simultaneously, tracing a single request through the various OracleAS Portal components can be difficult because the information relating to these requests is intermingled.

OracleAS Portal makes use of an ECID, which is a unique number that is assigned to a request and attached to the information recorded for that request. As a request is passed from one component to another, the ECID can be incremented to form a

sequence. This means that an individual request can be tracked through any number of components by following this ECID sequence.

An ECID is generated by the first Oracle Application Server component to receive a request without an ECID. You can observe this generation and propagation in Figure K–1, where a dotted arrow depicts a request with an ECID.

Figure K–1 Request Flow with ECID Generation and Propagation



ECID generation is available in OracleAS Web Cache, Oracle HTTP Server, and the Parallel Page Engine (PPE). An ECID is generated only if it does not already exist. In this release, logging of portal invalidations in OracleAS Web Cache now includes the ECID of the original request. This can be used to relate invalidations to original edits or personalizations.

Oracle Containers for J2EE (OC4J) can include the ECID with each log entry it writes, which can be useful for debugging purposes.

> **See Also:** For more information about ECIDs and how they can help you to correlate messages from application server components, refer to the *Oracle Application Server Administrator's Guide*.

K.2.2 Generating Trace Files

When an internal error is detected by a process, information about the error can be written to a trace file. This information is useful in analyzing unhandled exception errors. Refer to Section K.1.19, "Unhandled Exception Errors" for more information.

You can generate a trace file for the sessions in a database instance by using any of the following methods:

- Using PlsqlBeforeProcedure and PlsqlAfterProcedure
- Setting the sql_trace Parameter

- Setting Database Event 10046

K.2.2.1 Using PlsqlBeforeProcedure and PlsqlAfterProcedure

You can enable SQL tracing for a particular session in a database instance by creating a new DAD and setting values for the procedures, `PlsqlBeforeProcedure` and `PlsqlAfterProcedure`.

> **Note:** You can set values for `PlsqlBeforeProcedure` and `PlsqlAfterProcedure` in the original DAD, but this can affect other users. Therefore, it is recommended to create a new DAD.

To enable tracing, perform the following steps:

1. Run the `utltrace.sql` script in the portal schema. The default portal schema name is `portal`.

   > **Note:** The script `utltrace.sql` is available in the *`Oracle_Home`*`/portal/admin/plsql/wwc` directory on the Oracle Application Server Repository Creation Assistant CD-ROM. This CD-ROM is part of the Oracle Application Server CD-ROM Pack from which you installed OracleAS Portal.
   >
   > For OracleAS Portal 10.1.2.0.2, the source code of `utltrace.sql` is available on Oracle *Metalink* at
   >
   > `http://metalink.oracle.com`

2. Create a new DAD, for example `portal_trc`. Refer to the *Oracle Application Server mod_plsql User's Guide*.
3. Click **OK** to go back to the PL/SQL properties for the HTTP Server.
4. In the DAD Status section, click the DAD that you created.
5. Click **Advanced**, and then set the following values:
   - PlsqlBeforeProcedure: `portal.wwutl_trace.trace_on`
   - PlsqlAfterProcedure: `portal.wwutl_trace.trace_off`
6. Stop and start the HTTP Server and OC4J_Portal.

   > **Note:** The cookie for the new DAD must be the same as the portal DAD so that you can replace the DAD in the portal URL. If the cookie name for the new DAD is different from the portal DAD, then update the cookie name of the new DAD with that of the portal DAD.
   >
   > Refer to Section 11.2.1, "Checking the PlsqlSessionCookieName Value" for details about checking or updating the cookie name.

7. Change the DAD in the OracleAS Portal URL to the new DAD that you defined in Step 2 and use this URL to access OracleAS Portal. For example, change:

   `http://<`*`hostname`*`>:<`*`port`*`>/portal/pls/portal/portal.home`

   to:

```
http://<hostname>:<port>/portal/pls/portal_trc/portal.home
```

After you set the event, two trace files will be written to the `user_dump_dest` directory. Open and view this file to check for any information about the unhandled exception error that you encountered.

K.2.2.2 Setting the sql_trace Parameter

You can enable tracing by setting the `sql_trace` database initialization parameter.

After setting an event, for the event to take effect, you must restart the database instance.

To enable tracing for all sessions in the database instance, set the `sql_trace` parameter to `true` in `SPFILE` using the following SQL syntax:

```
ALTER SYSTEM SET
sql_trace=true
COMMENT = 'turn tracing ON for all sessions'
SCOPE=SPFILE;
```

To turn tracing off, use the following syntax:

```
ALTER SYSTEM SET
sql_trace=false
COMMENT = 'turn tracing OFF for all sessions'
SCOPE=SPFILE;
```

If you are unfamiliar with the steps involved in updating a database initialization parameter file, refer to the section, "Managing Initialization Parameters Using a Server Parameter File", in the *Oracle Database Administrator's Guide* in the Oracle Database 10*g* documentation library.

K.2.2.3 Setting Database Event 10046

You can enable tracing for all sessions in a database instance by setting database event 10046. Event 10046 is the equivalent of setting the value of `sql_trace` to `true` in the parameter file. In addition, while setting event 10046, you can also specify the level of tracing.

> **Note:** Setting events should be done only with the help of Oracle Support Services.

Table K–1 describes different trace levels.

Table K–1 Trace Levels

Level	Description
1	Used to enable standard `SQL_TRACE` functions. This is the default value.
4	Used to enable standard `SQL_TRACE` functions and to trace bind values.

Table K–1 (Cont.) Trace Levels

Level	Description
8	Used to enable standard SQL_TRACE functions and to trace waits. This is used mainly for identifying latch wait, but it can also be used to identify full table scans and index scans.
12	Used to enable standard SQL_TRACE functionality and to trace bind values and waits.

Note: When you set a database event, consider the following points:

- You cannot set an event when a database instance is running.
- You can set events without having to mount or open the database. You can run the command with the database instance in NOMOUNT state.

To enable tracing by setting database event 10046 in SPFILE, use the following syntax:

```
ALTER SYSTEM SET
EVENT = '10325 trace name context forever, level 10:10015 trace name
context forever, level 1'
COMMENT = 'Debug tracing of control and rollback'
SCOPE=SPFILE;
```

If you are unfamiliar with the steps involved in updating a database initialization parameter file, refer to the section, "Managing Initialization Parameters Using a Server Parameter File", in the *Oracle Database Administrator's Guide* in the Oracle Database 10*g* documentation library.

