

Oracle9iAS Portal

Configuration Guide

Release 3.0.9

May 2001

Part No. A90096-01

ORACLE[®]

Part No. A90096-01

Copyright © 2001, Oracle Corporation. All rights reserved.

Contributors: Paul Encarnacion, Todd Vender, Susan Léveill , Susan Barton, Steve Buxton, Mark Clark, Rajiv Chopra, Demetris Christou, Pushkar Kapasi, Jeanne Kramer, William S. Lankenau, Eric Lee, Mark Loper, Sunil Marya, Frank Rovitto, Jerry Silver, Cheryl K. Smith, Rich Soule, Arun Arat Tharakkal, Dawn Tyler.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

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

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle is a registered trademark, and Oracle7, Oracle8, Oracle8i, Oracle9i, Oracle *interMedia*, PL/SQL, and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

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

This product includes software developed by the OpenSSL project for use in the OpenSSL Toolkit (<http://www.openssl.org/>). This product includes cryptographic software written by Eric Young (ey@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

This product includes software developed by Ralf S. Engelschall (rse@engelschall.com) for use in the `mod_ssl` project (<http://www.modssl.org/>).

This product includes the Perspective for Java software licensed from Three D Graphics, Inc. Copyright 2000 Three D Graphics, Inc.

Table of Contents

Send Us Your Comments	xi
Preface	xiii
1 Verifying Requirements	
1.1 System Requirements	1-3
1.1.1 Operating Systems	1-3
1.1.2 Oracle Databases	1-3
1.1.3 Oracle Home	1-4
1.1.4 Web Browsers	1-4
1.1.5 Tablespace Requirements	1-4
1.1.6 init.ora Settings	1-5
1.1.7 Terminal Settings	1-5
1.1.8 <i>interMedia</i> Text Requirements	1-5
1.1.9 Unix Shell Settings.....	1-7
2 Post-installation and Configuration Tasks	
2.1 Oracle Portal Default Schemas	2-2
2.2 Oracle Portal Default Accounts.....	2-3
2.3 Oracle Portal Default Groups.....	2-4
2.4 Accessing Oracle Portal in Your Browser	2-5
2.4.1 Simplifying the Full URL of an Oracle Portal Installation.....	2-6
2.5 Manually Starting JServ	2-7
2.6 Installing Language Support in Oracle Portal	2-7
2.6.1 Installing Support for a Language.....	2-8
2.6.2 Installing Graphic Support for BiDi Languages	2-10
2.7 Deinstalling Oracle Portal	2-10
2.7.1 Deinstalling a Single Oracle Portal Schema or the Login Server	2-11
2.8 Configuring Self-registration	2-12

2.9	Enabling Oracle Portal Beta Features.....	2-14
2.10	What are Certificates?	2-14
2.10.1	What are Signature and Chain Files?	2-15
2.10.2	Securing Ports to Use Certificates and HTTPS.....	2-16
2.10.3	Configuring Oracle Portal to Use HTTPS.....	2-18
2.10.4	Adding Certificate Entries in httpd.conf	2-18
2.11	Enabling Secure Socket Layer (SSL).....	2-20
2.11.1	Configuring SSL on Oracle Portal and the Login Server.....	2-21
2.11.2	Setting Login Server Query Path URL	2-22
2.11.3	Adding SSO Enabler Configuration Entries for HTTPS Mode	2-23
2.11.4	Configuring HTTPS with Microsoft Internet Explorer.....	2-24
2.11.5	Configuring HTTPS with Virtual Hosts.....	2-24
2.12	Configuring the Reports Security Portlet	2-25

3 Basic Oracle Portal Configurations

3.1	Configuring Oracle Portal on a Standalone Laptop	3-1
3.2	Configuring Oracle Portal as an Integrated Server.....	3-3
3.3	Configuring a Traditional Three-tier Architecture	3-4
3.3.1	Configuring Distinct Oracle Portal and Login Server Instances.....	3-6

4 Distributed Oracle Portal Installations

4.1	What is a Node?.....	4-2
4.2	Benefits of a Distributed Oracle Portal Environment.....	4-3
4.2.1	Portlet Provider Information Shared Across Nodes	4-3
4.2.2	Scalable Solutions	4-3
4.3	Node Requirements	4-4
4.3.1	Common Cookie Domain.....	4-5
4.3.2	Oracle HTTP Server powered by Apache Configuration	4-5
4.3.3	Common Cookie Name	4-6
4.3.4	Common Login Server	4-6
4.3.5	Symmetric Node Registration	4-6
4.3.6	URLs in Portlets	4-7
4.4	Configuring a Distributed Oracle Portal Environment.....	4-7
4.4.1	Step 1: Create Oracle Portal Nodes.....	4-8
4.4.2	Step 2: Create Same Cookie Domain.....	4-8

4.4.3	Step 3: Edit Oracle Portal DADs	4-10
4.4.4	Step 4: Associate Nodes with the Same Login Server.....	4-11
4.4.5	Step 5: Create a User on the Login Server with Administrator Privileges	4-14
4.4.6	Step 6: Discover the Name of Each Node.....	4-14
4.4.7	Step 7: Register Nodes Between Themselves.....	4-15
4.4.8	Step 8: Refresh the Portlet Repository for Each Node.....	4-16
4.4.9	Step 9: Create Additional Nodes.....	4-17

5 Oracle9iAS Middle-tier and Firewall Configuration

5.1	Oracle Portal as a Partner Application	5-1
5.1.1	Associating Portal to an Existing Login Server That You Have No Control Over	5-2
5.1.2	Associating Portal to an Existing Login Server That You Have Control Over	5-4
5.1.3	Login Server Configured with Oracle Internet Directory (OID)	5-5
5.2	Configuring Virtual Hosts.....	5-6
5.3	Parallel Page Engine Configuration	5-8
5.3.1	Configuring Parallel Page Engine Parameters	5-9
5.4	Working with Firewalls and Load Balancers.....	5-10
5.4.1	Configuring Reverse Proxy Servers Over the Internet	5-10
5.4.2	Configuring Oracle HTTP Server	5-13
5.4.3	Resolving Domain Names	5-15
5.5	Configuring Load Balancing Routers	5-16
5.5.1	Placing a Firewall Between the Middle-tier and the Database	5-18
5.6	Tuning the Oracle HTTP Server	5-18
5.6.1	Configuring the MaxClient Setting.....	5-19

6 Setting up the Search Feature in Oracle Portal Content Areas

6.1	New Search Features	6-2
6.2	Prerequisites	6-2
6.2.1	<i>interMedia</i> Text on UNIX.....	6-3
6.3	Searching in Oracle Portal.....	6-3
6.3.1	Basic Search.....	6-4
6.3.2	Advanced Search.....	6-5
6.3.3	<i>interMedia</i> Text Search.....	6-6
6.3.4	Viewing <i>interMedia</i> Text Search Results	6-7
6.4	<i>interMedia</i> Text Performance	6-7

6.4.1	Query Considerations	6-7
6.4.2	Indexing Considerations.....	6-8
6.4.3	Update Considerations	6-9
6.5	Setting up <i>interMedia</i> Text Searching.....	6-10
6.5.1	Step 1: Set up the Global Page Settings	6-10
6.5.2	Step 2: Enable <i>interMedia</i> Text Searching	6-11
6.5.3	Step 3: Create an <i>interMedia</i> Text Index	6-13
6.5.4	Step 4: Maintain an <i>interMedia</i> Text Index	6-15
6.6	Setting up <i>interMedia</i> Text Indexes	6-17
6.7	Dropping an <i>interMedia</i> Text Index.....	6-17
6.8	Setting up Your Environment for <i>interMedia</i> Text.....	6-18
6.8.1	listener.ora.....	6-18
6.8.2	tnsnames.ora.....	6-19
6.9	Multilingual Functionality (Multilexer)	6-20
6.10	<i>interMedia</i> Text-related Procedures Created in Oracle Portal	6-20

7 Building Oracle Portal Reports and Forms with Rich Content

7.1	<i>interMedia</i> Object Types.....	7-1
7.2	Rich Content in Oracle Portal Reports	7-2
7.2.1	Object Attribute Display	7-3
7.2.2	Display Options.....	7-3
7.2.3	Building the Report	7-4
7.2.4	QBE Reports - Parameter Entry Form	7-6
7.3	Rich Content in Oracle Portal Forms	7-6
7.3.1	Building the Form	7-7
7.4	Browsing Tables with the Oracle Portal Navigator.....	7-9
7.5	Known Issues.....	7-9

8 Troubleshooting

8.1	Verify System Requirements.....	8-1
8.1.1	Check Installation Log	8-1
8.2	Identify the Component Causing the Problem	8-2
8.2.1	Location of Apache Log Files.....	8-2
8.3	Troubleshooting Connection Problems with the Diagnostics Tool.....	8-3
8.3.1	Problems Detected by the Diagnostics Tool	8-3

8.3.2	Problems Not Detected by the Diagnostics Tool	8-4
8.3.3	Running the Diagnostics Tool.....	8-4
8.3.4	Sample Diagnostics Report	8-5
8.4	Configuration Control Points and File Locations	8-7
8.4.1	Oracle Portal Installation Directory Name Change	8-8
8.5	Troubleshooting Oracle Portal	8-8
8.5.1	Installation and Configuration Problems	8-10
8.5.2	Problems Logging on to Oracle Portal.....	8-18
8.5.3	Problems Running Oracle Portal.....	8-21
8.5.4	Miscellaneous Issues Using Oracle Portal	8-26

A Oracle9i Application Server Configuration Files

A.1	Control Points and File Locations.....	A-1
A.1.1	Oracle HTTP Server Configuration File (httpd.conf)	A-2
A.1.2	JServ Configuration File (zone.properties)	A-2
A.1.3	Database Access Descriptor (DAD) Configuration File (wdbsvr.app).....	A-3
A.1.4	Oracle Database Connection File (tnsnames.ora)	A-5
A.1.5	Login Server Configuration Table	A-5
A.1.6	Login Server's Partner Application Table.....	A-7
A.1.7	Local HOSTS File.....	A-7

B Oracle Portal Installation and Configuration Scripts

B.1	Overview	B-2
B.2	Manually Installing Oracle Portal with the wininstall Script.....	B-3
B.2.1	Manually Installing Online Help.....	B-9
B.3	Manually Installing a Login Server with the linstall Script	B-10
B.4	Configuring a New Oracle Portal Instance and Login Server with the ssodatan Script	B-11
B.5	Updating an Existing Portal Instance with the ssodatax Script.....	B-14
B.6	Modifying the Scope of the Portal Session Cookie.....	B-17
B.7	Managing the Session Cleanup Job.....	B-18

Index

List of Figures

1-1	Oracle9i Application Server services and components.....	1-2
2-1	Login Server Query Path URL Prefix field	2-23
3-1	Integrated server configuration.....	3-4
3-2	Traditional three-tier architecture	3-5
3-3	Three-tier architecture with distinct Oracle Portal and Login Server.....	3-6
4-1	Distributed Oracle Portal installations topology	4-2
4-2	Displaying portlets created on different nodes on a single page	4-4
5-1	Oracle Portal as a Partner Application	5-2
5-2	Login Server configured with Oracle Internet Directory (OID)	5-6
5-3	Internet configuration with reverse proxy server.....	5-11
5-4	Example of reverse proxy server configuration.....	5-13
5-5	Load balancing router configuration.....	5-17
6-1	Basic search screen in Oracle Portal.....	6-5
6-2	Advanced search screen in Oracle Portal.....	6-6
6-3	Global page settings - Proxy Server	6-11
6-4	Services portlet - interMedia Text properties.....	6-12
7-1	Application component links on the Navigator	7-4
7-2	Oracle Portal Reports Wizard - common options.....	7-5
7-3	Oracle Portal Reports example with interMedia rich content	7-5
7-4	Portal Access tab - Publish to Portal	7-6
7-5	Oracle Portal Form - Item Level option.....	7-8
7-6	Oracle Portal Form example with interMedia rich content.....	7-8

List of Tables

2-1	Default Oracle Portal schemas created.....	2-2
2-2	Default Oracle Portal accounts created.....	2-3
2-3	Default Oracle Portal groups created.....	2-4
2-4	URL to enter in browser to access Oracle Portal.....	2-5
2-5	Supported languages and abbreviations	2-7
2-6	Language script parameters (langinst).....	2-9
2-7	Self-registration parameter descriptions.....	2-13
2-8	Certificate types	2-15
2-9	Certificate files and locations.....	2-16
2-10	zone.properties file location.....	2-16
2-11	Certificate Entries in the Oracle HTTP Server configuration file.....	2-18
4-1	Partner Application Configuration Example	4-12
4-2	Login Server Create New User Example	4-14
4-3	Node a to node b registration information.....	4-15
4-4	Creating additional nodes for distributed environment	4-17
5-1	Parallel Page Engine (PPE) parameters.....	5-9
6-1	<i>interMedia</i> Text search types	6-4
6-2	<i>interMedia</i> Text indexes created	6-14
7-1	<i>interMedia</i> object types	7-2
7-2	Oracle Portal Report wizards	7-2
8-1	Apache log file locations	8-2
8-2	Diagnostics tool's diag parameters.....	8-5
8-3	Oracle Portal installation directory structure.....	8-8
8-4	List of installation and configuration problems.....	8-8
8-5	List of problems logging on to Oracle Portal	8-9
8-6	List of problems running Oracle Portal	8-9
8-7	List of miscellaneous issues using Oracle Portal	8-10
A-1	List of Oracle Portal Control Points.....	A-2
A-2	Database Access Descriptor (DAD) configuration parameters	A-4
B-1	Oracle Portal installation scripts	B-2
B-2	linstall script parameters	B-10
B-3	ssodatan script parameters	B-12
B-4	ssodatax script parameters.....	B-15
B-5	ctxjsub parameters.....	B-20

Send Us Your Comments

Oracle9iAS Portal Configuration Guide, Release 3.0.9

Part No. A90096-01

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

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

If you find any errors or have any other suggestions for improvement, please indicate the part number, chapter, section, and page number (if available). You can send comments to us by electronic mail to webdbdoc_us@oracle.com.

If you have problems with the software, please contact your local Oracle World Wide Support Center.

Preface

This guide provides information about configuring Oracle9iAS Portal.

Note: This guide uses *Oracle Portal* when referring to Oracle9iAS Portal.

This preface includes the following sections:

- [Intended Audience](#)
- [Structure](#)
- [Conventions](#)
- [Documentation](#)
- [Documentation Accessibility](#)
- [Oracle Services and Support](#)

Intended Audience

This guide is intended for users who are responsible for configuring and maintaining Oracle Portal.

Structure

This guide is comprised of the following:

Chapter	Contents
Chapter 1, "Verifying Requirements"	Provides information about the Oracle Portal system requirements.
Chapter 2, "Post-installation and Configuration Tasks"	Provides general post-installation information including how to access Oracle Portal in your browser, installing and changing the language of Oracle Portal in your browser, and descriptions of the default Oracle Portal schemas, accounts, and groups created upon installation. Deinstallation information is also provided.
Chapter 3, "Basic Oracle Portal Configurations"	Provides various topologies that you can set up with Oracle Portal including standalone laptop, integrated server, and traditional three-tier.
Chapter 4, "Distributed Oracle Portal Installations"	Provides instructions on running Oracle Portal in a distributed environment to enable portlet providers to be shared across nodes and for increased scalability, availability, and performance.
Chapter 5, "Oracle9iAS Middle-tier and Firewall Configuration"	Provides instructions on how to configure a separate Login Server listener from the Oracle Portal listener.
Chapter 6, "Setting up the Search Feature in Oracle Portal Content Areas"	Provides instructions on setting up <i>interMedia</i> Text to perform text searching in content areas created with Oracle Portal.
Chapter 7, "Building Oracle Portal Reports and Forms with Rich Content"	Provides instructions on including rich content such as graphics, audio, images, and video into Oracle Portal Reports and Oracle Portal Forms.
Chapter 8, "Troubleshooting"	Provides solutions to problems you may encounter while installing or using Oracle Portal.
Appendix A, "Oracle9i Application Server Configuration Files"	Provides information about the configuration files which can affect the connection to and the behavior of the Oracle9i Application Server and its components in the middle tier as well as on other machines to which it is connecting.

Chapter	Contents
Appendix B, "Oracle Portal Installation and Configuration Scripts"	Provides information about various scripts that are used for customizing the configuration. In addition, information about installing additional components, such as a standalone Login Server, additional Oracle Portal nodes, loading additional language translations into Oracle Portal, and import and export utilities, are provided.

Conventions

The following typographical conventions are used in this guide:

Convention	Meaning
boldface	Used for emphasis. Also used for button names, labels, links, and other user interface elements.
<i>italics</i>	Used to introduce new terms. Also used enclosed in angled brackets (<>) to represent a variable. Substitute an appropriate value.
<code>courier</code>	Used to represent text you need to type. Also used for file names and directories.
<ORACLE_HOME>	Refers to the location of the Oracle9i Application Server installation files, including those for the Oracle Portal component.

Documentation

The Oracle Portal Online Help Content Area is an online help system which provides detailed step-by-step instructions and reference information, as well as an introduction to Oracle Portal and troubleshooting information.

This guide refers you to various "topics" included in the online help system.

Note: To perform full text searching in the online help system, *interMedia Text* must be configured. See [Chapter 6, "Setting up the Search Feature in Oracle Portal Content Areas"](#) for more information.

Oracle Portal Publications

You can also refer to the following publications which are available from the Oracle Technology Network at:

<http://technet.oracle.com>

Part Number	Title	Description
A90095-01	<i>Release Notes</i>	Describes last minute changes to the product or documentation.
A90097-01	<i>Tutorial</i>	Provides step-by-step lessons that teach you the basics of Oracle Portal.
A90098-01	<i>Building Advanced Portals</i>	Provides several cases that show you how to use Oracle Portal's advanced features.
N/A	<i>Oracle Portal Development Kit</i>	Provides detailed information about the Oracle Portal API set as well as numerous examples that demonstrate API implementation.

Documentation Accessibility

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

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

Oracle Services and Support

A wide range of information about Oracle products and services is available at:

<http://www.oracle.com>

Oracle Support Services

Technical Support contact information worldwide is listed at:

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

Product and Documentation

For U.S.A customers, the Oracle Store is at:

<http://store.oracle.com>

Links to Stores in other countries are provided from this site.

Product documentation can be found at:

<http://docs.oracle.com>

Education and Training

Training information and worldwide schedules are available from:

<http://education.oracle.com>

Oracle Technology Network

Register with the Oracle Technology Network (OTN) at:

<http://technet.oracle.com>

OTN delivers technical papers, code samples, product documentation, and self-service developer support.

Verifying Requirements

Before installing Oracle Portal, make sure that your system meets the necessary requirements which are presented in this chapter.

As part of a typical Oracle Portal installation, the following components are installed:

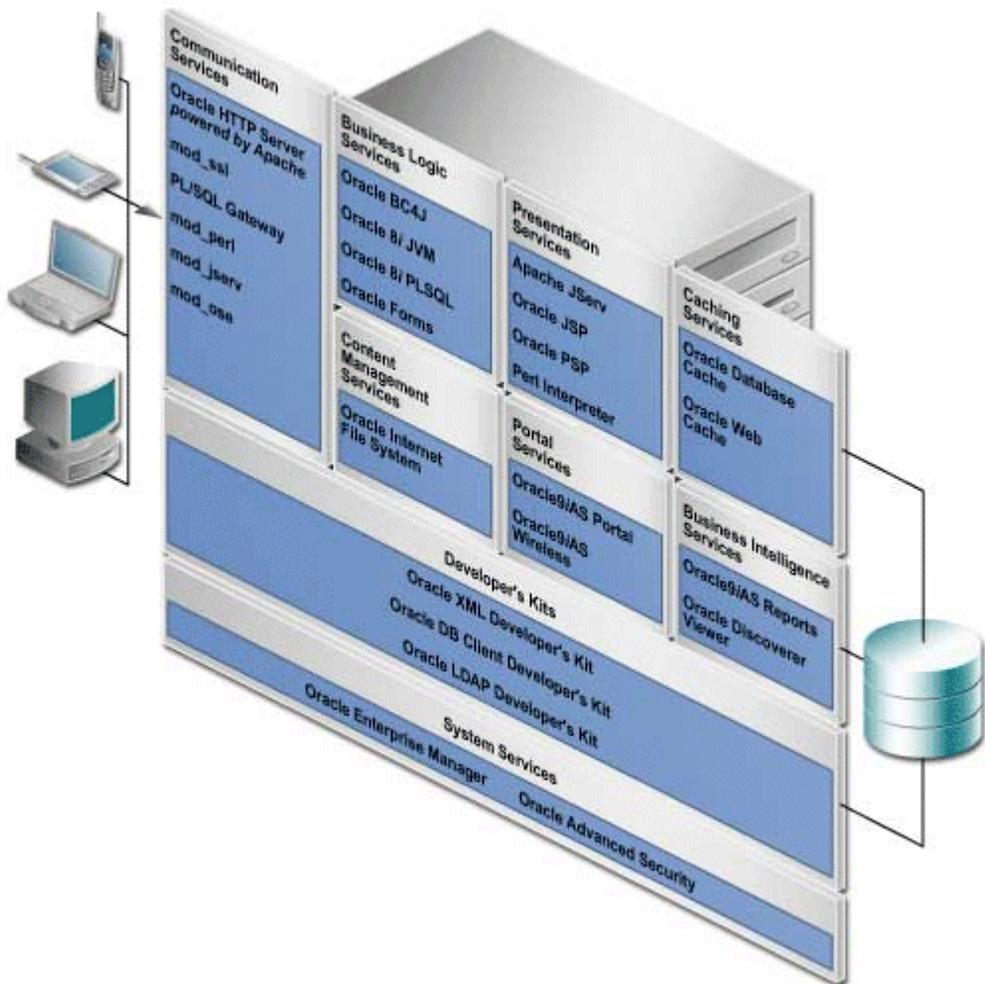
- Oracle Portal Database Objects: images, demos, content areas, common services, utilities, required support files, documentation, and Single Sign-On (SSO).
- Oracle9i Application Server Web Server: Servlets, Oracle PL/SQL Gateway (including mod_plsql), and the Oracle HTTP Server *powered by Apache*.

See also:

- Oracle HTTP Server documentation in the Oracle9i Application Server Documentation Library for detailed information about Oracle HTTP modules (mods) and the overall Oracle9i Application Server architecture
 - *Oracle9i Application Server Installation Guide* for your particular operating system
-
-

The following is an illustration of the various components packaged with the Oracle9i Application Server. Oracle Portal is packaged under Portal Services.

Figure 1-1 Oracle9i Application Server services and components



1.1 System Requirements

Before you install Oracle Portal, you must have an Oracle8*i* database and Oracle9*i* Application Server running, including the Oracle HTTP Server.

Oracle Portal is packaged and installed with the Oracle9*i* Application Server, in the same Oracle Home location, but in a separate Oracle Home location from the Oracle8*i* database.

Ensure that your system also meets the following requirements before you install Oracle Portal.

1.1.1 Operating Systems

Refer to the *Oracle9*i* Application Server Installation Guide* for details.

1.1.2 Oracle Databases

Oracle Portal can be installed in the following databases:

- Oracle8*i* Release 8.1.6.2.0 Enterprise Edition
- Oracle8*i* Release 8.1.7 Standard and Enterprise Edition

See the "Administration" section of the *Oracle9*i*AS Portal Release Notes* for more information on the required database patches.

Notes: Oracle 8.1.6.2 is required since Oracle Portal makes extensive use of Oracle8*i* PL/SQL features like Native Dynamic SQL, Bulk Collect, Autonomous Transactions, nested tables and objects. Additionally, some of the newer features are exposed through Java.

If you are planning to enable *interMedia* Text in Oracle Portal on Windows NT/2000, you must have an Oracle 8.1.7 database.

Also, verify that the following conditions exist on your database before installing Oracle Portal:

- Oracle JServer Option, which includes the Enterprise JavaBeans and CORBA Tools, Java Virtual Machine, and Oracle Java Tools, are installed.
- You have the SYS user password on your database.
- Your database is up and running.

1.1.3 Oracle Home

You *must* install Oracle9i Application Server and its component, Oracle Portal, in a *separate* Oracle home directory from your Oracle8i database.

1.1.4 Web Browsers

- Netscape 4.0.8 and 4.72
- Microsoft Internet Explorer 4.0.1 with Service Pack 1 and 5.0.1

Note: You may encounter JavaScript errors if you use a browser older than the recommended minimum.

1.1.5 Tablespace Requirements

Since Oracle Portal is installed with Oracle9i Application Server, the tablespace amounts are calculated as a whole for all Oracle9iAS components. Refer to the *Oracle9i Application Server Installation Guide* for your particular platform for the documented tablespace requirements.

Notes:

- Before installing Oracle Portal, it is a good idea to increase the sizes for your tablespaces, tables, and indexes to shorten the installation time and improve performance. See [Chapter 8, "Troubleshooting"](#) for more information.
 - If you are using a FAT file system in your Windows NT/2000 environment, double the amount for the default tablespace (for example, 300 MB).
 - During installation, you are prompted to select a tablespace from the database in which to install the Oracle Portal schema and database objects. The default tablespace set by the Oracle Portal Configuration Assistant is USERS.
 - Do not allocate additional tablespace for each content area that you create since content areas are part of the Oracle Portal schema.
-
-

1.1.6 `init.ora` Settings

Set the `init.ora` parameters in the Oracle Home containing your Oracle Portal installation as follows:

```
MAX_ENABLED_ROLES = 25
OPEN_CURSORS = 50
COMPATIBLE = 8.1.0
JAVA_POOL_SIZE = 20971520
SHARED_POOL_SIZE = 31457280
```

Notes:

- The preceding `init.ora` example specifies the minimum values required with the exception of `COMPATIBLE` which indicates compatibility with an Oracle8i database Release 8.1.0 or above.
 - If you are encountering error number 503 in the Oracle HTTP Server, it could be that the database is not configured with enough sessions to handle the load. Try doubling the number of database processes configured in the `init.ora` file.
-
-

1.1.7 Terminal Settings

On UNIX, set your terminal (using the `set TERM` command) to one of the following types:

```
3151 386 386s 386u 386x
ansi avx3
dec dgd2 dgd4
hft hftc hp iris ncd
sun sun5 tandm
vt100 vt220
wy150 wy50
xsun xsun5
```

1.1.8 *interMedia* Text Requirements

To enable *interMedia* Text searching in Oracle Portal, make sure that the following requirements are met:

- Install and configure the Oracle8i database with the *interMedia* Text option by running the Oracle Universal Installer (OUI) and the Database Configuration

Assistant. See the appropriate Oracle8*i* installation guide for more information about installing and configuring an Oracle8*i* database to use *interMedia* Text.

Notes: If you are planning to enable *interMedia* Text in Oracle Portal on Windows NT/2000, the following requirements apply:

- Oracle Portal must be installed in an Oracle 8.1.7 database.
- Disable connection pooling from the Database Access Descriptor page. For more information, see *Using the PL/SQL Gateway* which is provided as part of the Oracle9*i* Application Server documentation set.

interMedia Text should be installed in the same Oracle Home as your database. It is *not* installed if you perform a Minimal installation of the Oracle8*i* database.

- Install Oracle Portal according to the instructions in the *Oracle9i Application Server Installation Guide*.
- Set the LD_LIBRARY_PATH to the following before starting your database and Net8 Listener:

```
LD_LIBRARY_PATH:$ORACLE_HOME/ctx/lib
```

See also:

- [Chapter 6, "Setting up the Search Feature in Oracle Portal Content Areas"](#)
 - *Net8 Administrator's Guide* for more information about creating the `tnsnames.ora` and `listener.ora` configuration files
-
-

If you encounter any problems during installation, check the installation session log that describes the actions performed and the components created upon installation. The log file is located in:

```
ORACLE_HOME>/assistants/opca/install.log
```

See Also: [Chapter 8, "Troubleshooting"](#)

1.1.9 Unix Shell Settings

For Unix, you should use `csh` or `ksh` for installing Oracle Portal. The Oracle Portal Configuration Assistant uses various Oracle utilities which may not work properly in other shells. The Oracle Portal Configuration Assistant invokes sub-shells to run various utilities, including SQL*Plus, SQL*Loader, `import`, and `loadjava`. You should disable the sourcing to other Oracle homes in the shells before installing Oracle Portal. For example, if your shell sources to an old Oracle home by default, the Oracle Portal Configuration Assistant when invoked creates a sub-shell that sources to a different Oracle home. This causes the utilities to fail.

Post-installation and Configuration Tasks

This chapter provides information about Oracle Portal after installation and the configuration tasks that you can perform.

See: For information about installing Oracle Portal with Oracle9i Application Server, see the *Oracle9i Application Server Installation Guide*.

Specific topics covered include:

- [Oracle Portal Default Schemas](#)
- [Oracle Portal Default Accounts](#)
- [Oracle Portal Default Groups](#)
- [Accessing Oracle Portal in Your Browser](#)
- [Manually Starting JServ](#)
- [Installing Language Support in Oracle Portal](#)
- [Deinstalling Oracle Portal](#)
- [Configuring Self-registration](#)
- [Enabling Oracle Portal Beta Features](#)
- [What are Certificates?](#)
- [Enabling Secure Socket Layer \(SSL\)](#)
- [Configuring the Reports Security Portlet](#)

2.1 Oracle Portal Default Schemas

Oracle Portal is installed primarily in the Oracle8i database, with some supporting components installed on the middle-tier in Oracle9i Application Server.

If you install Oracle Portal in the default mode, six schemas are created. The default base schema name is *portal30* which you can change at installation time.

Important: For security reasons, it is highly recommended that the default passwords for the *portal30*, *portal30_sso*, and *portal30_sso_ps* schemas be changed immediately. These passwords should be changed by the database administrator using the "alter user <schema> identified by <password>" command.

When you change the password for *portal30* or *portal30_sso*, you also need to update the password accordingly in the respective DADs. When you change the *portal30_sso_ps* password, and your login server is on a database instance which is separate from your portal instance, you need to rerun *ssodatax*, specifying the new password so that the database link information is updated.

Table 2–1 Default Oracle Portal schemas created

Schema	Description
<i>portal30</i>	The product schema for Oracle Portal and contains the installed portal database objects.
<i>portal30_public</i>	The schema that the portal users map to when executing procedures in the Oracle Portal product schema. The schema name is constructed from the base schema with "_public" appended to it.
<i>portal30_sso</i>	The product schema for the Login Server. This schema can be renamed in the installer. If not specified, it defaults to the base schema name with "_sso" appended to it.
<i>portal30_sso_ps</i>	The schema on the Login Server database instance used to access the password store for external applications. This schema is named by appending "_ps" to the Login Server schema name. The Portal accesses the Login Server's password store for external applications through this schema, either directly, if it resides on the same database instance, or over a dblink, if the Login Server and Portal are on different database instances.

Table 2–1 Default Oracle Portal schemas created

Schema	Description
<i>portal30_sso_public</i>	The schema that the portal users map to when executing procedures in the Login Server product schema. This name is constructed from the Login Server schema name with "_public" appended to it.
<i>portal30_demo</i>	The schema which is installed with the Oracle Portal demonstration code. The name of this schema is the base schema name with "_demo" appended to it.

2.2 Oracle Portal Default Accounts

With each Oracle Portal installation, a default set of login accounts is created. If the product is installed in a schema named *portal30*, the following default accounts are created:

Table 2–2 Default Oracle Portal accounts created

Account	Description
<i>portal30</i>	This account is created for the Database Administrator (DBA) with the highest privileges in Oracle Portal.
<i>portal30_admin</i>	This is the account created for the portal administrator. This account is similar to the DBA account, however, it does not have privileges that provide access to database administration features, such as creating and managing schemas and other database objects.
public	This account is created for public users for unauthenticated sessions. This is the account that all sessions are associated with prior to authentication.
<i>portal30_SSO</i>	This account is created for the Login Server application. Since the Login Server is implemented with significant reuse of Oracle Portal infrastructure code, this user account is created as a result of this reuse.
<i>portal30_sso_admin</i>	This account is an administrative account which is created in conjunction with the Login Server installation. It has Login Server "Full Administrator" privileges, but it does not have any Portal administrative privileges.

Note: For security reasons, change all the passwords for these accounts after initial login. By default, the password is set to the user name. Change the password by logging on to the Login Server and editing the appropriate user accounts and changing their passwords.

See also: Oracle Portal Online Help topics in the *Working with Users* folder.

2.3 Oracle Portal Default Groups

The following groups are created at installation time:

Table 2–3 *Default Oracle Portal groups created*

Group	Description
DBA	This group has the maximum privilege levels in the system. All global privileges are granted to this group. When this group is installed, it has only one member, the user with the name of the product schema, for example, portal30.
PORTAL_ADMINISTRATORS	This group has most of the global privileges, except for the database-related privileges: ANY_SCHEMA/MANAGE and ANY_SHARED_COMPONENT/MANAGE. This group is comprised of the admin user, portal30_admin, and includes the dba group.
PORTAL_DEVELOPERS	This group has privileges to build and manage Oracle Portal components and applications.
PORTLET_PUBLISHERS	This group has the privilege of publishing portlets. Members of this group can create components in the system such as folders, charts, calendars, and so on. This group is initially composed of the portal_administrators group who can then decide which users or groups should be added to this group.
AUTHENTICATED_USERS	All users that log on to Oracle Portal are added to this group. This is a convenient mechanism to allow logged on users to perform privileged actions. Specified privileges are granted to this group and group membership cannot be changed.
RW_ADMINISTRATOR	This group can CREATE, UPDATE, and DELETE registered report definition files, servers, and printer objects.
RW_BASIC_USER	This group can only run a report if they have been given the privilege to run it.