K.2.3 Viewing the Diagnostic Output of Components

The various OracleAS Portal components can have their diagnostic output configured. The following are the components:

- JPDK
- Portal Services
- Parallel Page Engine
- Oracle Application Server Portal Developer Kit
- OracleAS Metadata Repository
- OracleAS Web Cache

K.2.3.1 JPDK

Java Portal Development Kit (JPDK) provides a framework for the construction of Java-based portlets and portlet providers. A Java-based provider or Web provider is written as a Web application. The JPDK includes a logging mechanism that is controlled based on each Provider Adapter.

Table K–2 lists and describes the available logging levels. The acceptable logging level values range from 1 to 8 and build incrementally. For example, at logging level 3, the output for logging levels 1 and 2 are also recorded.

Table K–2 Logging Levels

Logging Level	Description
1	Configuration
2	Severe Errors
3	Warnings
4	Exceptions
5	Performance
6	Detailed Performance Information
7	Information
8	Debug

JPDK Log File Contents

Diagnostic information about a provider adapter is recorded in the servlet context log file named `application.log`.

There are two types of JPDK messages:

- Standard JPDK Messages
- Performance JPDK Messages

Standard JPDK Messages

Here is an example of a standard JPDK message that you might find in a Provider Adapter's `application.log` file:

```
03/12/31 02:58:59 jpdk: [instance=1926_EXPIRESSAMPLE_886361,
id=1024597399815ApplicationServerThread-12,4] Beginning rendering of portlet:
1926_EXPIRESSAMPLE_886361
```

The content of the standard JPDK message is as follows:

- Date and time: 03/12/31 02:58:59
- Web application: jpdk
- ECID, sequence number: id=1024597399815ApplicationServerThread-12,4
- Portlet instance identifier: instance=1926_EXPIRESSAMPLE_886361
- Message: Beginning rendering of portlet: 1926_EXPIRESSAMPLE_886361

The portlet instance identifier identifies a specific portlet instance on a specific page, and can be broken down as follows:

- Internal sequence number: 1926
- Portlet name: EXPIRESSAMPLE
- Provider identifier: 886361

Additional details about some of these values are shown in Table K–3.

Table K–3 JPDK Standard Message Attributes

Value	Detail
ECID	Some messages carry null ECID and portlet instance identifier values. These are typically SOAP messages from the repository.
Portlet instance identifier	This is the same as ECID except that the portlet instance identifier is `null` in this case, because the message does not relate to a particular portlet instance.

K.2.3.2 Portal Services

Portal Services performance is logged through Oracle HTTP Server. The default directory for the `error_log` file is *`ORACLE_HOME`*`/Apache/Apache/log` on UNIX and *`ORACLE_HOME`*`\Apache\Apache\log` on Windows. Logging is controlled by the `LogLevel` parameter in the configuration file `httpd.conf`.

K.2.3.3 Parallel Page Engine

The Parallel Page Engine (PPE) is a shared server process servlet that accepts data representing a page layout, and then converts this data into a page containing portlets.

PPE logging can be controlled at the servlet and request level. If a request logging level is not specified, then the servlet level is used for the request. If both servlet and request logging levels are specified, then the higher of the two is used for the request.

Servlet-Level Logging

PPE servlet-level logging is controlled by the `logmode` servlet initialization argument. The values for `logmode` are the following:

- `none`
- `perf`
- `debug`
- `request`
- `content`
- `parsing`
- `all`

The values build incrementally. For example, if `logmode` is set to `content`, then `content`, `request`, `debug`, and `perf` messages are also recorded. The default value is `none`. A value of `all` allows every logging message to be included.

As the PPE is a servlet, configuration varies with the Servlet container on which it is deployed. Under OracleAS Portal, the servlet container is OC4J and `logmode` can be found in the portal `web.xml` file. This XML file contains properties for more than just the PPE, and consequently, `logmode` can appear more than once. It is important to modify the correct `logmode` value.

The following can be found inside the `page` servlet clause:

```
<servlet>
     <servlet-name>page</servlet-name>
     <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
     .
     .
     .
```

```
     <init-param>
          <param-name>logmode</param-name>
          <param-value>perf</param-value>
     </init-param>
     .
     .
     .
</servlet>
```

If the value of `logmode` is altered, then OC4J must be restarted for this change to take effect. The `web.xml` file can be found at:

ORACLE_HOME`/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF`

Request-Level Logging

PPE request-level logging is controlled by the `_debug` URL parameter. For example, to specify request-level logging for the following URL:

```
http://myserver.myplace.com:3000/portal/page?_pageid=111&_dad=myDAD&_
schema=mySchema
```

You must manually insert the following:

```
&_debug=3
```

The resultant URL is as shown:

```
http://myserver.myplace.com:3000/portal/page?_pageid=111&_dad=myDAD&_
schema=mySchema&_debug=3
```

Table K–4 lists the values for `_debug`.

Table K–4 PPE Request Log Levels

Value	Detail
0	Activates page-debugging information
1	Activates page-debugging information
2	Logs to page and sets the request log mode to `debug`
3	Logs to page and sets the request log mode to `request`
4	Logs to page and sets the request log mode to `content`
5	Logs to page and sets the request log mode to `parsing`

Page Logging

With `_debug` set to `2`, `3`, `4`, or `5`, page logging is activated. This means that messages logged for the request are recorded in the PPE log file, and in the page returned.

Page logging is a means by which you can obtain detailed information relating to a request. As a result, it is also a security issue, for which the `urlDebugMode` servlet initialization argument is provided.

The `urlDebugMode` argument can be found alongside `logmode` in the portal `web.xml` file:

```
<init-param>
     <param-name>urlDebugMode</param-name>
```

```
     <param-value>4</param-value>
</init-param>
```

The `urlDebugMode` argument can be found inside the `page` servlet clause:

```
<servlet>
     <servlet-name>page</servlet-name>
     <servlet-class>oracle.webdb.page.ParallelServlet</servlet-class>
     .
     .
     .
     <init-param>
          <param-name>urlDebugMode</param-name>
          <param-value>4</param-value>
     </init-param>
     .
     .
     .
</servlet>
```

Table K–5 lists the values for the `urlDebugMode` argument. The default value is `1`.

Table K–5 PPE urlDebugMode Levels

Value	Detail
`None`	Ignore the `_debug` URL parameter.
`0`	Allow `_debug` to be `0`.
`1`	Allow `_debug` to be `0` or `1`.
`2`	Allow `_debug` to be `0`, `1`, or `2`.
`3`	Allow `_debug` to be `0`, `1`, `2`, or `3`.
`4`	Allow `_debug` to be `0`, `1`, `2`, `3`, or `4`.
`5`	Allow `_debug` to be `0`, `1`, `2`, `3`, `4`, or `5`.