Table 2–3 Default Oracle Portal groups created

Group	Description
RW_DEVELOPER	In addition to the privileges of the RW_POWER_USER and RW_BASIC_USER groups, this group can run commands which show the system environment. This group might be assigned to a developer who needs to perform testing and to retrieve detailed error messages.
RW_POWER_USER	In addition to the privileges of the RW_BASIC_USER group, this group can see more detailed error messages.

See also: Oracle Portal Online Help topics in the *Working With Users* folder.

2.4 Accessing Oracle Portal in Your Browser

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

`http://<hostname>:<portnumber>/pls/<dad>`

Table 2–4 defines the URL to enter in the browser to access Oracle Portal.

Table 2–4 URL to enter in browser to access Oracle Portal

Parameter	Description
hostname	Defines the machine on which you installed Oracle Portal. <ul style="list-style-type: none"> Enter both the hostname and the fully-qualified domain name. For example, enter <i>host.domain.com</i>. This name must also match the <code>ServerName</code> parameter in the Apache configuration file, <code>httpd.conf</code>, located in: <pre><ORACLE_HOME>/Apache/Apache/conf</pre>
portnumber	Defines the port number you specified earlier to access Oracle Portal.
pls	Defines the virtual path and indicates that the request is for a PL/SQL procedure which alerts the Oracle HTTP Server <i>powered by Apache</i> to reroute the request to the PL/SQL Gateway.

Table 2–4 URL to enter in browser to access Oracle Portal

Parameter	Description
dad	Defines the Database Access Descriptor (DAD) you specified earlier for your Oracle Portal installation. The DAD contains information on how to connect to the database.

See also:

- [Section 2.2, "Oracle Portal Default Accounts"](#)
 - *Using the PL/SQL Gateway* included in the Oracle9i Application Server documentation library.
-
-

2.4.1 Simplifying the Full URL of an Oracle Portal Installation

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

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

```
http://<hostname>:<portnumber>/pls/<dad>
```

You can simplify this URL to:

```
http://<hostname>/<redirectpath>
```

1. Open the Oracle HTTP Server configuration file, `http.conf`. By default this file is located in the following directory:

```
<ORACLE_HOME>/Apache/Apache/conf/
```

2. Enter the redirect path as follows:

```
Redirect /<DADnamepath> http://<hostname>:<portnumber>/pls/<dad>
```

For example:

```
Redirect /portalhome http://mysite.oracle.com/pls/portal30
```

In this example, end users can enter the following:

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

to access the full URL which is as follows:

`http://mysite.oracle.com:80/pls/portal30`

This technique also works with any valid path that is appended to the URL. For example, if you want to display the Oracle Portal Online Help Content Area, enter:

`http://mysite.us.oracle.com/portalhome/url/folder/ONLINE_HELP`

See also: Oracle Portal Online Help topic: *What are direct access URLs.*

2.5 Manually Starting JServ

If you use manually started JServ processes in Oracle9i Application Server, then you need to add the Oracle Portal classpaths to the CLASSPATH variable. Manually started JServ processes do not read `wrapper.classpath` entries in the `jserv.properties` file.

2.6 Installing Language Support in Oracle Portal

You can choose to install and display the Oracle Portal user interface in any of the 24 supported languages from your browser. To install support for a given language, run the `wwwseedus.sql` and `langinst.csh` scripts. Once the language is installed you use the Set Language portlet to choose which language Oracle Portal should use.

[Table 2-5](#) lists the supported languages and their abbreviations.

Table 2-5 Supported languages and abbreviations

Language	Abbreviation	Language	Abbreviation
Arabic	ar	Dutch	nl
Czech	cs	Polish	pl
German	d	Portuguese	pt
Danish	dk	Brazilian Portuguese	ptb
Spanish	e	Romanian	ro
Greek	el	Russian	ru
French	f	Swedish	s

Table 2–5 Supported languages and abbreviations

Language	Abbreviation	Language	Abbreviation
Hungarian	hu	Finnish	sf
Italian	i	Slovak	sk
Hebrew	iw	Thai	th
Japanese	ja	Turkish	tr
Korean	ko	Chinese (Simplified)	zhs
Norwegian	n	Chinese (Traditional)	zht

2.6.1 Installing Support for a Language

To install support for a given language in Oracle Portal:

1. Start a command line prompt.
2. Change to the <ORACLE_HOME>/portal30/admin/plsql/nlsres/ctl/us directory.
3. Start SQL*Plus and log in to the database where Oracle Portal is installed.
4. From SQL*Plus, run the `wwwseedus.sql` script against the portal30 schema.
5. Change to the <ORACLE_HOME>/portal30/admin/plsql directory.
6. Enter one of the following commands, depending upon your operating system:

UNIX

```
langinst.csh <-s portal_schema> <-p portal_password> <-o sso_schema> <-d sso_
password> <-c portal_connect_string> -l <language> -available
```

Windows NT/2000

```
langinst.cmd <-s portal_schema> <-p portal_password> <-o sso_schema> <-d sso_
password> <-c portal_connect_string> -l <language> -available
```

Example

```
langinst.csh -s portal30 -p portal30 -o portal30_sso -d portal30_sso -c orcl -l
cs -available
```

Table 2–6 lists the language script parameters for `langinst`.

Table 2–6 Language script parameters (*langinst*)

Parameter	Description
-s portal_schema	The database schema for the Oracle Portal database objects. The default is PORTAL30.
-p portal_password	The Oracle database password for the Oracle Portal schema. The default is <portal_schema>
-o sso_schema	The Oracle database schema for Login Server objects. The default is <portal_schema>_SSO.
-d sso_password	The Oracle database password for Login Server schema. The default is <sso_schema>.
-c connect_string	The connect string for the database in which the Oracle Portal schema is installed. Provide the connect string only if the schema is located on a remote database.
-l language	The abbreviation for the language you want to install. See Table 2–5, "Supported languages and abbreviations" .
-available	Ensures that the tabs are translated. This is required.

7. Repeat Step 6 for each language desired in Oracle Portal.
8. To enable users to choose their desired language, add the Set Language portlet to a portal page. This portlet displays all of the languages currently installed. Users can then select the language of their choice when logging on.

Typically, the Set Language portlet is added to your content area home page, but you can add it to any page. See the Oracle Portal Online Help for instructions on how to add portlets to pages.

Notes:

- In previous releases, Oracle Portal's language support used the setting of your browser's language preference. With the Set Language portlet, this is no longer the case.
 - The Login Server's language is set separately from that of Oracle Portal. The Login Server automatically displays a list of installed languages on its login page, which determines the language used for the Login Server regardless of what you set in the Set Language portlet.
-
-

2.6.2 Installing Graphic Support for BiDi Languages

Arabic and Hebrew are languages which are displayed from right to left on the screen, rather than left to right. These are also known as BiDi (bidirectional) languages. If using these languages, you need to run a script which updates certain Oracle Portal metadata to provide proper graphics support for some of the required images. To install the graphic support for BiDi languages:

1. Locate the following directory:

```
<ORACLE_HOME>/portal30/admin/plsql/nlsres
```

2. On the database where Oracle Portal is installed, log on to SQL*Plus with the appropriate user name and password for that schema. For example:

```
sqlplus portal30/portal30
```

3. Enter the following command to install proper graphic support for BiDi languages:

```
@imginst.sql
```

2.7 Deinstalling Oracle Portal

This section describes how to use Oracle Universal Installer to deinstall Oracle products (which deinstalls them from the Oracle Universal Installer inventory) instead of removing them manually.

See also: *Oracle9i Application Server Installation Guide* for your operating system.

To deinstall Oracle Portal, follow these steps:

1. Launch the Oracle Universal Installer.
 - On UNIX, enter `./runInstaller`
 - On Windows NT/2000, choose Start -> Programs -> Oracle Installation Products -> Universal Installer.

The *Welcome* window for Oracle Universal Installer appears.

2. Click **Deinstall Products**.

The *Inventory* dialog box appears.

3. Expand the tree of installed products until you find the products to deinstall. In this case, choose Oracle Portal.
4. Check the boxes of products to deinstall.
5. Click **Remove**.
The *Inventory Confirmation* window appears.
6. Click **Yes** to deinstall the selected products.

Note: A message may display indicating that removing some products may cause other products to function improperly.

The products are deinstalled from your computer. The *Inventory* dialog box appears without the deinstalled products.

7. Click **Close** to close the *Inventory* dialog box.
8. Click **Exit** to exit Oracle Universal Installer.

2.7.1 Deinstalling a Single Oracle Portal Schema or the Login Server

Deleting a single Oracle Portal schema and/or the Login Server is performed from the Oracle Portal Configuration Assistant.

To deinstall an Oracle Portal schema or the Login Server:

1. Launch the Oracle Portal Configuration Assistant:

Windows NT/2000

Choose Start -> Programs -> Oracle Home -> Oracle Portal Configuration Assistant

UNIX

Go to the `$OH/assistants/opca` directory and run the `launch.sh` script

2. The Step 1 of 6: Installation Options window appears.
Click the option to **Deinstall Oracle Portal or the Login Server**.
3. Follow the instructions on the remaining screens to complete this task.

See also: *Oracle9i Application Server Installation Guide* for your particular operating system

2.8 Configuring Self-registration

If you want to allow users to create new accounts for themselves through a link on the Login Portlet, you do so by installing the self-registration feature as follows:

1. Start SQL*Plus and log in to the database in which the Login Server is installed.
2. Create a schema. This schema is used for accessing user administration objects on the Login Server. In this task this is referred to as the *user administration access schema*.
3. In the user administration access schema, create a synonym for the Login Server SSO schema package `wssso_api_user_admin`. This synonym must be called `wssso_api_user_admin`.
4. Grant Execute privileges on the `wssso_api_user_admin` package in the Login Server SSO schema to the user administration access schema.
5. If Oracle Portal and the Login Server are installed in different databases:
 - a. In the user administration access schema, create a synonym for the Login Server SSO schema package `wwctx_api_vpd` package. This synonym must be called `wwctx_api_vpd`.
 - b. Grant Execute privileges on the `wwctx_api_vpd` package in the Login Server SSO schema to the user administration access schema.
 - c. In the schema in which Oracle Portal is installed, create a database link to connect to the user administration access schema.
6. Start a command line prompt.
7. Change to the `<ORACLE_HOME>/portal30/admin/plsql` directory of the directory in which Oracle Portal is installed.
8. Enter the following command:

UNIX

```
selfreg.csh -s <portal_schema> -p <portal_password> -ua <sso_uadmin_schema> -c  
<portal_connect_string> -dblink <sso_uadmin_dblink>
```

Windows NT/2000

```
selfreg.cmd -s <portal_schema> -p <portal_password> -ua <sso_uadmin_schema> -c
<portal_connect_string> -dblink <sso_uadmin_dblink>
```

Table 2-7 Self-registration parameter descriptions

Parameter	Description
portal_schema	The database schema in which Oracle Portal is installed. The default is PORTAL30.
portal_password	The password for the above schema. The default is <portal_schema>.
sso_uadmin_schema	The user administration access database schema you created in step 1. The default is <portal_schema>_SSO_UA. Note: You do not need to provide a value for this parameter if you specify a database link for the dblink parameter.
portal_connect_string	The connect string for the database in which Oracle Portal is installed. Note: You need to provide the connect string only if you are running the script on a different database.
sso_uadmin_dblink	The name of the database link created in Step 5c. Note: You need to provide the database link only if the Login Server is installed in a different database instance from the Oracle Portal installation. If you do not provide a value for this parameter, it is assumed that the user administration access schema is in the same database instance as Oracle Portal.

Example

```
selfreg.csh -s myportal -p myportal -ua myportal_sso_ua -c orcl -dblink uadmin_
link
```

9. Press the Enter or Return key.
10. In the Services portlet, click **Global Settings**. By default, the Services portlet is located on the Oracle Portal home page's Administer tab.
11. In the Self-Registration Options section, select **Enable Users To Log On Immediately** if you want users to be able to log on to Oracle Portal immediately after they create their own user account using the self-registration feature.

Note: If you do not select this check box, you need to assign the user as an authorized user before he or she is able to log on.

12. To expose the self-registration feature to users, customize the Login portlet to include a self-registration link.

See: Oracle Portal Online Help topic: *Customizing the Login Portlet*.

2.9 Enabling Oracle Portal Beta Features

You can switch on and off Beta features from the Oracle Portal Global Settings page, in the following way:

1. In the **Services** portlet, click **Global Settings**. By default, the **Services** portlet is located on the Oracle Portal home page's **Administer** tab.
2. In the **Beta Features** section:
 - a. Select **Image Charts From Query Wizard** to enable Oracle Portal application developers to create Java-based image charts using a wizard. The wizard to create these charts is accessible from the Application Navigator.
 - b. Select **Data Component** to enable Oracle Portal application developers to create spreadsheets and other data entry tables. After the developer creates an application, one of the components available to add to the application is Data.

2.10 What are Certificates?

Certificates are encrypted files which allow a client and server to pass sensitive data securely. Oracle Portal supports the x.509 certificate standard. This is the type of certificate supplied by most major certificate authorities.

Certificates can be either 40 bit or 128 bit encryption strengths. The greater the number of bits, the more secure the certificates.

[Table 2-8](#) lists the major types of certificates.

Table 2–8 Certificate types

Certificate Type	Description
Global Secure Site ID	This certificate is an extension in a certificate proposed by Verisign. The certificate is used for SSL Server authentication. Using Global Secure Site IDs (formerly called Global Server IDs), companies located anywhere in the world can communicate with 128-bit SSL encryption with their customers. For more information about this technology, visit: http://digitalid.verisign.com/server/global/help/globalFAQ.htm
Secure Site ID	This is a 128-bit certificate which causes the browser to operate at the best encryption level used by the client browser. Thus, if the browser is operating at 40-bit encryption, the server does the same. If the client is at 128-bit, then so is the server. In general this is not a problem since most browsers today operate using 128-bit encryption, however, this is not as secure as the Global Secure Site ID.
40 bit Certificate	This is the least secure type of certificate. In this case, the server and all connected clients operate at a 40-bit encryption level. Note: If you receive a trial certificate from a certificate authority, it is probably this certificate type.

These certificates can be purchased from many different authorities. Oracle Portal currently supports Thawte, Verisign, and Netscape certificate providers.

Note: The Oracle HTTP Server may not understand some v3 extensions that some certificates include. The Oracle HTTP Server should function properly by ignoring those extensions.

2.10.1 What are Signature and Chain Files?

In addition to the certificate, you also need specific signature, and/or chain files from the provider of your certificate. These files are available from your provider's Web site or customer service.

2.10.1.1 Certificate Authority File (CA)

The Certificate Authority (CA) file is the base signature file for the certificate file you have purchased. This file validates the certificate you are using. It informs

clients that they can trust the certificate they have received. You require a CA file for any type of certificate you use.

2.10.1.2 Certificate Chain File

The certificate chain file links your certificate to the CA file. You require one of these files if you are using a Global Site ID or if you are using other types of certificates from another provider.

2.10.1.3 Configuration Files

There are several files involved with the certificates. Put these files in the appropriate directory. You can set up the configuration differently, however, this is the standard configuration.

[Table 2–9](#) lists the certificate files and locations.

Table 2–9 Certificate files and locations

File	Directory location
Certificate File	<ORACLE_HOME>/Apache/Apache/conf/ssl.crt/
Certificate Authority (CA) Certificate File	<ORACLE_HOME>/Apache/Apache/conf/ssl.crt/
Certificate Chain File (if available)	<ORACLE_HOME>/Apache/Apache/conf/ssl.crt/
Key File	<ORACLE_HOME>/Apache/Apache/conf/ssl.key

2.10.2 Securing Ports to Use Certificates and HTTPS

With HTTPS, you use certificates for ports to increase security. To set this up, edit the `zone.properties` file.

[Table 2–10](#) lists the locations of the `zone.properties` files.

Table 2–10 zone.properties file location

Operating System	Location
Windows NT/2000	<ORACLE_HOME>\Apache\Jserv\servlets\zone.properties
UNIX	<ORACLE_HOME>/Apache/Jserv/etc/zone.properties

Then select a configuration method. You can configure Oracle Portal so that ports use HTTPS when needed, or at all times.

Ports Use HTTPS When Needed

You can set up HTTPS so that the browser and middle tier use HTTPS only when needed. HTTP is used for communication within the middle tier. This saves the overhead of HTTPS when it is not needed.

First, add the following line to the `zone.properties` file:

```
servlet.page.initArgs=useScheme=http
```

This specifies that the Parallel Page Engine should use HTTP for all requests to the middle tier on which it is running. The Parallel Page Engine uses HTTP for all requests to the portal.

Next, add the following line to the `zone.properties` file:

```
servlet.page.initArgs=usePort=80 (or any other valid port)
```

This specifies that the Parallel Page Engine should use port 80 for all requests to the middle tier on which it is running. If a request is made on port 433, the Parallel Page Engine ignores the port and uses port 80 instead.

Finally, in the `wdbsvr.app` file, in the DAD configuration add the following to the `cgi_env_list` parameter:

```
cgi_env_list=REQUEST_PROTOCOL=HTTPS,SERVER_PORT=(https port)
```

This provides the security of HTTPS, without the overhead of parallel connections over the secure protocol.

Ports Use HTTPS At All Times

You can set up HTTPS so that it is used by the ports at all times. The Parallel Servlet must be aware of which port(s) are operating under HTTPS.

Add the following line to the `zone.properties` file:

```
servlet.page.initArgs=httpsports=<port1>:<port2>:. . . :<portn>
```

Each port in this list operates using the HTTPS protocol, and must have a certificate created on the Oracle HTTP Server on that port.

See also:

- [Section A.1.2, "JServ Configuration File \(zone.properties\)"](#)
 - Oracle Portal Online Help topic: *Configuring the Login Server for LDAP user authentication*
-
-

2.10.3 Configuring Oracle Portal to Use HTTPS

This section addresses how to configure Oracle Portal for HTTPS. It is possible to configure the system so that only the Login Server is configured for HTTPS, or configure it such that both Oracle Portal and the Login Server use HTTPS.

The Apache `mod_ssl` documentation describes how to configure the server to support HTTPS ports. After configuring the server to support HTTPS ports, run the `ssodatan` or `ssodatax` script(s), specifying the appropriate protocol and ports. For example, if you wanted to configure the Login Server to use HTTPS, but have Oracle Portal on HTTP, then run the `ssodatan` script as follows:

```
ssodatan -w http://portal.acme.com/pls/portal30/ -l  
https://login.acme.com/pls/portal30_sso/ -s portal30 -o portal30_sso
```

See also: [Section B.4, "Configuring a New Oracle Portal Instance and Login Server with the `ssodatan` Script"](#)

The following sections address the particular requirements for configuring Oracle Portal for HTTPS.

2.10.4 Adding Certificate Entries in `httpd.conf`

The Oracle HTTP Server configuration file, `httpd.conf`, contains all of the configuration information for the Oracle HTTP Server *powered by Apache*, including the certificate configuration. Enter the path locations for the following configuration lines. These configuration lines should already exist in comment form (`#`).

[Table 2-11](#) lists the certificate entries in the Oracle HTTP Server configuration files.

Table 2-11 *Certificate Entries in the Oracle HTTP Server configuration file*

File	Description
<code>SSLCertificateFile</code>	Enter the path location for the certificate file, either the Trial or the purchased certificate.

Table 2–11 Certificate Entries in the Oracle HTTP Server configuration file

File	Description
SSLCertificateKeyFile	Enter the path location for the Key file which contains the key to decrypt your certificate.
SSLCertificateChainFile	Enter the path location for the Certificate Chain File you received from your provider.
SSLCACertificateFile	Enter the path location for the CA Certificate File you received from your provider.

Note: Do not use the environment variables such as your `<Oracle Home>` to specify the path location of these configuration files. Use the fully-qualified path location.

The following subsections indicate the configuration entries required in the `httpd.conf` file, corresponding to each type of certificate. These configuration entries have been used successfully to set up Verisign certificates.

The usage varies slightly depending upon the certificate type you are installing. For example, if the certificate you are using has a chain file, then follow the Global Site ID configuration described in the following. If your certificate only uses a CA certificate file, then use the Secure Site ID configuration.

2.10.4.1 Global Site ID

```
SSLCertificateFile      <ORACLE_HOME>/Apache/Apache/conf/ssl.crt/<certificate file>
SSLCertificateKeyFile   <ORACLE_HOME>/Apache/Apache/conf/ssl.key/<key file>
SSLCACertificateFile   <ORACLE_HOME>/Apache/Apache/conf/ssl.crt/<certificate chain file>
SSLCertificateChainFile <ORACLE_HOME>/Apache/Apache/conf/ssl.crt/<CA Certificate file>
```

2.10.4.2 Secure Site ID

```
SSLCertificateFile      <ORACLE_HOME>/Apache/Apache/conf/ssl.crt/<certificate file>
SSLCertificateKeyFile   <ORACLE_HOME>/Apache/Apache/conf/ssl.key/<key file>
SSLCACertificateFile   <ORACLE_HOME>/Apache/Apache/conf/ssl.crt/<CA Certificate file>
```

2.10.4.3 40 bit Site ID

SSLCertificateFile <ORACLE_HOME>/Apache/Apache/conf/ssl.crt/<certificate file>
SSLCertificateKeyFile <ORACLE_HOME>/Apache/Apache/conf/ssl.key/<key file>
SSLCACertificateFile <ORACLE_HOME>/Apache/Apache/conf/ssl.crt/<CA Certificate file>

Note: The Chain file and CA Certificate file appear to be inverted with their configuration entries. This is intentional, and necessary for Oracle Portal to work properly.

See also: [Section A.1.1, "Oracle HTTP Server Configuration File \(httpd.conf\)"](#)

2.11 Enabling Secure Socket Layer (SSL)

Oracle Portal and the Login Server can be configured to run in HTTPS mode if your portal requires increased security. For optimal performance, you can also choose to have a mixed configuration where Oracle Portal is run in HTTP mode and the Login Server is run in HTTPS mode.

Secure Socket Layer (SSL) is responsible for securing Web HTTP communication between a browser and a Web server in plain HTTP over SSL (named HTTPS). Enabling SSL to work with the Oracle HTTP Server is handled by the `mod_ssl` package which is provided with the Apache Web server. It uses the URL scheme HTTPS rather than HTTP and a different server port.

Notes:

- You must be the portal administrator to enable or disable Secure Socket Layer (SSL) in Oracle Portal and on the Login Server.
- For more information on enabling SSL on the server, see the *Oracle9i Application Server Installation Guide*.
- For more information about enabling and disabling SSL, see the Apache interface to `mod_SSL` at the following location:

<http://www.modssl.org>

2.11.1 Configuring SSL on Oracle Portal and the Login Server

Follow these steps to configure SSL on both Oracle Portal and the Login Server:

1. Obtain a valid server certificate. The certificate that ships with Oracle9i Application Server will not work for Oracle Portal.
2. Generate the certificate request from the following location:

```
cd <ORACLE_HOME>\Apache\open_ssl\bin
```

where <ORACLE_HOME> is the location of your Oracle9i Application Server product.

3. Edit the `openssl.cnf` file and find the following line:

```
RANDFILE = $ENV::HOME/.rnd
```

4. Add `HOME = .` before this line so that it appears as follows:

```
HOME = .
RANDFILE = $ENV::HOME/.rnd
```

5. Run the following commands:

```
openssl md5 * > rand.dat
openssl genrsa -rand rand.dat -des3 1024 > key.pem
openssl req -new -key key.pem -out csr.pem -config openssl.cnf=
```

6. In the last command, which generates the certificate request, provide the name of your server, including the domain, when prompted for Common Name. For example: `abc.oracle.com`.

This action produces a `key.pem` and `csr.pem` file. Send the `csr.pem` file to the certificate authority to obtain the certificate. For example, in the case of a Verisign trial certificate, you would do the following:

- a. Go to <http://www.verisign.com> and request a trial certificate by clicking "Secure Your Website."
- b. In the form fields, locate the CSR field. Copy and paste the contents of the `csr.pem` file that was generated earlier in this step.
- c. Verisign sends you an e-mail with the trial certificate attachment (let's name it `portalcert.crt`) with further instructions. One of the steps describes how to obtain the Root Trial CA from Verisign. Import this certificate into your browser according to these instructions.

- d. Export the certificate from your browser into a file that you can later install onto the Oracle HTTP Server Apache listener.
- e. Save the certificate as a "Base-64 encoded X.509" certificate. Let's call this file `trialcacert.crt`.

Exporting from Internet Explorer

Go to Tools, Internet Options, Content, Certificates, Trusted Root, Certification Authorities, and then Export.

Exporting from Netscape

Refer to Netscape documentation on exporting certificates.

On Windows NT

Remove the password from `key.pem` as follows:

```
copy key.pem key.pem-orig  
  
openssl rsa -in key.pem-orig -out key.pem
```

- 7. Copy the certificates to the appropriate locations:

```
copy portalcert.crt to Apache\Apache\conf\ssl.crt  
copy trialcacert.crt to Apache\Apache\conf\ssl.crt  
copy key.pem to Apache\Apache\conf\ssl.key  
copy key.pem-orig to Apache\Apache\conf\ssl.key
```

2.11.2 Setting Login Server Query Path URL

Oracle Portal maintains the URL prefix of the Login Server which accesses certain information through HTTP calls from the database, using the `UTL_HTTP` package. These calls must be done through HTTP rather than HTTPS.

Thus, if Oracle Portal and the Login Server are configured to use HTTPS, access to an HTTP port on the Login Server is still required to support these interfaces. The calls made across this interface are required for the following reasons:

- Obtain the list of external applications to allow the external applications portlet to customize.
- Perform the mapping of the Single Sign-On user name to the external application user name.
- Present an interface to verify the existence of a user.

To set this URL prefix, which is called the Login Server Query Path URL, complete these steps:

1. Log on to Oracle Portal as the portal administrator.
2. Click the **Administer** tab.
3. Click **Global Settings** in the Services Portlet.
4. Scroll down to the section on Login Server, and edit the **Query Path URL**. Set this field to an HTTP URL for the Login Server.

Figure 2–1 Login Server Query Path URL Prefix field

Login Server Settings

The fields below list the information used by Oracle Portal to identify itself to the Login Server. To update these values, click  for more information.

Application Id:	1321
Login URL:	http://webdbsvr.us.oracle.com:5000/pls/WWW_30624_SSO/WWW_30624_SSO.wwsso_app_ad
Cookie Version:	v1.0
Query Path URL Prefix:	<input style="width: 90%;" type="text" value="http://webdbsvr.us.oracle.com:5000/pls/WWW_30624_S"/>

By default, it is the same prefix as specified for the Login Server when running the `ssodatan` or `ssodatax` scripts. However, if these scripts specify an HTTPS protocol, then manually update this parameter to use an HTTP protocol.

See also:

- [Section B.4, "Configuring a New Oracle Portal Instance and Login Server with the `ssodatan` Script"](#)
 - [Section B.5, "Updating an Existing Portal Instance with the `ssodatax` Script"](#)
 - [Section 4.4.4, "Step 4: Associate Nodes with the Same Login Server"](#)
-

2.11.3 Adding SSO Enabler Configuration Entries for HTTPS Mode

If you are using SSL, the default port is 443. With Oracle Portal versions prior to 3.0.8, you need to create two enabler configuration entries, and two corresponding partner configuration entries on the Login Server. Specify the :443 port for one entry, and exclude it for the additional entry.

To add the additional entry, follow the basic procedure of adding the partner entry on the Login Server using the Login Server Administration user interface, and then add the configuration entry on the Oracle Portal side by using the `ssodatax` script.

If using Oracle Portal version 3.0.8 or later, you only need a single entry, one which excludes the `:443` from the URL.

Note: This step is only required to support Netscape browsers. Microsoft Internet Explorer does not require this step.

2.11.4 Configuring HTTPS with Microsoft Internet Explorer

In the Oracle HTTP Server configuration file, `httpd.conf`, comment out the following line to permit Microsoft Internet Explorer browsers to work in HTTPS mode:

```
SetEnvIf User-Agent ".*MSIE.*" nokeepalive  
ssl-unclean-shutdown
```

2.11.5 Configuring HTTPS with Virtual Hosts

If you want to setup a virtual host, it can be done in one of two ways:

- Through an IP address (for example, `123.1.3.2`)
- Through an IP name (for example, `server.oracle.com`)

When the IP name is used, several aliases use the same IP address. In this case, Apache (or any browser supporting virtual name addresses) looks at the `Host` field in the HTTP request and determines which of the virtual addresses should be emulated.

However, when SSL is used, the IP name is encrypted. This causes the problem, because the software does not know which decryption key to use since the keys differ by virtual name. If there were 1000 separate virtual addresses supported, then on average the software would try 500 different keys to determine which key to use to decode the message. This is not practical, at least for performance reasons.

Notes:

- It is more difficult to configure virtual hosts to use HTTPS since the SSL encryption prevents virtual hosts from being resolved in the way that it is done in non-SSL mode.
 - There are some workarounds from which to choose. One is to only use virtual names on the home page and other pages where you do not need protection.
-
-

2.11.5.1 SSL Protection Pages

- Obtain an IP address for each virtual host and then obtain a TCP/IP card that can handle multiple IP addresses, one for each virtual host.
- Use one IP address with different port numbers for each virtual host.
- Use one IP address but use different directories, one for each "virtual host" (for example, `https://ssladdress.com/virtualname1/<page desired>`).

2.12 Configuring the Reports Security Portlet

The Reports Security Portlet is installed as part of the Oracle Portal installation. Depending on the edition of the Oracle9i Application Server installed, the Reports Security Portlet may be hidden. To show the Reports Security Portlet:

1. Log on to Oracle Portal as the portal administrator.
2. Click the **Administer** tab.
3. Click **Edit Page**.
4. Select **Oracle Reports Security**.
5. Click **Show**.
6. Click **Close**.

The Reports Security Portlet displays in the **Administer** tab.

Basic Oracle Portal Configurations

The Oracle Portal architecture supports a wide variety of topologies and configuration options. Factors that determine how to configure Oracle Portal depend on the intended purpose, its local network environment, the anticipated load, and how it is accessed by users.

The basic Oracle Portal configurations covered in this chapter include:

- [Configuring Oracle Portal on a Standalone Laptop](#)
- [Configuring Oracle Portal as an Integrated Server](#)
- [Configuring a Traditional Three-tier Architecture](#)
 - [Configuring Distinct Oracle Portal and Login Server Instances](#)

3.1 Configuring Oracle Portal on a Standalone Laptop

The standalone laptop configuration is very appropriate if you require a very compact and portable installation. For example, you would use this setup if you are demonstrating Oracle Portal without a network connection or if you change your hostname frequently.

In this configuration, the browser, Oracle9i Application Server middle-tier, and Oracle8i database containing the Login Server and Oracle Portal objects, all reside on a single laptop. To set this up, follow these steps:

1. Install Oracle9i Application Server with Oracle Portal according to the instructions in the *Oracle9i Application Server Installation Guide* for your particular operating system.

Note: When prompted for the name of the host, enter "localhost."

2. Edit your local TCP/IP host file, HOSTS, so that an entry exists for the local host. For example:

```
127.0.0.1 localhost
```

Note: The HOSTS file is created when you install TCP/IP to include remote host names and their IP addresses for each computer with which you will communicate. Specify only a valid DNS name for your host. If you are unfamiliar with editing your host file, consult your network administrator for assistance.

3. Verify that the Oracle HTTP Server's *powered by Apache* file, `httpd.conf`, specifies the `ServerName` entry as local host. For example, replace `<hostname>` with `localhost` as follows:

```
ServerName localhost
```

Note: This file is located in the Oracle Home containing your Oracle9i Application Server installation:

```
<ORACLE_HOME>/Apache/Apache/conf/httpd.conf
```

See also: [Section A.1.1, "Oracle HTTP Server Configuration File \(httpd.conf\)"](#)

4. Associate the new Oracle Portal installation with the Login Server by running the `ssodatan` script as follows:

- a. Start a command line prompt.
- b. Change to the `<ORACLE_HOME>/portal30/admin/plsql/` directory where Oracle Portal is installed.
- c. Enter the following command:

```
ssodatan <-w portal_url> <-l login_server_url> <-s portal_schema>  
<-p portal_password> <-o sso_schema> <-d sso_password> <-e pstore_  
schema> <-c portal_connect_string>
```

Example

```
ssodatan -w http://localhost/pls/portal/ -l http://localhost/pls/portal_
sso/ -s portal30 -p portal30 -o portal30_sso -c orcl
```

See: For parameter descriptions, see [Section B.4, "Configuring a New Oracle Portal Instance and Login Server with the ssodatan Script"](#).

Note: In this example, the ORACLE_HOME environment variable is set to the Oracle9i Application Server's home and the connect string "orcl" points to the database containing Oracle Portal and the Login Server schema. The default Oracle Portal schema is assumed to be "portal30" and the Login Server schema is assumed to be "portal30_sso".

For parameter descriptions, see [Table B-3, "ssodatan script parameters"](#).