PPE Log File Contents

PPE diagnostic messages are recorded in the servlet context `application.log` file. This file can be found at:

```
ORACLE_HOME/j2ee/OC4J_
Portal/application-deployments/portal/<island>/application.log
```

There are two types of PPE messages:

- Standard PPE Messages
- Performance PPE Messages

Standard PPE Messages

The following is an example of a standard PPE message found in its log file:

```
03/12/31 11:54:35 portal: id=22020914339,0 DEBUG: active=53  ContentFetcher
 Unexpected Exception Request Failed:java.lang.IllegalArgumentException
 name=content-fetcher52 label=dbPortlet url=https://abc.company.com:5001/pls/ptl_
9_0_4_0_87/!PTL_9_0_4_0_87.wwpro_app_provider.execute_portlet/391497559/4
 time=38975ms timeout=15000ms process=ResponseHeaders
```

The content of this standard PPE message is as follows:

- Date and time: 03/12/31 11:54:35
- Web application: portal
- logmode flag: DEBUG
- Active count: active=53
- ECID: id=22020914339, 0
- Message: ContentFetcher Unexpected Exception Request Failed

Table K–6 provide details relating to some of these values.

Table K–6 PPE Standard Message Attributes

Value	Detail
`logmode` flag	Indicates that log mode is `debug` or higher. If `logmode` is set to `perf` and is therefore lower than `debug`, then the `logmode` flag is not included in the message.
Active count	Indicates the number of threads in the PPE thread group. If `logmode` is set to `perf` and is therefore lower than `debug`, then the active count is not included in the message.
ECID	Can be `null`. A message with such a value relates to a PPE background task (such as clearing pooled objects). Background tasks do not relate to a request, and therefore do not have an ECID specified.

Performance PPE Messages

The following is an example of a performance PPE message found in the log file:

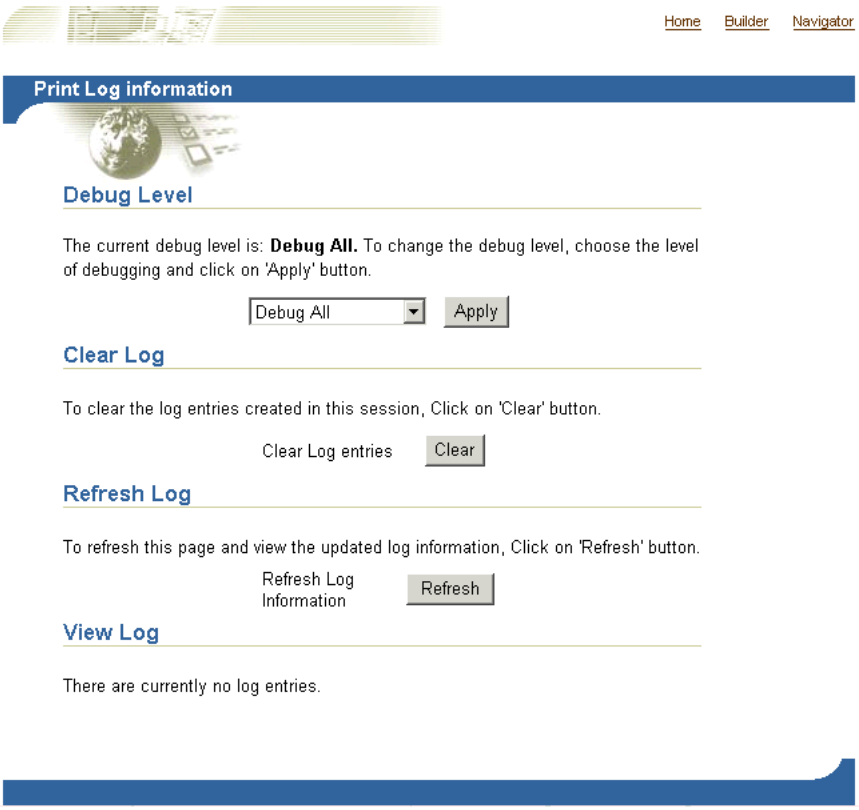
```
05/06/16 06:06:37 portal: [perf] 140.87.20.124
 https://abc.company.com:8250/portal/ page?_pageid=40,1&_dad=portal&_
schema=PORTAL&_mode=16 id=8198110376563,1 type=page name=40,1 status=200
 user=PORTAL subscriberID=1 reqTime=187ms waitTime=0ms cache=(null) timeout=No
 redirects=0 bytes=33865 authLevel=10 webCacheStatus=(null) webCacheExpires=(null)
 webCacheAge=(null) csConv=No readTime=No,0ms pageTimeout=No procTime=0ms
```

K.2.3.4 Oracle Application Server Portal Developer Kit

Oracle Application Server Portal Developer Kit (PDK) provides a framework for the construction of portlets and portlet providers in a variety of Web languages including Java, Web Services, XML, ASP, Perl, and PL/SQL. The PDK therefore includes JPDK.

The PDK provides a core logging mechanism, which is augmented by logging in specific developers kits. PDK logging is controlled through a Web-based user interface as shown in Figure K–2.

Figure K–2 PDK Logging Page



This PDK Logging Page can be found at:

`http://<host>:<port>/portal/pls/<dad>/<schema>.wwpro_log.render`

A sample URL is as follows:

`http://myserver.myplace.com:3000/portal/pls/portal/PORTAL.wwpro_log.render`

From this page you can apply the logging levels described in Table K–7.

Table K–7 PDK Log Levels

Level	Detail
No debugging	No logging
PROHTTPJ	Provider framework logging
PROGRP	Provider logging
ADAPTER	Federated portal adapter logging
CACHE	Cache logging
FORCE	Internal to Oracle
INVAL	Invalidation logging
PROREG	Provider registration logging
PROLOGIN	Page metadata generation, login, and session initialization logging
PROPROV	Provider communication logging
PROPMR	Portlet metadata repository logging

Table K–7 (Cont.) PDK Log Levels

Level	Detail
PROHTTP	Web provider framework logging
All	All logging levels activated

PDK Log File Contents

You can view PDK log entries from the same page used to configure PDK logs, as shown in Figure K–3.

Figure K–3 Log Entries in the PDK Logging Page

View Log

The following table lists the log entries that were created.

Log ID	Start Time	Name	Information	key_1	key_2
12165	09-SEP-2003 04:25:52	wwpro_app_provider.execute_portlet	[msecs] Portlet Title: SSO Server Administration Reference_path: 34_LOGINSERVERADMIN_604753661 **Before Show** Caching Level: Caching Key: Caching Period:	604753661	4
12166	09-SEP-2003 04:26:06	wwpro_app_provider.execute_portlet	[msecs] Portlet Title: SSO Server Administration Reference_path: 34_LOGINSERVERADMIN_604753661 **Before Show** Caching Level: Caching Key: Caching Period:	604753661	4

K.2.3.5 OracleAS Metadata Repository

OracleAS Metadata Repository consists of all the metadata, portal content, and PL/SQL code that reside in the OracleAS Portal database schema. The PL/SQL code that executes in the OracleAS Portal schema also generates diagnostics output that can be correlated with diagnostics output generated from the other components of OracleAS Portal.

Because the log file is produced by OracleAS Metadata Repository, the database running OracleAS Portal must be configured to allow this. To do this, you must use the `CREATE DIRECTORY` statement to create a directory object.

A directory object specifies an alias for a directory on the server file system where external files and external table data are located.

> **Note:** All directories are created in a single namespace and are not owned by an individual schema. You can secure access to the files stored within the directory structure by granting object privileges on the directories to specific users.

To use the `CREATE DIRECTORY` statement, you must have the `CREATE ANY DIRECTORY` system privilege. When you create a directory, you are automatically granted the `READ` and `WRITE` object privileges on that directory. You, or the database administrator, can in turn grant these privileges to other users and roles.

Note: WRITE privileges on a directory are useful in connection with external tables. They let the grantee determine whether the external table agent can write a log file or a bad file to the directory.

You must also create a corresponding operating system directory for file storage. Your system or database administrator must ensure that the operating system directory has the correct READ and WRITE privileges for Oracle Database processes

Privileges granted for the directory are created independently from the privileges defined for the operating system directory, and the two may, or may not, correspond exactly. For example, an error occurs if a sample user hr is granted READ privilege on the directory object, but the corresponding operating system directory does not have READ privilege defined for the Oracle Database processes.

To create a directory object, use the following syntax:

```
CREATE [OR REPLACE] DIRECTORY AS 'path_name';
```

Table K–8 describes the parameters used in CREATE DIRECTORY syntax.

Table K–8 CREATE DIRECTORY Parameters

Semantics	Description
OR REPLACE	Specify OR REPLACE to re-create the directory database object, if it already exists. You can use this clause to change the definition of an existing directory without deleting, re-creating, and regranting database object privileges previously granted on the directory. Existing users with privileges to access a redefined directory can to access the directory without being granted the privileges again.
directory	Specify the name of the directory object to be created. The maximum length of the directory name is 30 bytes. You cannot qualify a directory object with a schema name. Oracle Database does not verify that the directory you specify actually exists. Therefore, you must ensure that you specify a valid directory in your operating system. In addition, if your operating system uses case-sensitive path names, then be sure to specify the directory in the correct format. You need not include a trailing slash at the end of the path name.
path_name	Specify the full path name of the operating system directory of the server where the files are located. The single quotation marks are required, with the result that the path name is case-sensitive.

For example, the following statement creates a directory database object that points to a directory on the server:

```
CREATE DIRECTORY admin AS 'oracle/admin';
```

The following statement redefines the bfile_dir directory database object to enable access to files stored in the operating system directory /private1/lob/files:

```
CREATE OR REPLACE DIRECTORY bfile_dir AS '/private1/LOB/files';
```

In the case of Oracle Database releases earlier than 9.2, for the PL/SQL code to generate diagnostics output, update the database initialization parameter file by adding the following line:

```
UTL_FILE_DIR=<directory where you want to write the log file>
```

If you are unfamiliar with the steps involved in updating a database initialization parameter file, refer to the section, "Managing Initialization Parameters Using a Server Parameter File", in the *Oracle Database Administrator's Guide* in the Oracle Database 10*g* documentation library.

There can be many `UTL_FILE_DIR` entries, so if the directory you wish to write to is already defined, then there is no need to modify this file.

> **Note:** On installing of OracleAS Metadata Repository, if the database you are installing into has the `UTL_FILE_DIR` parameter set, then the OracleAS Portal installer configures OracleAS Metadata Repository such that it uses the first directory defined by the database parameter as the location for the OracleAS Metadata Repository log file. If the `UTL_FILE_DIR` directory is not configured, then OracleAS Metadata Repository logging is not set up on installation.

OracleAS Metadata Repository logging is performed through a logging package. This logging package is controlled using the script `logcfg.sql` which you must run from SQL*Plus.

The `logcfg.sql` script can be found at:

ORACLE_HOME`/portal/admin/plsql/wwc`

The `logcfg.sql` script can take five parameters in the following order: `log_level`, `log_state_level`, `log_format`, `log_file`, and `log_directory`. If less than five parameters are supplied, then one or more values are requested. If no value is received in response to this request, then the current value is maintained.

Table K–9 details the `logcfg.sql` parameters.

Table K–9 Repository Logging Package Parameters

Parameter	Detail
`log_level`	Describes the level of messages recorded. The values are the following: ■ `0`: None ■ `1`: Error ■ `2`: Warning ■ `3`: Information ■ `4`: Trace ■ `5`: Debug ■ `6`: Fine Debug The values build incrementally. The default value is `1`.

Table K–9 (Cont.) Repository Logging Package Parameters

Parameter	Detail
log_state_level	Describes the level of messages for which state information will automatically be logged. The values are the following: ■ 0: None ■ 1: Error ■ 2: Warning ■ 3: Information ■ 4: Trace ■ 5: Debug ■ 6: Fine Debug The values build incrementally.
log_format	Describes the format of automatically recorded context information, which is different from state information. The values are the following: ■ 0: Simple ■ 1: Detailed
log_file	Specifies the name of the log file to write to. An attempt is made to create this file if it does not already exist.
log_directory	Specifies the directory in which the log_file parameter exists. This value can be either a physical path or a directory object. If the value is a physical directory, it must be defined in the database parameter file under the UTL_FILE_DIR property. For example: utl_file_dir=/export/home/oracle/as1014/logs If the database parameter file is modified, then the database must be restarted for this change to take effect. If the value is a directory object, then you must specify the directory object name in uppercase. For example, LOGS.