5. Access Oracle Portal by entering the following in your browser's URL address field:

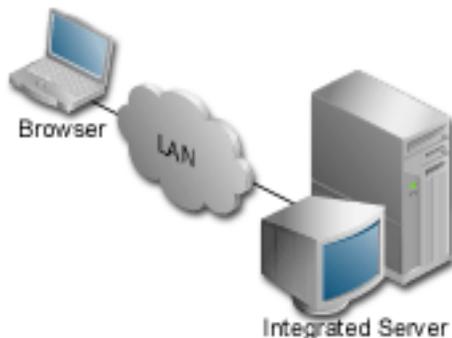
```
http://localhost/pls/portal30/
```

See: [Section 2.4, "Accessing Oracle Portal in Your Browser"](#)

3.2 Configuring Oracle Portal as an Integrated Server

While a standalone laptop configuration is only accessible to the person using the laptop, the integrated server configuration allows any authorized user to access the Oracle Portal installation remotely across a LAN with a browser.

Figure 3–1 Integrated server configuration



1. Install Oracle9i Application Server with Oracle Portal according to the instructions in the *Oracle9i Application Server Installation Guide* for your particular operating system.

Note: When prompted for the name of the host, enter the full domain name of the server where you are installing Oracle Portal. Take a note of this name for future reference.

2. In the Oracle HTTP Server's *powered by Apache* file, `httpd.conf`, verify that the `ServerName` entry is specified with the full domain name of the server containing your Oracle Portal installation. For example:

```
ServerName portal.acme.com
```

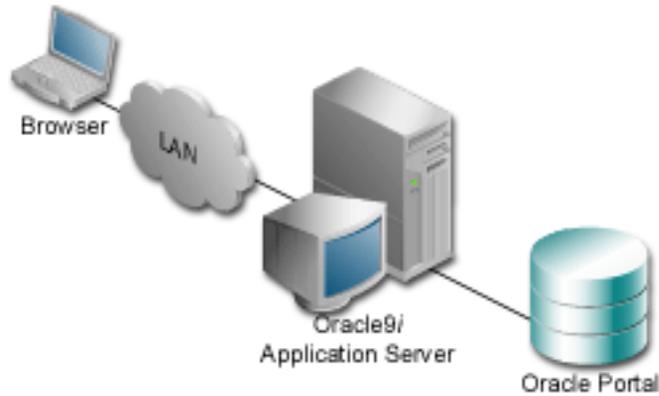
3. You would access Oracle Portal by entering the following in your browser's URL address field:

```
http://portal.acme.com/pls/portal30/
```

3.3 Configuring a Traditional Three-tier Architecture

If the Oracle9i Application Server middle-tier, which includes the Oracle HTTP Server and the Servlet Engine, is installed on a separate machine from Oracle Portal, the traditional three-tier architecture is used. For more information, see the *Oracle9i Application Server Overview Guide* in the documentation library.

Figure 3–2 *Traditional three-tier architecture*



This architecture is set up the same way as the integrated server configuration with the following exceptions:

1. Install Oracle9i Application Server with Oracle Portal according to the instructions in the *Oracle9i Application Server Installation Guide* for your particular operating system. The Login Server is installed as part of the Oracle Portal installation in a separate schema. The default Oracle Portal schema is assumed to be `portal30` and the Login Server schema is assumed to be `portal30_sso`.

The Oracle9i Application Server middle-tier is installed on a separate machine.

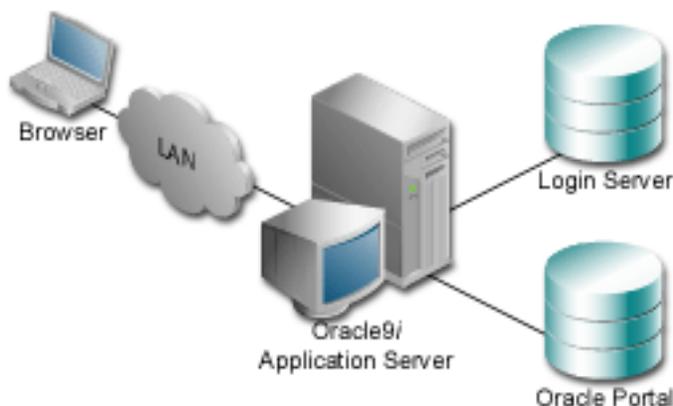
2. During Oracle Portal installation, when prompted for a connect string, the database connection string must identify a database instance in a host which is separate from the Oracle9i Application Server middle-tier.
3. In the Database Access Descriptor (DAD) configuration page, make sure that the connect string for both the Oracle Portal DAD and the Login Server DAD is the same.

Note: The DAD configuration page is accessible by clicking Listener Gateway Settings in the Services portlet.

3.3.1 Configuring Distinct Oracle Portal and Login Server Instances

This configuration can be used if you want to have multiple Partner Applications accessing a centralized Login Server to leverage Single Sign-On. In this case, Oracle Portal and the Login Server reside on separate database instances.

Figure 3–3 Three-tier architecture with distinct Oracle Portal and Login Server



Since this configuration uses distinct instances for the Login Server and Oracle Portal, a slightly different installation sequence is required. Running the Oracle Universal Installer for Oracle9i Application Server on the middle-tier installs the application server and loads up all the source files and scripts enabling more complex topologies such as this to be configured.

The following is one of several ways to set up this configuration:

1. Run the Oracle9i Application Server installer on the middle-tier according to the instructions in the *Oracle9i Application Server Installation Guide* for your particular operating system.

For the connect string, specify the hostname that points to the Oracle Portal host.

Upon installation, the middle-tier software such as the PL/SQL Gateway reside in the Oracle9i Application Server host in the following Oracle Home location:

```
<ORACLE_HOME>/portal30/admin/plsql
```

where <ORACLE_HOME> is the location of the Oracle9i Application Server.

2. Verify that a connect string for the Login Server host is defined in the `<ORACLE_HOME>/network/admin/tnsnames.ora` file on the Oracle9i Application Server.
3. From the Oracle9i Application Server, run the `linstall` script located in `<ORACLE_HOME>/portal30/admin/plsql` to install a standalone Login Server on the Login Server host, by specifying the connect string to the Login Server database instance.
4. Associate the Oracle Portal node with the Login Server.
 - a. Log on to the Login Server by accessing its DAD, for example:
`http://server.company.com/pls/portal30_sso/`
 - b. Log on, and add Oracle Portal as a Partner Application. The home URL should be similar to:
`http://server.company.com/pls/portal30/portal30.home`

 The success URL should be similar to:
`http://server.company.com/pls/portal30/portal30.wwsec_app_priv.process_signon`
 - c. Click **Apply**.
 - d. This generates a site id, site token, and encryption key for the newly added Oracle Portal Partner Application. Use these values to run the `ssodatax` script on Oracle Portal.
5. From the Oracle9i Application Server's `<ORACLE_HOME>/portal30/admin/plsql` location, run the `ssodatax` script:

Syntax

```
ssodatax <-i portal_site_id> <-t portal_site_token> <-k encryption_key>
<-w portal_url> <-l login_server_url> <-s portal_schema><-p portal_password>
<-v cookie_version> <-o sso_schema> <-e pstore_schema> <-r pstore_password>
<-b pstore_dblink> <-c connect_string> <-n ps_connect_string>
```

Specify the site ID, site token, and encryption key obtained in Step 4. For the other parameters, enter:

Example

```
ssodatax -i 1234 -t A1B2C3 -k X9Y8Z7 -w
http://server.oracle.com:3000/pls/portal30/ -l
```

```
http://server.oracle.com:3000/pls/portal30_sso/ -s portal30 -v v1.1 -o  
portal30_sso -c orcl
```

See: For parameter descriptions, see [Section B.5, "Updating an Existing Portal Instance with the ssodatax Script"](#).

For the connect string (-c parameter), enter the TNS entry for the Oracle Portal database connection.

When the `ssodatax` is run, it only updates information in the specified database connection. Furthermore, it does not clear any information that may be in the enabler configuration table unless the host or port specified in the URL prefix for the -w parameter matches one that is already in the table. In this case, the newly-provided information replaces the previous values.

This chapter discussed basic Oracle Portal configurations. For topologies with several database providers or multiple content areas, each residing on a separate Oracle Portal node, consider a distributed Oracle Portal configuration to aggregate these into a single Oracle Portal instance discussed in [Chapter 4, "Distributed Oracle Portal Installations"](#).

Distributed Oracle Portal Installations

In a distributed Oracle Portal configuration, there is a centralized Login Server, and two or more Oracle Portal nodes which all access the same Login Server for Single Sign-On authentication. Furthermore, each Portal node is on a separate database instance.

There are many benefits to such an environment, including the ability to share portlet provider information across all nodes as well as increased scalability, availability, and system throughput.

Specific topics covered include:

- [What is a Node?](#)
- [Benefits of a Distributed Oracle Portal Environment](#)
- [Node Requirements](#)
- [Configuring a Distributed Oracle Portal Environment](#)

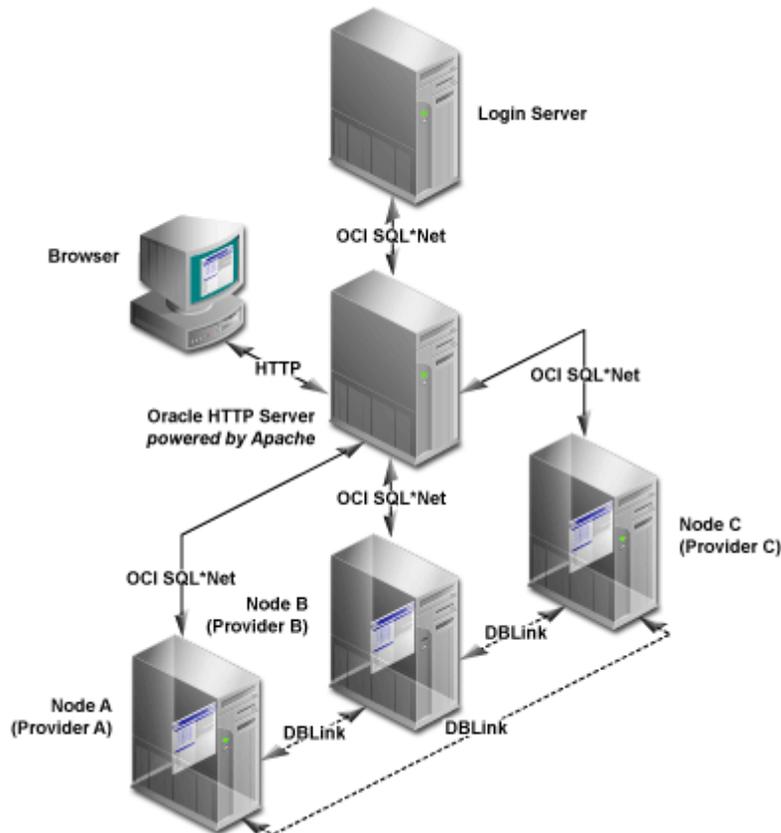
See also:

The following are available on the Oracle Technology Network at:
<http://technet.oracle.com/products/iportal>:

- *Building Scalable and Performant Portal Solutions Using Oracle Portal*
 - *Page Generation and Assembly Scalability in Oracle9iAS Portal*
-
-

Figure 4-1 illustrates a distributed Oracle Portal environment showing the communication channels that exist between the nodes themselves, between each of the nodes and the Oracle HTTP Server powered by Apache, with the Login Server.

Figure 4-1 Distributed Oracle Portal installations topology



4.1 What is a Node?

A distributed environment refers to several installations of Oracle Portal to create a multi-node environment. Each node is a complete Oracle Portal installation which resides in a separate database instance and is configured to operate in a distributed

manner. Each node in the system may operate either independently of the other nodes or in conjunction with the other nodes.

The node containing the page that you are currently viewing is considered the *local* node. All other installations are considered *remote* nodes. However, a page in Oracle Portal can contain portlets that were created on either the local or remote nodes.

Node registration refers to associating nodes to each other so that they may be able to share information. Node registration is done by completing a set of configuration steps which are discussed later in this chapter.

4.2 Benefits of a Distributed Oracle Portal Environment

A distributed or multi-node Oracle Portal environment provides the following benefits over a single node environment.

4.2.1 Portlet Provider Information Shared Across Nodes

In a distributed Oracle Portal environment, provider information can be shared across nodes. During node registration, provider registration also occurs. When a provider is registered on a remote node, the portlets for that provider are populated in the node's portlet repository which allows you to build pages with portlets residing on remote nodes. In addition to sharing provider information, the distributed environment also lets you group providers accordingly.

4.2.2 Scalable Solutions

When provider registration occurs, the provider registry information is replicated on the remote node. Only the provider registry information is replicated, not the actual provider implementation. The provider implementation package resides only on the host node of that provider.

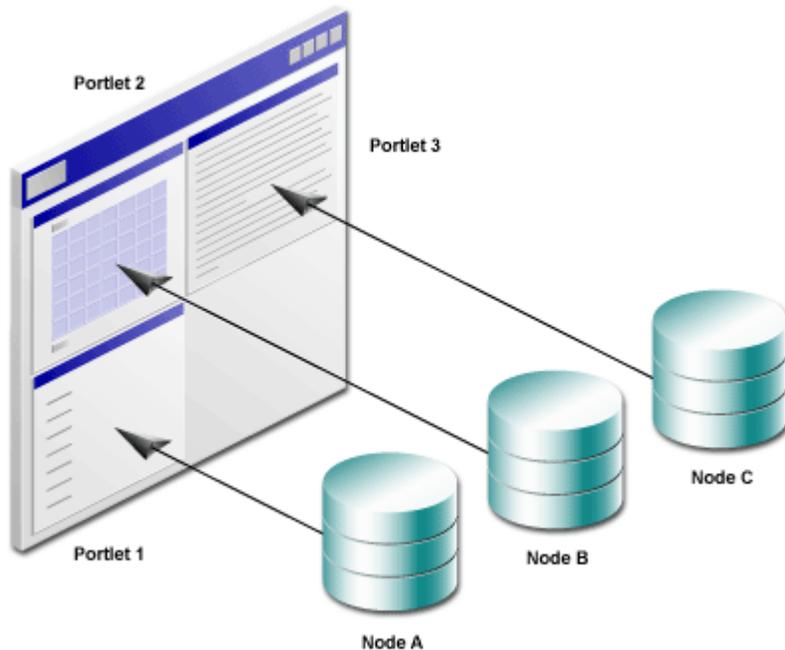
A page may consist of portlets from any number of nodes that participate in the distributed environment. When such a page is rendered, the portlets are executed on the host node of each of the portlet providers (the node where the provider implementation package reside).

For example, consider the following scenario which can be implemented in scalable environments:

A page consists of portlet 1 which resides on node a and portlet 2 which resides on node b. When the page is rendered, portlet 1 is executed on node a and portlet 2 is executed on node b.

Figure 4-2 illustrates the display of portlets created on different nodes on a single page.

Figure 4-2 *Displaying portlets created on different nodes on a single page*



Such a scenario enables you to access multiple machines with increased performance and increased system throughput since the rendering of a page is distributed among several database instances. The execution of the portlets on the different databases is done in parallel.

The distributed environment provides high availability. If one node fails, the other nodes continue to process with full access except on portlets residing on the failed node.

4.3 Node Requirements

You must meet the following node requirements for configuring a distributed Oracle Portal architecture:

- [Section 4.3.1, "Common Cookie Domain"](#)
- [Section 4.3.2, "Oracle HTTP Server powered by Apache Configuration"](#)
- [Section 4.3.3, "Common Cookie Name"](#)
- [Section 4.3.4, "Common Login Server"](#)
- [Section 4.3.5, "Symmetric Node Registration"](#)
- [Section 4.3.6, "URLs in Portlets"](#)

4.3.1 Common Cookie Domain

The cookie domain for the Oracle Portal session cookie must be common for all nodes that participate in the distributed Oracle Portal environment. If the cookie domain is changed on one node, it must be changed on all other nodes. Otherwise, the Oracle Portal nodes in your environment fail in a distributed manner.

Cookies are scoped to the host which created them, unless they are specified to be scoped to a larger domain. By default, the Oracle Portal session cookies are scoped to the root path of the server that generated them. For more information, see [Section A.1.5, "Login Server Configuration Table"](#).

See: [Section 4.4.1, "Step 1: Create Oracle Portal Nodes"](#)

4.3.2 Oracle HTTP Server *powered by Apache* Configuration

A Portlet Provider that resides on a node may be accessed by any other node that exists in the network and for which a communication path has been established. The Oracle HTTP server *powered by Apache* is responsible for establishing a communication path and for displaying portlets for each node.

Choose either of the following scenarios in your distributed environment:

- Have multiple Oracle HTTP Servers *powered by Apache* running, one for each node
- Have a single Oracle HTTP Server *powered by Apache* running and accessible by all nodes

When communicating between browsers, the Oracle Portal session cookie is sent to each portlet execution request. Also, the cookie domain consists of the `<host.domain:port>`.

When using multiple Oracle HTTP Servers, this results in a different `<host.domain:port>`. Only one node has the same cookie domain as the Login Server. Thus, in this case, when the user tries to access a node by clicking a portlet's URL, the Oracle Portal session cookie is *not* sent by the browser.