For example, you can run the logcfg.sql script from SQL*Plus as follows:

```
@logcfg.sql 3 3 1 portal.log /export/home/oracle/as1014/logs
```

If you point to a directory object instead, then you must specify the directory object name in uppercase. For example, to point to a directory object named logs, you must run the logcfg.sql script from SQL*Plus as follows:

```
@logcfg.sql 3 3 1 portal.log LOGS
```

After running logcfg.sql, the usage is displayed:

```
Configure Portal diagnostics
usage:
logcfg.sql <log_level> <log_state_level> <log_format> <log_file> <log_directory>
If for any of the params a null value is specified the existing value will be
maintained.
Log levels:
0 : None (turn diagnostics off)
1 : Error
2 : Warning
3 : Information
4 : Trace
5 : Debug
```

```
6 : Fine Debug
Log formats:
0 : Simple
1 : Detailed
```

The current values are also displayed:

```
Current settings:
Log level:     3
Log state level:     3
Log format:     1
Log file:     portal.log
Log directory:     /export/home/oracle/as101202/dblogs
```

To truncate the OracleAS Metadata Repository diagnostics log file, run the SQL script `logtrunc.sql` located at: *ORACLE_HOME*`/portal/admin/plsql/wwc`

Repository Log File Contents

The location of the OracleAS Metadata Repository diagnostic information is dictated by the repository diagnostics package parameters `log_file` and `log_directory`.

The following is an example of an ERROR type message found in the OracleAS Metadata Repository log file:

```
[06-AUG-2002 15:02:15] [ERROR] id=(102733434) ctx=wwsrc_simple_edit.render_simple_
edit_prefs user=PORTAL subscriberId=1 language=us userAgent="Mozilla/5.0"
ip=192.0.0.1
ORA-30625: method dispatch on NULL SELF
[START-ERROR-STACK]
ORA-30625: method dispatch on NULL SELF
[END-ERROR-STACK]
[START-CALL-STACK]
----- PL/SQL Call Stack -----
object          line          object
handle          number        name
81b35e6c        350           package body PORTAL.WWLOG_API_DIAG
81b35e6c        443           package body PORTAL.WWLOG_API_DIAG
81b35e6c        526           package body PORTAL.WWLOG_API_DIAG
86765ac8        259           package body PORTAL.WWSRC_SIMPLE_EDIT
86765ac8        334           package body PORTAL.WWSRC_SIMPLE_EDIT
84317130        19            package body PORTAL.WWSBR_BASIC_SEARCH
88857980        713           package body PORTAL.WWSBR_SITEBUILDER_PROVIDER
8323ad18        1             anonymous block
87e53d5c        648           package body PORTAL.WWPRO_API_PROVIDER
81ae1e50        2644          package body PORTAL.WWPOB_PAGE
877a0d9c        12            anonymous block
[END-CALL-STACK]
[START-QUERY-STRING]
_providerid=102274117
_portletid=14
_mode=5
_title=Basic%20Search
_referencepath=1875_BASICSEARCH_102274117
_back_url=http%3A%2F%2Fmyserver.myplace.com%3A3000%2Fpls%2Fportal%
_portlet_reference=33_31293_33_1_1
[END-QUERY-STRING]
```

The message in the log file is as follows:

`ORA-30625: method dispatch on NULL SELF: -` The message itself.

The log file also has context and state information.

Context Information

Context information is produced in one of two formats, detailed or simple, as specified by `log_format` parameter. In the following example, the format is detailed:

- **`06-AUG-2002 15:02:15:`** Date and time
- **`ERROR:`** Message level
- **`id=(102733434, 1):`** ECID
- **`ctx=wwsrc_simple_edit.render_simple_edit_prefs:`** Message context
- **`user=PORTAL:`** Database user
- **`subscriberId=1:`** Subscriber identifier
- **`language=us:`** Globalization Support language
- **`userAgent="Mozilla/5.0":`** User agent
- **`ip=192.0.0.1:`** Client IP address

The simple format is a subset of the detailed format and includes the following information:

- **`06-AUG-2002 15:02:15:`** Date and time
- **`ERROR:`** Message level
- **`ctx=wwsrc_simple_edit.render_simple_edit_prefs:`** Message context

Table K–10 provides additional details relating to some of these values.

Table K–10 Repository Context Attributes

Value	Detail
Client IP address	Typically, this is the IP address of the client browser or HTTP proxy in use. Because the OracleAS Portal page assembly process uses loopback calls, the IP address can also represent the middle tier itself.
Subscriber identifier	This identifies which subscriber has accessed the repository.
User agent	This is description of the browser in use.

State Information

State information consists of the error stack, call stack, and a query string. Examples of each of these are as follows:

> **Note:** The PL/SQL error stack is displayed only if a message of type ERROR is logged.

Error stack:

```
[START-ERROR-STACK]
ORA-30625: method dispatch on NULL SELF
[END-ERROR-STACK]
```

Call stack:

```
[START-CALL-STACK]
```

```
----- PL/SQL Call Stack -----
object          line        object
handle          number      name
81b35e6c        350         package body PORTAL.WWLOG_API_DIAG
81b35e6c        443         package body PORTAL.WWLOG_API_DIAG
81b35e6c        526         package body PORTAL.WWLOG_API_DIAG
86765ac8        259         package body PORTAL.WWSRC_SIMPLE_EDIT
86765ac8        334         package body PORTAL.WWSRC_SIMPLE_EDIT
84317130        19          package body PORTAL.WWSBR_BASIC_SEARCH
88857980        713         package body PORTAL.WWSBR_SITEBUILDER_PROVIDER
8323ad18        1           anonymous block
87e53d5c        648         package body PORTAL.WWPRO_API_PROVIDER
81ae1e50        2644        package body PORTAL.WWPOB_PAGE
877a0d9c        12          anonymous block
[END-CALL-STACK]
```

Query string:

```
[START-QUERY-STRING]
_providerid=102274117
_portletid=14
_mode=5
_title=Basic%20Search
_referencepath=1875_BASICSEARCH_102274117
_back_url=http%3A%2F%2Fmyserver.myplace.com%3A3000%2Fpls%2Fportal%
_portlet_reference=33_31293_33_1_1
[END-QUERY-STRING]
```

Repository Diagnostics Log File Registration

Oracle Enterprise Manager 10*g* provides a Log Reader and Log Viewer. The Log Reader allows administrators to upload log files to a file-based log repository. The Log Viewer allows administrators to view and query log entries loaded into the repository. See Section K.2.4, "Using Application Server Control Console Log Viewer" for more information.

To load and view the Repository Diagnostics log file entries, you must first register the log file with Oracle Enterprise Manager 10*g*. To do this, edit the following file:

```
ORACLE_HOME/diagnostics/config/registration/PORTAL.xml
```

In this file, there is a template entry that you can copy and expand to reflect details of your log file. The template is as follows:

```
<logs xmlns="http://www.oracle.com/iAS/EMComponent/ojdl" helpIDLogs="psm_cs_xml_
log_info">

<!--
<log path="<PATH>" componentId="PORTAL">
<logreader type="SimpleTextLog">
    <property name="ComponentId" value="PORTAL"/>
    <property name="ModuleId" value="Portal:<INSTANCE>"/>
    <property name="TimestampFormat" value="[dd-MMM-yyyy HH:mm:ss]"/>
    <property name="TimestampLocale" value="en_US"/>
</logreader>
<logviewer ComponentName="ID_VLOGS_PORTAL_REP@ResourceBundle"
           LogType="ERROR"
           LogName="Diagnostics for Portal instance <INSTANCE>"/>
</log>
-->

</logs>
```

Modify the following information in the copied template entry:

- `<PATH>`: The absolute path and file name of the log file.
- `<INSTANCE>`: The name of the OracleAS Portal target in Oracle Enterprise Manager 10*g*, if it is defined. If there is no corresponding OracleAS Portal target in Oracle Enterprise Manager 10*g*, then use the name of the OracleAS Portal instance and database details, for example, `<portal schema name>-<db service name>`. This value is used to distinguish this log entry in the Log Viewer from other OracleAS Portal instance log entries.

After you have saved the new `PORTAL.xml` entry, the Log Reader starts uploading the log file periodically, and you can use the Log Viewer to view and query this log file.

Because the OracleAS Metadata Repository can be accessed through many middle tiers, you need to do the following:

- Register the Repository Diagnostics log file with one of the Oracle Enterprise Manager 10*g* Application Server Control Console instances that is monitoring an OracleAS Portal middle tier.
- If the OracleAS Portal database is on a computer other than the OracleAS Portal middle tier, ensure that the log file is accessible over a network file system.
- To perform log correlation in a multiple middle-tier environment, you need to register the Repository Diagnostics log file with each Oracle Enterprise Manager 10*g* instance monitoring an OracleAS Portal middle tier.

Note: . Using Oracle Enterprise Manager 10*g*, you have to update the location of the Repository Diagnostics log file in the `PORTAL.xml` file located at *ORACLE_HOME*`/diagnostics/config/registration/`.

K.2.3.6 OracleAS Web Cache

Oracle Application Server Web Cache events and errors are stored in an event log. The event log helps you to determine which documents or objects have been inserted into the cache. It can also identify listening port conflicts or startup and shutdown issues. By default, the event log has a file name of `event_log` and is stored in *ORACLE_HOME*`/webcache/logs` on UNIX and *ORACLE_HOME*`\webcache\logs` on Windows.

See Also: *Oracle Application Server Web Cache Administrator's Guide*

K.2.4 Using Application Server Control Console Log Viewer

You can use Oracle Enterprise Manager 10*g* Application Server Control Console to view and query entries from the following Oracle Application Server log files. This helps you to diagnose issues relating to OracleAS Portal. The relevant Oracle Application Server component log files include the following:

- **Portal:<instance>**: Displays a single, diagnostic error log file for each portal instance named <*customer_specified_log_name*>. This log file is generated by the relevant OracleAS Metadata Repository.
- **HTTP_Server**: Displays multiple error or access log files named `error_log` and `access_log`. These log files contain all relevant Portal Service logging information.
- **OC4J_Portal**: Displays multiple application log files named `application.log`. This log file contains all relevant PPE logging information.

- **JPDK**: Displays the location `j2ee/home/application-deployments/jpdk/application.log` for the JPDK sample providers in a standalone OC4J. In an Oracle Application Server middle tier, the location is similar to the addition of a directory for the default island.
- **Web Cache**: Displays error and access log file names `event_log` and `access_log`.

Before you can use the OracleAS Metadata Repository log file with Application Server Control Console Log Viewer, you must complete a registration process. Refer to the section "Repository Diagnostics Log File Registration" for instructions.

If your JPDK OC4J instance is *not* located in the OracleAS Portal middle tier Oracle home, then you may view its log file only through the local Application Server Control Console instance. If you want to perform diagnostic correlation, then you must follow a similar remote registration process to that described for the OracleAS Metadata Repository log file when it is remotely located.

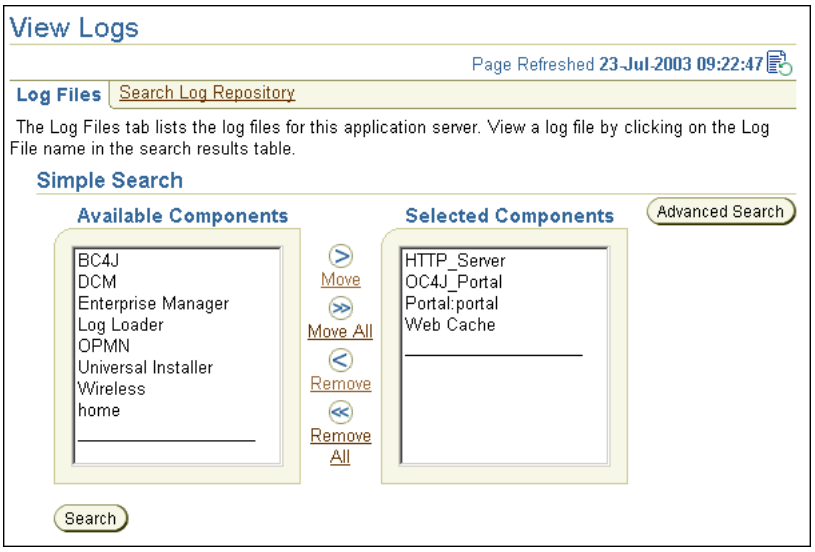
In addition to viewing the log file entries with Application Server Control Console Log Viewer, you can also perform advanced diagnostics by correlating entries across log files using the ECID value. See Section K.2.1, "Enabling ECID Logging" for more information. This drill-down correlation is automatically provided by the Application Server Control Console Log Viewer.

To view log file entries, click **Logs** in Application Server Control Console, which is located at the top and bottom of every Application Server Control Console component home page.

> **See Also:** For detailed instructions on how to use the Log Viewer, refer to the *Oracle Application Server Administrator's Guide*. It describes how to perform advanced queries for diagnostic log file information, search through diagnostic messages (collected from selected Oracle Application Server components) in the Log Repository, and correlate messages across log files and components.