To resolve this situation, a common cookie domain name is required. To do this, run the `ctxckupd.sql` script on all nodes in your distributed environment.

See: [Section 4.4.2, "Step 2: Create Same Cookie Domain"](#)

4.3.3 Common Cookie Name

In an Oracle Portal distributed environment, each Oracle HTTP Server *powered by Apache* must have a Database Access Descriptor (DAD) configuration for each of the portal nodes that participate in the distributed system. Also, the **Session Cookie Name** field in the DAD configuration must be the same across nodes.

See: [Section 4.4.3, "Step 3: Edit Oracle Portal DADs"](#)

4.3.4 Common Login Server

All the nodes that participate in the distributed Oracle Portal environment must use the same Login Server. Otherwise, you may encounter a runtime error if a node that is registered to participate in the distributed Oracle Portal environment is not using the same Login Server as the other nodes. In this case, you would fail to log onto the Oracle Portal node via Single Sign-On (SSO) and not have access to the portlets on that node.

See: [Section 4.4.4, "Step 4: Associate Nodes with the Same Login Server"](#)

4.3.5 Symmetric Node Registration

All nodes included in the distributed architecture must be symmetrically registered between themselves. For example, if the distributed Oracle Portal environment consists of three nodes (a, b, and c), make sure that the following registrations exist.

- Node a is registered on b and c
- Node b is registered on a and c

- Node c is registered on a and b

See: [Section 4.4.7, "Step 7: Register Nodes Between Themselves"](#)

4.3.6 URLs in Portlets

If you are creating your own custom portlets using the Oracle Portal Development Kit (PDK), use absolute URLs (not relative URLs) for portlets destined to be run in a distributed Oracle Portal environment.

See: The Oracle Portal Development Kit on the Oracle Technology Network at:

<http://technet.oracle.com/products/iportal>

Click the Portal Development Kit (PDK) link to access the latest PDK software and documentation.

4.4 Configuring a Distributed Oracle Portal Environment

You must have the required privileges on the node and on the Login Server to perform the steps in this section:

- Full Administrator privileges on the Login Server to change any of its settings.
- Oracle Portal Administrator privileges to access the Administer tab on the Oracle Portal home page.

See: Oracle Portal Online Help Content Area Help topics: *What is the Login Server and Single Sign-On?* and *How does security work?*

This section describes the process for setting up a distributed Oracle Portal environment. For the purpose of the following example, the environment consists of two nodes, named node a and node b.

The steps include the following:

- [Section 4.4.1, "Step 1: Create Oracle Portal Nodes"](#)
- [Section 4.4.2, "Step 2: Create Same Cookie Domain"](#)
- [Section 4.4.3, "Step 3: Edit Oracle Portal DADs"](#)

- [Section 4.4.4, "Step 4: Associate Nodes with the Same Login Server"](#)
- [Section 4.4.5, "Step 5: Create a User on the Login Server with Administrator Privileges"](#)
- [Section 4.4.6, "Step 6: Discover the Name of Each Node"](#)
- [Section 4.4.7, "Step 7: Register Nodes Between Themselves"](#)
- [Section 4.4.8, "Step 8: Refresh the Portlet Repository for Each Node"](#)
- [Section 4.4.9, "Step 9: Create Additional Nodes"](#)

4.4.1 Step 1: Create Oracle Portal Nodes

As stated earlier, a node is an Oracle Portal installation. To configure a distributed Oracle Portal environment, you must have at least two Oracle Portal installations, one for node a and the other for node b.

To create a node, install Oracle Portal as instructed in the *Oracle9i Application Server Installation Guide* for your particular operating system.

After creating the first node, additional nodes can be created without associated Login Server schemas. The `-noosso` parameter creates only an Oracle Portal schema. For more information, see [Section B.2, "Manually Installing Oracle Portal with the wininstall Script"](#).

You must perform an installation of Oracle Portal for each node you want to have in your distributed environment.

Note: Be sure that the two nodes are created on two different databases since the distributed Oracle Portal functionality is not supported on nodes that exist on the same database.

4.4.2 Step 2: Create Same Cookie Domain

Note: This step applies only if your distributed Oracle Portal environment is running multiple Oracle HTTP Servers *powered by Apache*. If you are running only one Oracle HTTP Server, skip this step.

To resolve the issue of a different `<host.domain:port>` configuration for each node, the same cookie domain *must* exist across nodes in a distributed Oracle Portal environment in order for the Oracle Portal session cookie to be sent successfully by the browser. The solution is to run the `ctxckupd.sql` script on all the nodes in your distributed Oracle Portal environment.

To create the same cookie domain on all nodes:

1. If you have an Oracle HTTP Server *powered by Apache* running on the computer on which your node is located, stop the server by entering the following command from a command prompt:

```
<ORACLE_HOME>/Apache/Apache/bin/apachectl stop
```

Note: On Windows NT/2000, stop the HTTP Server from the System Control Panel.

2. On the database where your node is installed, log on to SQL*Plus with the appropriate username and password. For example:

```
sqlplus nodea/nodea
```

3. Enter the following command:

```
@ctxckupd.sql
```

4. When prompted, enter the domain name for the session cookie as required.

Note: Note this name as you need to enter the same cookie domain name for all remote nodes.

5. Repeat the above steps for all other remote nodes in your distributed Oracle Portal environment.

See: [Section B.6, "Modifying the Scope of the Portal Session Cookie"](#)

4.4.3 Step 3: Edit Oracle Portal DADs

A distributed Oracle Portal environment requires that each node has a separate Database Access Descriptor (DAD) for each Oracle HTTP Server *powered by Apache*. Also, the **Session Cookie Name** field in the DAD configuration must be the *same* across nodes.

Upon installation, a DAD is created for each node. This step requires you to edit the DAD on each node and specify a common cookie name across nodes.

Note: In a distributed Oracle Portal environment, make sure there is only one installation of the Login Server.

4.4.3.1 Access DAD Configuration Page

DADs are created from the Database Access Descriptor configuration page in Oracle Portal which you can access in the following ways:

In the **Services** portlet, click **Listener Gateway Settings**. By default, the Services portlet is located on the Oracle Portal home page's **Administer** tab.

- a. Click **Gateway Database Access Descriptor Settings**.
- b. In the **Edit/Delete Database Access Descriptors** section, click  next to the node's DAD entry. The DAD configuration page is displayed.

In your browser, enter the following:

- a. `http://<hostname.domain>:<port>/pls/admin_/dadentries.htm`
- b. In the **Edit/Delete Database Access Descriptors** section, click  next to the node's DAD entry. The DAD configuration page is displayed.

4.4.3.2 Configuration

To provide the *same* cookie name for Oracle Portal nodes in your distributed environment:

1. From the appropriate node's Database Access Descriptor configuration page, edit the DAD for node a.

In the **Session Cookie Name** field, enter a name. For example:

`dist_portal_session_cookie`

2. From the appropriate node's Database Access Descriptor configuration page, edit the DAD for node a. In the **Session Cookie Name** field, enter the *same* cookie name you entered for node b. For example:

```
dist_portal_session_cookie
```

Note: It is important that the **Session Cookie** is the *same* for both (all) DADs in your distributed environment, except for the Login Server. The cookie is used by the Oracle Portal security subsystem to perform session establishment and authentication during Single Sign-on.

3. From the Login Server's Database Access Descriptor configuration page, edit the DAD for the Login Server of node a. A Login Server DAD is appended with `_SSO` in its name.

In the **Session Cookie Name** field, enter a cookie name that is *different* from the name given for node a and node b. For example:

```
dist_portal_sso_session_cookie
```

Note: It is important to use a *different* name for the Session Cookie because the Login Server uses its own cookie. If the same name was used as that of Oracle Portal, the Oracle Portal session cookie would be overwritten by the Login Server resulting in Oracle Portal authentication failures.

4.4.4 Step 4: Associate Nodes with the Same Login Server

Important: Performing this step ensures that all nodes in your distributed Oracle Portal environment share the same Login Server. When installing Oracle Portal with the Oracle Universal Installer (OUI), each node is installed with its own Login Server. Therefore, when installing multiple nodes, they do not, by default, share the same Login Server.

For the purpose of our example, we must make node a and node b share the same Login Server.

Otherwise, any node that is not sharing the same Login Server as the other nodes in the distributed environment fail when performing any type of distributed functionality.

1. Associate node a with the Login Server of node a.

Note: By default, the OUI installs the first node and associates it to the appropriate Login Server. It is safe to skip this step unless you intend on editing the default Login Server association.

- a. Start a command line prompt.
- b. Change to the `<ORACLE_HOME>/portal30/admin/plsql/` directory in which Oracle Portal for node a is installed.
- c. Run the `ssodatan` script to associate a node to the Login Server.

See: For parameter descriptions, see [Section B.4, "Configuring a New Oracle Portal Instance and Login Server with the `ssodatan` Script"](#).

- d. Enter the script parameters as you would if node a were to function in a single node environment.
2. Register node b as a Partner Application to the Login Server of node a.
 - a. In the **Services** portlet, click **Login Server Administration**. By default, the Services portlet is located on the Oracle Portal home page's **Administer** tab.
 - b. Click **Administer Partner Applications**.
 - c. Click **Add Partner Application**.
 - d. Enter the following information on the Partner Application page:

Table 4–1 Partner Application Configuration Example

Field	Example Entry
Name	Oracle Portal Note: This registers node b as a Partner Application.
Home URL	<code>http://OraclePortalsvr.us.oracle.com:<port>/pls/<node b>/<node b>.home</code>

Table 4–1 Partner Application Configuration Example

Field	Example Entry
Success URL	<code>http://OraclePortalsvr.us.oracle.com:<port>/pls/<node b>/<node b>.wwsec_app_priv.process_signon</code>

Note: Specify the DAD name <node b> in lowercase characters.

3. Click **Apply.**

The Edit Partner Application page displays.

- 4. On the page that appears, copy *exactly* (cut and paste) the displayed information which you require to run the script in the next step. For example:**
- ID: 1323
 - Token: G06U7W36
 - Encryption Key: a21255e6b139ca34
- 5. Associate node b with the Login Server of node a.**
- a. Start a command line prompt.
 - b. Change to the <ORACLE_HOME>/portal30/admin/plsql/ directory in which Oracle Portal for node a is installed.
 - c. Run the `ssodatax` script to associate a node to a specific Login Server:

See: For parameter descriptions, see [Section B.5, "Updating an Existing Portal Instance with the `ssodatax` Script"](#).

- d. Enter the script parameters as required. For example, let's use the information generated by the example for registering a Partner Application to the Login Server of node a described above in step 4. This example also assumes that node b is installed on the database named "w816dev5" and node a is installed on the database named "w816dev6".

```
ssodatax -i 1323 -t G06U7W36 -k a21255e6b139ca34 -w
http://OraclePortalsvr.us.oracle.com:5000/pls/<node b>/ -l
http://OraclePortalsvr.us.oracle.com:5000/pls/<node_A_SSO>/ -s node_B -v
v1.0 -o node_A_SSO -c w816dev5
```

You have completed this step. Node a and node b are associated to the same Login Server.

4.4.5 Step 5: Create a User on the Login Server with Administrator Privileges

In this step, you need to create a user on the Login Server with full administrator privileges on node b. This user must be the schema owner of node b.

1. In the **Services** portlet, click **Login Server Administration**. By default, the Services portlet is located on the Oracle Portal home page's **Administer** tab.
2. Click **Administer Users**.
3. Click **Create New Users**.
4. Enter the configuration information as required.

Table 4–2 Login Server Create New User Example

Parameter	Sample Entry
User Name	<schema_of_node_b>
Password	<schema_of_node_b>
Confirm Password	<schema_of_node_b>

5. Click **Create**.

A new user for the Login Server is created.

See also: Oracle Portal Online Help topics: *What is a Login Server administrator?* and *Assigning a Login Server administrator*.

4.4.6 Step 6: Discover the Name of Each Node

You must have the name of each node if you plan on registering the node.

1. In Oracle Portal, log on to node a as required by entering the username and password.
2. In the **Node** portlet, click **Edit the Local Node**. By default, the Node portlet is located on the Oracle Portal home page's **Administer** tab.
3. Write down the name of the **Local Node**.

4. Close all open browser windows.

Note: You must close all browser windows before accessing node b. The Oracle Portal session cookie that was created by node a must expire and perform authentication on the appropriate node (node b).

5. Open a new browser window and repeat the above steps for node a by logging onto node b.

4.4.7 Step 7: Register Nodes Between Themselves

1. While on node b, register node a to node b.
 - a. In the Nodes portlet, click **Add a Remote Node**. By default, the Nodes portlet is located on the Oracle Portal home page's **Administer** tab.
 - b. Enter the configuration information for node a as required on the configuration page.

Table 4–3 Node a to node b registration information

Field	Example Entry
Remote Node Name	Name of the remote node (node a) obtained in the Section 4.4.6, "Step 6: Discover the Name of Each Node" .
Oracle Portal Database User	The schema owner for node a.
Oracle Portal Database Password	The schema password for node a.
Database Link Name	Oracle recommends that you leave this field blank. The default name is used when the database link is created on this page. Note that the default name is not displayed on this page.
TNS Name	The TNS Names alias (connect string) for the database on which node a is installed. Example: w816dev6
Remote Oracle Portal DAD	The DAD for node a created in Section 4.4.3, "Step 3: Edit Oracle Portal DADs" .

Table 4–3 Node a to node b registration information

Field	Example Entry
Remote Listener URL	The machine name on which the Oracle HTTP Server <i>powered by Apache</i> is installed. Example: OraclePortalsvr.company.com
Remote Listener Port	The port on which the Oracle HTTP Server <i>powered by Apache</i> is running for that node. Example: 5000

2. Click **OK**.
3. Quit all the browser windows.

Note: You must close all browser windows before accessing node a for the Oracle Portal session cookie that is created by node b to expire and perform authentication on the appropriate node (node a).

4. Repeat the above steps to register node b to node a.

When this step is completed successfully, the Oracle Portal nodes are fully configured to operate in the distributed environment.

4.4.8 Step 8: Refresh the Portlet Repository for Each Node

The providers for each node that is configured for a distributed Oracle Portal environment, can be used by the other node. However, the Portlet Repository needs to be refreshed on each node to see the providers and portlets created on remote nodes.

To refresh the portlet repository:

1. In the Oracle Portal Navigator, click the **Content Areas** tab.
2. In the Name column, click **Portlet Repository**.
The Portlet Repository Content Area is displayed.
3. Click **Refresh** in the upper right corner of the page.

Note: This operation may take a few minutes to complete because the Portlet Repository is refreshed for all the providers that are registered on the node.

Once this step is completed, the distributed portlets appear in the Portlet Repository. The providers that are not local (i.e. remote) are easily identifiable by their names which are prefixed with the name of the node to which they belong.

The distributed portlets are now displayed on the **Add Portlets** page and can be used when creating a page.

4. Repeat the above steps for each node in your distributed Oracle Portal environment.

4.4.9 Step 9: Create Additional Nodes

You can always create additional nodes to participate in the distributed Oracle Portal environment. For example, to register node c, complete the following steps in the order presented:

Table 4–4 *Creating additional nodes for distributed environment*

Step	For more information, see...
1. Create node c on a different database from that of node a and node b.	Section 4.4.1, "Step 1: Create Oracle Portal Nodes"
2. Create a DAD for node c.	Section 4.4.3, "Step 3: Edit Oracle Portal DADs"
3. Associate node c with the Login Server used by node a and node b.	Section 4.4.4, "Step 4: Associate Nodes with the Same Login Server"
4. Create a user for node c on the Login Server.	Section 4.4.5, "Step 5: Create a User on the Login Server with Administrator Privileges"
5. Register node c on node a.	Section 4.4.7, "Step 7: Register Nodes Between Themselves"
6. Register node c on node b.	Section 4.4.7, "Step 7: Register Nodes Between Themselves"
7. Register node a on node c.	Section 4.4.7, "Step 7: Register Nodes Between Themselves"

Table 4–4 *Creating additional nodes for distributed environment*

Step	For more information, see...
8. Register node b on node c.	Section 4.4.7, "Step 7: Register Nodes Between Themselves"
9. Refresh the portlet repository on node a, b, and c.	Section 4.4.8, "Step 8: Refresh the Portlet Repository for Each Node"

The registration of nodes must be symmetric. In addition, it is important to register the new node, in this case node c, on an existing node, either node a or node b, before registering an existing node on the new node. This is required to maintain the cookie encryption key used by the other nodes of the distributed environment.

Oracle9iAS Middle-tier and Firewall Configuration

In an enterprise deployment of Oracle Portal, it is typical to have a separate Oracle HTTP Server *powered for Apache* listener serving Login Server request and one for Oracle Portal requests. For performance reasons and connection pooling, it makes sense to configure the Login Server on a separate listener so that it is dedicated to authentication service only.

Architecturally, the Login Server is designed to be a central authentication service. As such, several Partner Applications exist within the enterprise and are hosted on various hosts. These Partner Applications are then configured to share a centralized Login Server.