Figure K–4 shows an example of Oracle Application Server components selected in the View Logs page.

Figure K–4 Application Server Control Console View Logs Page



K.2.5 Using OracleAS Portal Diagnostics Assistant

Use OracleAS Portal Diagnostics Assistant to gather information if you are troubleshooting issues after OracleAS Portal installation. Problems can vary from accessing the OracleAS Portal, to users getting errors at different levels within OracleAS Portal.

You can diagnose issues by reviewing the results from OracleAS Portal Diagnostics Assistant. Alternatively, you can upload the results to Oracle Support Services so that they can assist in troubleshooting the problem for you.

The generated report includes the following sections:

- Errors and violations summary (available only if violations are detected by OracleAS Portal Diagnostics Assistant)
- OracleAS Portal Repository database information
- OracleAS Single Sign-On database information
- Oracle Internet Directory diagnostics report
- Oracle Text diagnostics report
- Apache error log file analysis

In addition, all OracleAS Portal-related configuration files and log files are collected and zipped for your convenience.

To run OracleAS Portal Diagnostics Assistant, you need to use the script `pda.csh` (UNIX) or `pda.cmd` (Windows). Each time you run the script, a new directory is created for the generated files under the directory where you downloaded the OracleAS Portal Diagnostics Assistant zip file. The directory names have a timestamp format, for example, `040623132344` which means:

Year: 04
Month: 06
Day: 23
Hour: 13
Minutes: 23
Seconds: 44

After running OracleAS Portal Diagnostics Assistant, locate the appropriate directory and open the HTML report named `pda.htm` in a browser window. You can navigate through the report and review the diagnostics information.

If you want Oracle Support Services to assist you in troubleshooting the problem, then log on to Oracle *Metalink* at `http://metalink.oracle.com`, and upload the generated ZIP file named `PDA<directory_name>.zip`, for example, `PDA040623132344.zip`.

For detailed information about using OracleAS Portal Diagnostics Assistant and the information collected, refer to the `readme` file located in the directory in which you downloaded OracleAS Portal Diagnostics Assistant.

Running OracleAS Portal Diagnostics Assistant

To generate diagnostics information using OracleAS Portal Diagnostics Assistant, perform the following steps:

1. Check the Support and Metalink section on OTN for the latest update/patch information for OracleAS Portal Diagnostics Assistant at:

   `http://www.oracle.com/technology/`

   Download the latest OracleAS Portal Diagnostics Assistant script. Support/Upgrade is located in the Product Information section.

2. Ensure that the *`ORACLE_HOME`* environment variable is set to the correct OracleAS Portal middle tier Oracle home directory.

   If you try to run OracleAS Portal Diagnostics Assistant from a database Oracle home directory, it fails and no diagnostics information is collected.

3. Navigate to the location where you downloaded and unzipped OracleAS Portal Diagnostics Assistant.

4. Run OracleAS Portal Diagnostics Assistant on UNIX as follows:

```
pda.csh
-schema <portal schema name>
-password <portal schema password>
-connect <Portal connect string>
-ssoSchema <SSO schema name>
-ssoPassword <SSO schema password>
-ssoConnect <SSO connect string>
[-apacheLogDir <directory name>]
[-apacheLogName <file name>]
[-logFileLimit <number of rows>]
[-show]
[-showall]
```

   Run the script without any parameters to get Help information.

   Table K–11 lists and describes the parameters used with the OracleAS Portal Diagnostics Assistant script.

Table K–11 OracleAS Portal Diagnostics Assistant Script Parameters

Parameter	Description
-schema	Name of the OracleAS Portal schema. This parameter is mandatory. Default = `Portal`.
-password	Password for the OracleAS Portal schema. This parameter is mandatory. Default = `portal_schema`.

Table K–11 (Cont.) OracleAS Portal Diagnostics Assistant Script Parameters

Parameter	Description
-connect	Connect string for the OracleAS Portal schema. Use the format `<host>:<port>:<sid>`. This parameter is mandatory.
-ssoSchema	Name of the OracleAS Single Sign-On schema. This parameter is mandatory.
-ssoPassword	Password for the OracleAS Single Sign-On schema. This parameter is mandatory.
-ssoConnect	Connect string for the OracleAS Single Sign-On schema. Use the format `<host>:<port>:<sid>`. This parameter is mandatory.
-apacheLogDir	Directory for Oracle HTTP Server error log file. This parameter is optional. Default = *ORACLE_HOME*`/Apache/Apache/logs`.
-apacheLogName	Error log file name. This parameter is optional. Default = `error_log`.
-logFileLimit	The number of rows in the error log file. This parameter is optional. Default = `10000`.
-show	Generates diagnostics information with only the necessary set of queries. This is the default mode for generating diagnostics information when no other parameters are selected.
-showall	Generates diagnostics information with all the queries. This mode has an additional query that retrieves all the portal objects and their privileges from the relevant security table. Because of this, generating diagnostics information in the `-showall` mode takes a very long time.

The following is an example of running OracleAS Portal Diagnostics Assistant on a Unix platform:

```
# Set the environment
#
setenv ORACLE_HOME /oracle/productsAS
#
# Run PDA
#
pda.csh \
-schema portal  \
-password <portal_password>  \
-connect abc.oracle.com:1521:orcl1 \
-ssoSchema orasso \
-ssoPassword <orasso_password> \
-ssoConnect defg.oracle.com:1521:orcl2
-show
```

Run OracleAS Portal Diagnostics Assistant on Windows as follows:

a. Start up a command prompt, and run the following command:

```
pda.cmd
-schema <portal schema name>
-password <portal schema password>
-connect <Portal connect string>
-ssoSchema <SSO schema name>
-ssoPassword <SSO schema password>
-ssoConnect <SSO connect string>
[-apacheLogDir <directory name>]
[-apacheLogName <file name>]
[-logFileLimit <number of rows>]
[-show]
[-showall]
```

Run the script without any parameters to get help information.

Table K–11 lists and describes the parameters used with the OracleAS Portal Diagnostics Assistant script.

5. Open the latest HTML report (`pda.htm`) in a browser window and use the information to help diagnose any OracleAS Portal issues.