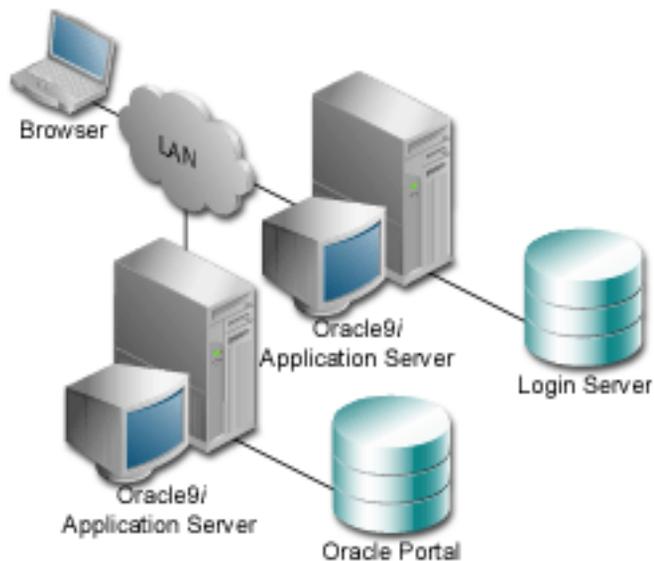
Specific topics covered include:

- [Oracle Portal as a Partner Application](#)
- [Configuring Virtual Hosts](#)
- [Working with Firewalls and Load Balancers](#)
- [Configuring Load Balancing Routers](#)
- [Tuning the Oracle HTTP Server](#)

5.1 Oracle Portal as a Partner Application

Oracle Portal is a Partner Application of the Login Server. The following figure illustrates such a configuration. Oracle Portal has a separate listener from the Login Server.

Figure 5–1 Oracle Portal as a Partner Application



You can set up Oracle Portal as a Partner Application in the following ways depending on preference, scalability, or manageability factors:

- Associating Oracle Portal to an existing Login Server (that you may not have any control over).
- Configuring a new Oracle Portal installation.
- Oracle Portal and the Login Server are serviced from separate middle-tier servers.

5.1.1 Associating Portal to an Existing Login Server That You Have No Control Over

If you do not have any administrative authority over the Login Server (for example, it is managed by the IT department), then you would associate an Oracle Portal node in the following way:

1. Inform the IT department that you need your Oracle Portal associated with the Login Server. Provide them with the following information:
 - Name of your application (arbitrary)
 - Home page URL

- Success URL
- Contact information – name and e-mail address

Note: The Success URL points to a Web page where the browser is redirected after a successful login. It must correspond to the procedure that processes the user identification information from the Login Server. Your Oracle Portal's Success URL must be the full URL to the `wwsec_app_priv.process_signon` procedure in your Oracle Portal's schema. For example:

```
http://server.domain.com:5000/pls/DAD/portal.wwsec_app_priv.process_signon
```

2. Based on this information, the IT department creates a new Partner Application entry on the Login Server for your Oracle Portal and provides you with the following information:
 - Site ID
 - Site Token
 - Encryption Key
3. Associate Oracle Portal to the Login Server by running the `ssodatax` script and entering the input information provided above. You need to know the following information to run the script:
 - Site ID
 - Site Token
 - Encryption Key
 - URL Prefix of the Login Server
 - Schema Name of the Login Server

Syntax

```
ssodatax <-i portal_site_id> <-t portal_site_token> <-k encryption_key> <-w portal_url> <-l login_server_url> <-s portal_schema> <-p portal_password> <-v cookie_version> <-o sso_schema> <-e pstore_schema> <-r pstore_password> <-b pstore_dblink> <-c connect_string> <-n ps_connect_string>
```

See also: For syntax and parameter descriptions, see [Section B.5, "Updating an Existing Portal Instance with the ssodatx Script"](#).

Example (Windows NT/2000)

```
ssodatx.cmd -i 1234 -t A1B2C3 -k X9Y8Z7 -w
http://webdbsvr.us.oracle.com:3000/pls/portal30/ -l
http://webdbsvr.us.oracle.com:3000/pls/portal30_sso/ -s portal30 -p portal30 -v
v1.1 -o portal30_sso -e portal30_sso_ps -r portal30_sso_ps -b portal30_dblink -c
orcl -n orcl01
```

After running the `ssodatx` script, your Oracle Portal is associated with the Login Server, each running on a separate listener providing you specified a different host name in the URL prefix for the Oracle Portal and the Login Server (`ssodatx` parameters `-w` and `-l` parameters respectively).

Note: If your Login Server is on a separate database instance, then you also need the Login Server administrator to provide you with the schema name, password, and connect string for the schema that your portal should use to access the Login Server's password store for External Applications. Your portal needs to create this link to support External Application providers which require access to mapped user's passwords to provide the single sign-on. These parameters are passed in the `ssodatx` script in the `-e`, `-r` and `-n` parameters, respectively. You also have the option of naming the database link with the `-b` parameter.

5.1.2 Associating Portal to an Existing Login Server That You Have Control Over

If you have full administrative control over the Login Server, then you can perform the association between your Oracle Portal and the Login Server using the `ssodatan` script, providing that they both reside on the *same* database instance.

If the database instances are on separate instances, complete the following steps first on your Oracle Portal and then on the Login Server.

These steps are summarized below:

1. Install a standalone Login Server by running the `linstall` script:

```
linstall <-o sso_schema> <-i pstore_password> <-s login_server_url> <-r random_
seed> <-p sys_password> <-u default_tablespace> <-t temporary_tablespace> <-d
```

```
document_tablespace> <-l logging_tablespace> <-c connect_string>
```

See also: For syntax and parameter descriptions, see [Section B.3, "Manually Installing a Login Server with the install Script"](#).

2. On the Oracle Portal Home Page, click the **Administer** tab.
3. In the **Services** portlet, click **Login Server Administration**.
4. Click **Administer Partner Applications**.
5. Click **Add Partner Application**.
6. In the **Partner Application Login** section, enter the Partner Application's name, the URL to the application's home page, and a success URL.
7. In the **Valid Login Timeframe** section, enter the dates when users can log on to the application through the Login Server. If you leave the **End Date** field blank, users can log on to the application indefinitely. In the **Application Administrator** section, enter an e-mail address and other information for the application's contact person or administrator.
8. Click **OK**. The new Partner Application appears in the Edit/Delete Partner Application list on the Partner Application page.
9. Run the `ssodatax` script and enter the input information generated above for your Oracle Portal.

See also: For syntax and parameter descriptions, see [Section B.5, "Updating an Existing Portal Instance with the ssodatax Script"](#).

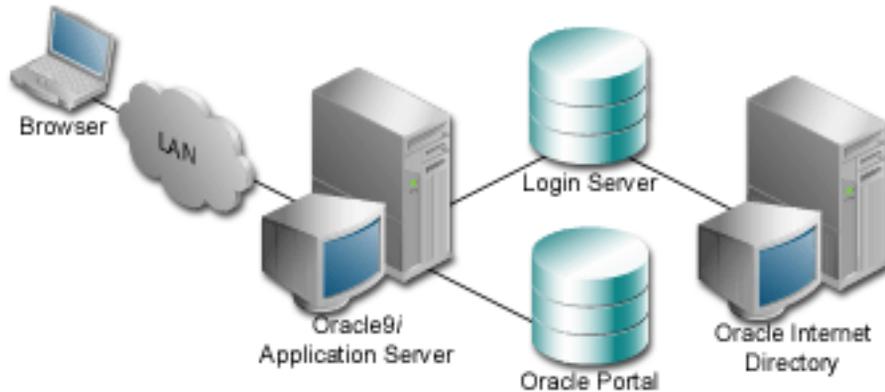
After running the `ssodatax` script, your Oracle Portal is associated with the Login Server. Each one is running on a separate listener, if you specified a different host name in the URL prefix for the Oracle Portal and the Login Server (`ssodatax` parameters `-w` and `-l` parameters respectively).

10. Repeat steps 2-10 on the Login Server.

5.1.3 Login Server Configured with Oracle Internet Directory (OID)

With any of the previously described configurations, you can also use the Oracle Internet Directory as a user repository.

Figure 5–2 Login Server configured with Oracle Internet Directory (OID)



In this architecture, instead of authenticating the user's credentials against a local table, the Login Server authenticates the user's credentials against an LDAP directory (OID). The Login Server makes LDAP API calls to the configured LDAP directory and authenticates the credentials against the directory.

Such a configuration requires that the Login Server be able to establish LDAP protocol communications between the database instance it resides in with the LDAP directory.

So, depending on the placement of the LDAP directory, the Login Server, and any firewalls, it must be noted that the machine on which the Login Server database schema resides needs to be able to have LDAP protocol access to the LDAP server. The default port for LDAP communication is port 389, however, it is configurable.

See also:

- *Configuring Oracle9iAS Portal for LDAP Authentication* at: <http://technet.oracle.com/products/iportal>.
 - *Oracle9iAS Single Sign-On Administrator's Guide* included in the Oracle9i Application Server documentation library.
-
-

5.2 Configuring Virtual Hosts

The Oracle9i Application Server HTTP server *powered by Apache* supports the configuration of virtual hosts. This allows a single machine and port to represent a

number of virtual hosts. To configure virtual hosts, you must set this up on both Oracle Portal as well as on the Oracle HTTP Server.

In our example, let's assume that we want to access Oracle Portal as `http://www.abc.com` as well as `http://www.xyz.com`. Also, let's assume that the Login Server's URL is `http://www.login.com`.

The steps for configuring virtual hosts are:

1. In `<ORACLE_HOME>/Apache/Apache/conf`, open and edit the Oracle HTTP Server configuration file, `httpd.conf`. Verify that the contents of the file includes the similar information in the Virtual Hosts section:

```
### Section 3: Virtual Hosts
NameVirtualHost 127.0.0.1

<VirtualHost 127.0.0.1>
    ServerName www.abc.com
</VirtualHost>

<VirtualHost 127.0.0.1>
    ServerName www.xyz.com
</VirtualHost>

<VirtualHost 127.0.0.1>
    ServerName www.login.com
</VirtualHost>
```

See also: [Section A.1.1, "Oracle HTTP Server Configuration File \(httpd.conf\)"](#)

This example uses the IP address `127.0.0.1`, which represents the local machine. This can be any valid IP address.

The domain names specified in the `ServerName` entries need to be valid domain names. If you are setting up Oracle Portal on a local laptop, make the appropriate entries in your local hosts file.

```
# Copyright (c) 1993-1995 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP
# for Windows NT.
#
127.0.0.1 localhost
```

```
127.0.0.1 www.abc.com
127.0.0.1 www.xyz.com
127.0.0.1 www.login.com
```

See also: [Section 3.1, "Configuring Oracle Portal on a Standalone Laptop"](#)

2. For Single Sign-On on the Login Server to work properly, it must always be referenced by any Partner Application with the same hostname in the URL. This is because cookies are sent back only to the host that generated them. So, in the preceding example, the Login Server must always be referenced as `http://www.login.com`.

Thus, you must register `www.abc.com`, `www.xyz.com`, and `www.login.com` as Partner Applications.

- a. Log on to the Login Server directly as an administrator with Full Privileges on the Login Server.
- b. Add a Partner Application entry for `www.abc.com`.
- c. Add a Partner Application for `www.xyz.com`.
- d. Add a Partner Application for the Login Server, `www.login.com`.
- e. Run the `ssodatax` script to create configuration entries on the Oracle Portal for each of these entry points.

After running the `ssodatax` script for each of the Partner Applications, the aliases should be correctly configured.

See also: [Section B.5, "Updating an Existing Portal Instance with the `ssodatax` Script"](#)

5.3 Parallel Page Engine Configuration

The Oracle 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, Oracle8i database, and Oracle Portal) to be upgraded individually as required.

5.3.1 Configuring Parallel Page Engine Parameters

When a page is requested from Oracle 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 Parallel Page Engine obtains the page metadata from the Portal Repository and is responsible for assembling the portlets on the page. You can tune the Parallel Page Engine for better performance by adding any of the following optional parameters in the `zone.properties` file.

See also: [Section A.1.2, "JServ Configuration File \(zone.properties\)"](#)

Table 5–1 Parallel Page Engine (PPE) parameters

Parameter	Description
<code>logpath=<PATH></code>	Instructs the Parallel Page Engine on where to log messages during execution. The default is the JServ log path.
<code>logmode=<debug></code>	Enables the Parallel Page Engine to run in debug mode. If this value is not set, the Parallel Page Engine runs in normal mode.
<code>showError=<TRUE/FALSE></code>	Enables and disables the display of error messages in the Oracle Portal user interface. The default is <code>TRUE</code> .
<code>poolSize=<some number></code>	Defines the total number of parallel fetchers to use in the page execution. The default is 25.
<code>stall=<duration in sec></code>	Specifies the maximum amount of time, in seconds, that a fetcher should wait for a request to complete. The default is 120 seconds.
<code>requesttime=<duration in sec></code>	Specifies the maximum amount of time, in seconds, that a fetcher should wait for an individual request to respond. Once a request starts responding, the fetcher waits for the rest of the data until the time specified by <code>stall</code> . The default is 40 seconds.
<code>httpsports=<port1>:<port2>:<...><portn></code>	Specifies which ports are configured for HTTPS.
<code>prefix=<PLSQL prefix like /pls></code>	Contains the PL/SQL prefix path value. The default is <code>/pls</code> .

Table 5–1 Parallel Page Engine (PPE) parameters

Parameter	Description
<code>offlinePath=</path/offlin efile></code>	Allows you to take Oracle Portal offline for maintenance. The contents of the offline file are sent to all page requests.
<code>proxyHost=<hostname></code>	Defines the host to use for requests which need to go through a proxy server.
<code>proxyPort=<proxyPort></code>	Specifies the port to use for requests which need to go through a proxy server.
<code>proxyIgnore=<domain1>&<do main2>&. . .<domainn></code>	Specifies those domains you want ignored for the proxy settings. According to the HTTP 1.1 standard, domains are required to start with a ". ". For example, <code>.oracle.com</code> is valid, <code>oracle.com</code> is not.
<code>useScheme=<http / https></code>	Forces the Parallel Page Engine to use the protocol specified on this line.
<code>usePort=<port number></code>	Forces the Parallel Page Engine to always use the specified port number for all requests.
<code>cacheBuffer</code>	Specifies the size of the memory buffer to use to return a cached page from the middle tier. Set this value as close as possible to the actual byte size of a complete page. If the value is set too small multiple reads are performed to the disk. (This can be inefficient.) The default is 32768 bytes.

See also: *Page Generation and Assembly Scalability in Oracle9iAS Portal* at: <http://technet.oracle.com/products/iportal>.

5.4 Working with Firewalls and Load Balancers

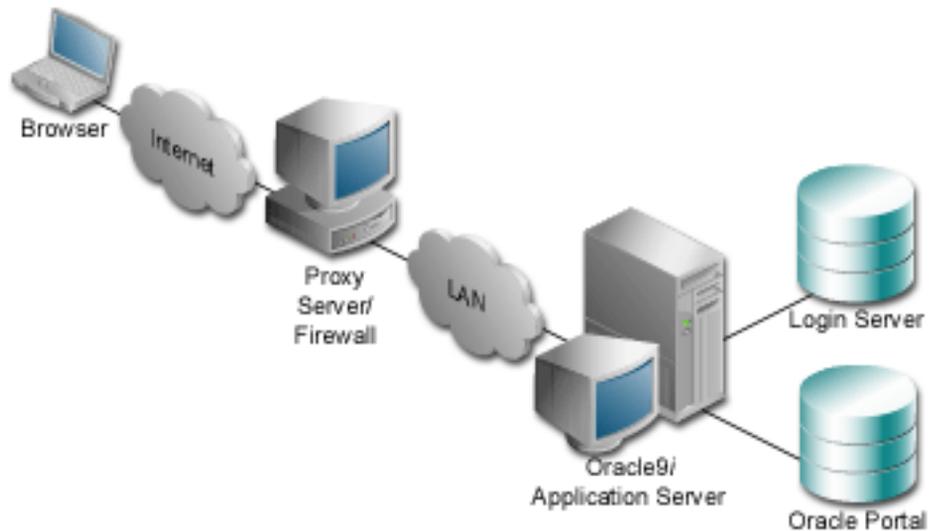
Once Oracle Portal is deployed for access over the Internet, there are typically other network devices that may complicate the configuration such as firewalls, reverse proxies, server farms, and so on.

5.4.1 Configuring Reverse Proxy Servers Over the Internet

A reverse proxy server is a host process which is used as part of a firewall architecture to isolate the internal hosts from the externally accessible host(s) by providing a proxy through which external requests must pass to access internal

services. Typically, such a proxy server takes the form of a dual-homed host. This means that it is a host with two network interface cards. One interface connects to the external network, and the other interface connects to the internal network, or demilitarized zone (DMZ) of the firewall.