K.2.6 Verifying the Portal Dependency Settings File

When troubleshooting OracleAS Portal, one of the first things to do is to review the contents of the Portal Dependency Settings file called `iasconfig.xml`. This file stores configuration data from all the dependent components in a central place, and the contents of this file are updated when there are configuration changes. Therefore, the file should reflect the current configuration of OracleAS Portal with OracleAS Web Cache, Oracle Internet Directory, and Oracle Enterprise Manager 10*g*. If the file does not accurately reflect your configuration settings, then you must update the file and run the Portal Dependency Settings tool `ptlconfig` to update OracleAS Metadata Repository.

Refer to Appendix A, "Using the Portal Dependency Settings Tool and File" for more information about the Portal Dependency Settings file, and examples of the `iasconfig.xml` file.

K.2.7 Analyzing Mobile-Related Problems in OracleAS Portal

Mobile devices do not provide good interfaces for displaying detailed error information when compared with standard desktop browsers. The following information will help you analyze mobile-related problems in OracleAS Portal.

Using the _debug parameter

All OracleAS Portal pages can be run in a special mode where timing and caching information is displayed. If you append the `_debug=1` parameter to a page URL, then extra timing information is added to the response that is displayed.

If you want to see the debug information for a few select pages and portlets, you can control the logging level by using the `_debug` URL parameter. Valid values for `_debug` are `0`, `1`, `2`, `3`, `4`, and `5`. For details about timing and caching statistics, refer to Section C.7, "Timing and Caching Statistics".

You may encounter problems in using the `_debug` URL parameter for mobile browser access because of the following reasons:

- The URL that the mobile device uses to access OracleAS Portal refers to an OracleAS Wireless service and not OracleAS Portal. Therefore, you cannot directly append the `_debug=1` parameter to the URL in the mobile browser.
- The method used for rendering information for a mobile device requires the response page to be valid OracleAS Wireless XML. Because the extra information may not be valid OracleAS Wireless XML, it cannot be added inline to a mobile response page.

To resolve this problem and to use the `_debug` parameter, perform the following tasks:

1. Create a new service in the OracleAS Wireless server to access a page directly, instead of using the default portal service that is registered with OracleAS Wireless. Specify a URL with a format similar to the following:

   `http://<`*`host.domain`*`>:<`*`port`*`>/portal/pls/portal/MyGroup/MyPage?_debug=1`

2. Request the new service not the default portal service in the mobile device.

3. View the servlet log file for the recorded performance information. This information will be in a format similar to the following:

```
4/23/02 5:38 AM portal: [perf] Information for Portlet 33,31071.
Portlet Timing: 234 msecs (wait=0)
Timing Status:
XSLT Timing: null msecs
Caching information of portlet:
Portlet Cache status:  <I>Web Cache:-</I> MISS,NON-CACHEABLE [N], <I>File
System Cache:-</I> MISS,NEW
From Cache:  <I>Web Cache:-</I> -,  <I>File System Cache:-</I> None
From Portlet: Cache Key: NORMAL, Cache Level: USER

4/23/02 5:38 AM portal: [perf] Information for Page 33%2C31060%2C33_31062
Elapsed Time: 2470 msecs
Page meta-time 7 msecs (wait = 994)
Page meta Cache Status:  <I>Web Cache:-</I> MISS,NON-CACHEABLE [N], Cache
Expires: null secs, Age in Cache: null secs ,  <I>File System Cache:-</I>
MISS,NEW
Login meta-time 1227 msecs (wait = 1)
Login meta Cache Status:  <I>Web Cache:-</I> MISS,NON-CACHEABLE [N], Cache
Expires: null secs, Age in Cache: null secs
```

Mobile Information Useful for Support

If you are not able to resolve mobile-related problems by using the troubleshooting steps described in Section K.1.22, "Problems in Accessing OracleAS Portal from a Mobile Device", then contact Oracle Support Services. It would be helpful if you have answers to the following questions before you contact Oracle Support Services:

1. What are the symptoms of the error? For example, did you get error messages, was there a lack of response, or was a blank screen displayed?
2. What is the context of the error?
   - Was the user logged on?
     - Do all authenticated users experience the same problem?
     - Does the public user experience the problem?
   - Does the problem occur during the logon phase?
     - How far did the user get in logging on?
     - Did the user try to log on in the standard manner, or was the user directed to log on?
   - Does the problem occur when viewing a page?
     - Describe the page structure.
     - Do any portlets allow titles to be personalized?
     - Can the page be previewed using a standard desktop browser without having OracleAS Wireless in the communication network?
   - Does the problem occur when viewing an individual portlet?
   - Does the problem occur on all mobile pages, a few pages, or one page?
3. If possible, run the portal service through the OracleAS Wireless debug tool. This requires specific OracleAS Wireless access. For details, refer to the *Oracle Application Server Wireless Administrator's Guide*.
4. Is there a record for the problem in any of the log files listed in Table K–12?

Table K–12 Error Log Files and Locations

Log Files	Location
OracleAS Wireless Server log files	*ORACLE_HOME*/wireless/logs/sys_panama.log or /var/tmp/sys_panama.log Server JVM standard output: *ORACLE_HOME*/opmn/logs/OC4J_Wireless_default_island/application.log Provider JVM standard output: *ORACLE_HOME*/j2ee/OC4J_wireless/application-deployments/portal/OC4J_Portal_default_island/application.log
Oracle HTTP Server log files	*ORACLE_HOME*/Apache/Apache/logs/access_log/application.log and ORACLE_HOME/Apache/Apache/logs/error_log
Parallel Page Engine Server log file	*ORACLE_HOME*/j2ee/OC4J_Portal/application-deployments/portal/OC4J_Portal_default_island_1/application.log

K.3 Need More Help?

You can find more solutions on Oracle *MetaLink* at

http://metalink.oracle.com

If you do not find a solution for your problem, then log a service request.

To help Oracle Support Services troubleshoot the problem, perform the following steps:

1. Run OracleAS Portal Diagnostics Assistant.

   You can diagnose portal-related issues by reviewing the report generated by OracleAS Portal Diagnostics Assistant. You can also refer to Section K.2.5, "Using OracleAS Portal Diagnostics Assistant" for more information.

2. Contact Oracle Support Services.

   If you cannot establish why your portal is not accessible, then contact Oracle Support Services. To help Oracle Support Services troubleshoot the problem, provide the following information:

   - ZIP file generated by OracleAS Portal Diagnostics Assistant.
   - Details of any command-line scripts that you have run (for example, `ptlconfig`) including all the parameters used.
   - A rough network diagram, showing how your Oracle Application Server components are configured.

   See Also:

   - *Oracle Application Server Release Notes*, available on OTN http://www.oracle.com/technology/documentation/index.html
   - *Oracle Application Server Portal Error Messages Guide*

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X