Figure 5–3 Internet configuration with reverse proxy server



In this architecture, the browser accesses the server through a hostname which is published by the proxy server. The proxy server then forwards the request to the actual host within the firewall, which could be some other host name.

For this example, let's assume the following:

- The published address is `www.myportal.com`
- Internal to the firewall, the server name for the Oracle9i Application Server middle-tier is actually `server.company.com`.
- Externally, the server is addressed with the default port 80; however, internally, the `server.company.com` is listening on port 7777.

Note: This configuration is only possible with Oracle Portal version 3.0.7 and above. Prior to 3.0.7, Oracle Portal was not using the `ServerName` specified by the Oracle HTTP Server *powered by Apache* configuration. Instead, it was using the `Host` value in the HTTP request. Thus, you could not control the generation of self-referential URLs.

Complete these steps to configure Oracle Portal for this architecture:

1. Define configuration settings that allow the Oracle9i Application Server middle-tier to listen on port 7777, but assert to the server that it is using port 80.
2. Create `VirtualHost` entries that accept the internal host name, but then assert the externally visible host name, using the `ServerName` directive, so that self-referential URLs rendered on the Oracle Portal pages are valid for the browser.
3. Edit the hosts file on the internal middle-tier servers to define the IP addresses for the `ServerNames` asserted above, so that they can resolve the hostnames that are generated by Oracle Portal, for HTTP calls looping back to fetch portlet content.

See also: [Section A.1.7, "Local HOSTS File"](#)

4. Run the `ssodatan` or `ssodatax` script(s), as appropriate, using the externally published server names; for example, `www.myportal.com`

See also:

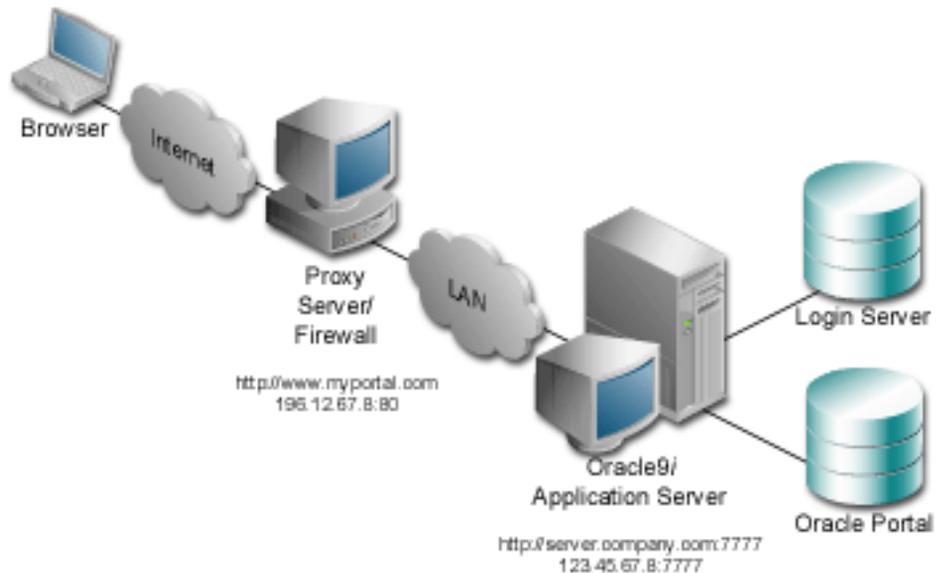
- [Section B.4, "Configuring a New Oracle Portal Instance and Login Server with the `ssodatan` Script"](#)
 - [Section B.5, "Updating an Existing Portal Instance with the `ssodatax` Script"](#)
-
-

5. Register the `www.myportal.com` domain name on a domain name server on the Internet, with IP address 196.12.67.8.

Note: The IP addresses used in this example are for illustration purposes only and may not be valid IP addresses.

The following figure shows the server names and ports used in the preceding example:

Figure 5–4 Example of reverse proxy server configuration



5.4.2 Configuring Oracle HTTP Server

You provide directives in the Oracle HTTP Server configuration file, `httpd.conf`, which specify the behavior described in the points above. The reverse proxy server contacts the internal middle-tier server as `server.company.com` over port 7777.

When the Oracle HTTP Server invokes the PL/SQL Gateway (`mod_plsql`), and `mod_jserv`, it then passes `www.myportal.com` as the `ServerName` and port 80. URLs that are generated by Oracle Portal code use `www.myportal.com` and port 80.

The directive "useCanonicalName on" instructs Apache to use the `ServerName` specified. Otherwise, it passes the name used in the `Host` field of the request.

The relevant sections in the `httpd.conf` file for the `server.company.com` Oracle9i Application Server configuration are shown in the following.

```
### Section 2: 'Main' server configuration
...
Port 80
Listen 7777
Listen 80

ServerName www.myportal.com
...
UseCanonicalName On
...
### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on
# your machine you can setup VirtualHost containers for them.
#
# If you want to use name-based virtual hosts you need to define at
# least one IP address (and port number) for them.
#

# This section is mandatory for URLs that are generated by
# the PL/SQL packages of the Oracle Portal
# These entries dictate that the server should listen on port
# 7777, but will assert that it is using port 80, so that
# self-referential URLs generated specify www.myportal.com:80
# This will create URLs that are valid for the browser since
# the browser does not directly see the host server.company.com.
NameVirtualHost 123.45.67.8:7777

<VirtualHost server.company.com:7777>
ServerName www.myportal.com
Port 80
</VirtualHost>

# Since the previous virtual host entry will cause all links
# generated by the Oracle Portal to use port 80, the server.company.com
# server needs to listen on 80 as well since the Parallel Page
# Engine will make connection requests to Port 80 to request the
# portlets.

NameVirtualHost 123.45.67.8:80

<VirtualHost server.company.com:80>
```

```
ServerName www.myportal.com
Port 80
</VirtualHost>
```

If you need to support multiple aliases for the published address `www.myportal.com`, then some additional directives are needed. For example, if the server also needs to be accessible as `www.portal.com`, then you need to define additional virtual host entries on the internal server. This is so the reverse proxy directs requests from each corresponding published hostname to a related internal host alias which can assert the correct published name.

In this example, the `VirtualHost` sections appear as follows:

```
NameVirtualHost 123.45.67.8:7777

<VirtualHost server.company.com:7777>
ServerName www.myportal.com
Port 80
</VirtualHost>

<VirtualHost server2.company.com:7777>
ServerName www.portal.com
Port 80
</VirtualHost>

NameVirtualHost 123.45.67.8:80

<VirtualHost server.company.com:80>
ServerName www.myportal.com
Port 80
</VirtualHost>

<VirtualHost server2.company.com:80>
ServerName www.portal.com
Port 80
</VirtualHost>
```

5.4.3 Resolving Domain Names

A local `HOSTS` file can help resolve domain names that are not normally visible to the internal network. For example, the Oracle9i Application Server host for `server.company.com` makes requests to itself, but the URLs that it is requesting are referring to `www.myportal.com`. You must create host entries in the local

HOSTS file on that machine allowing it to resolve this name, within the firewall. The hosts entry for this example should include the following lines:

```
# This is a sample HOSTS file used by Microsoft TCP/IP
# for Windows NT.
#
127.0.0.1    localhost
123.45.67.8 www.myportal.com
```

If you do not provide these entries in the local HOSTS file, then you need to set the Oracle9i Application Server host to recognize a proxy server that would take the request out to the Internet and back in through the reverse proxy (`www.myportal.com`). Avoid this setup as this may result in poor performance.

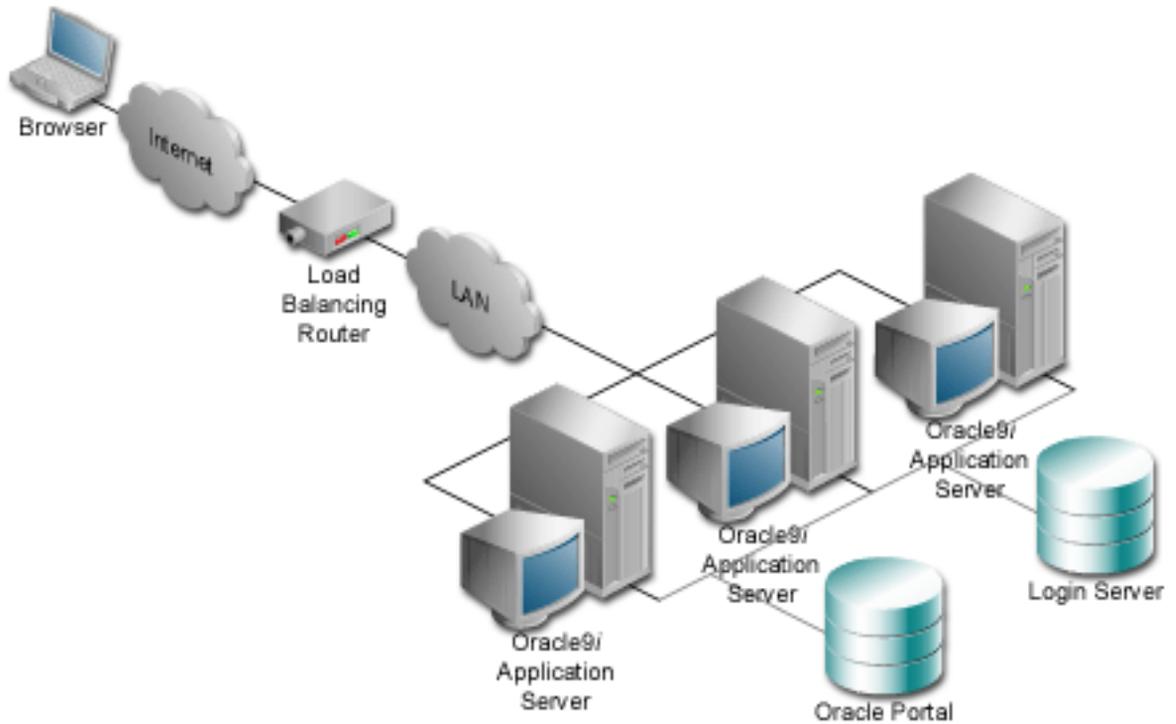
Note: On some platforms, such as HP, there is a file which indicates the search order that should be applied to the sources for IP name mapping. For the preceding example to work, if such a file exists on your platform, make sure that it specifies the local hosts file to be checked for IP mapping, before any DNS servers.

5.5 Configuring Load Balancing Routers

The purpose of a Load Balancing Router (LBR) is to provide a single published address to the client browsers, and provide a "farm" of Web servers which actually service the requests, based on the distribution of the requests done by the LBR. The LBR itself is a very fast network device which can distribute Web requests to a large number of physical servers.

If you want to install multiple Oracle9i Application Server middle-tier servers to handle a large load, you could configure Oracle Portal as illustrated in the following diagram:

Figure 5-5 Load balancing router configuration



This example shows that each of the servers can handle requests for either the Login Server or Oracle Portal. Each of the middle-tier servers must have DAD entries for each of the databases. A good way to accomplish this is to have the middle-tier servers share a file system that contains the configuration information for the DADs, and allows them to share cache files.

The important points to consider with this configuration include:

- The Internet DNS maps the name `www.myportal.com` to the external IP address on the LBR.
- The LBR performs load balancing of requests to `www.myportal.com` to `svr1.company.com`, `svr2.company.com`, and `svr3.company.com`, addressing the request to their IP addresses, but still containing `www.myportal.com` in the Host: field of the HTTP request.

- Each of the middle-tier hosts accepts requests to `www.myportal.com`, and their `httpd.conf` files assert that name as the `ServerName`. Hence the names `svr1`, `svr2`, and so on are irrelevant.
- The local hosts files on `svr1`, `svr2`, and `svr3` contain the entry `www.myportal.com`, each pointing to its own IP address.
- Unless your LBR does port mapping, you should configure the internal servers to use the same ports as the LBR.
- Optimal cache utilization can be realized by mounting a shared file system on which to write the cache files. If you decide not to have the middle-tier servers share a cache directory, caching will still work, but with a lower hit ratio.

5.5.1 Placing a Firewall Between the Middle-tier and the Database

It is fairly typical to have a firewall with a SQL*Net proxy between the application server and the database for Oracle installations. Keep in mind though that the Oracle Portal architecture requires HTTP connects from the database to the middle-tier servers, for example, when the Oracle Portal makes an HTTP request to the Login Server to get the list of external applications. Or, when the Oracle Portal repository makes an HTTP request to a particular provider which may be a Web provider, potentially outside the Intranet firewall. Keep this communication path in mind when planning where firewalls should go and what traffic should be allowed through them.

You need to allow HTTP traffic to pass on the ports that are being used, through any firewall set up between the middle-tier and the database on which the Oracle Portal code resides.

Similarly, if the Login Server is setup for LDAP authentication, then LDAP traffic must be allowed to reach the LDAP server. The LDAP calls are made from the Login Server database instance.

See also: [Section 5.5.1, "Placing a Firewall Between the Middle-tier and the Database"](#)

5.6 Tuning the Oracle HTTP Server

However you choose to configure the Oracle HTTP Server *powered by Apache* listener, you can optimize performance by setting an approximate number of simultaneous requests that can be handled by the Apache listener.

On UNIX, in particular, since Apache is process-based, each process needs to open a database connection for each DAD that has requested it. As a result, the number of requests can be quite high, which may result in clients being "locked out" if the number of sessions allowable has been exceeded. However, setting too high of a value unnecessarily consumes resources.

The scenario is described below:

1. For every service request from an Oracle Portal DAD, there is one network connection and two sessions (the two sessions use the same physical connection).

The first session is for "portal30" and the second is for "portal30_public".

2. If you are logging into Oracle Portal, then you'll need to open a connection for the Login Server DAD (SSO DAD). This will consume one network connection and two sessions.

In this case, the first session is for "portal30_sso" and the second session is for "portal30_sso_public".

3. The Apache configuration setting that determines the maximum number of requests being handled simultaneously is named `MaxClients`. It defaults to 150.

If each user were logging in and working in Oracle Portal, then scenario (1) and (2) above would result in four sessions per process. The total number of sessions for such a scenario is calculated as follows:

$$150 * 4 = 600$$

600 sessions and approximately 300 database connections (2 sessions per connection)

5.6.1 Configuring the MaxClient Setting

Since login frequency is generally lower than Oracle Portal access frequency, it makes sense to configure the Login Server on a separate Oracle HTTP Server *powered by Apache* listener. The objective is to tune down the `MaxClient` setting to a value that is reasonable without affecting the needs of the portal system.

Oracle Portal makes extensive use of Apache `mod_plsql` which maintains a pool of connections to the database. The `MaxClient` parameter tunes the number of Apache processes which directly relates to the number of database connections pooled by `mod_plsql`.

Note: For more information, see *Using the PL/SQL Gateway* which is provided as part of the Oracle9i Application Server documentation set.

1. For the Login Server's listener, once you've determined the approximate value to set for the `MaxClients` parameter, edit this accordingly in the Apache 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 Apache services on the Apache listener. This ultimately controls the maximum number of sessions that could be established.

2. For the Oracle Portal listener, you can separately tune the `MaxClients` parameter according to the needs of the Login Server and the needs of Oracle Portal, without affecting each other. This parameter directly corresponds to the number of sessions established and to the maximum workload that the Apache listener can handle on the Portal listener.

The `MaxClients` section in the `httpd.conf` file is shown below:

```
# Limit on total number of servers running, i.e., limit on the number
# of clients who can simultaneously connect --- if this limit is ever
# reached, clients will be 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 separately, you will have a separate listener for Oracle Portal and the Login Server. The former controlling the resources (sessions) on the portal database and the latter controlling the resources on the Login Server database.
 - The number of sessions and connections that the database permits is limited by the value set in the Oracle8i database's `init.ora`. Refer to the Oracle8i database documentation library for more information.
-
-

Setting up the Search Feature in Oracle Portal Content Areas

This chapter provides information on setting up the built-in *interMedia* Text search capabilities in Oracle Portal content areas.

This chapter contains the following sections:

- [New Search Features](#)
- [Prerequisites](#)
- [Searching in Oracle Portal](#)
- [interMedia Text Performance](#)
- [Setting up interMedia Text Searching](#)
- [Setting up interMedia Text Indexes](#)
- [Dropping an interMedia Text Index](#)
- [Setting up Your Environment for interMedia Text](#)
- [Multilingual Functionality \(Multilexer\)](#)
- [interMedia Text-related Procedures Created in Oracle Portal](#)

See also:

- For *interMedia* Text documentation, see your Oracle8*i* database documentation library.
- For *interMedia* Text technical papers, training materials, code samples, and so on, visit:

<http://technet.oracle.com/products/text/>

6.1 New Search Features

This release of Oracle Portal includes the following new features for search:

- Search results are now displayed in a portlet. The portlet can be added to any portal page (a default Search Results page is provided). The portlet can be customized to control the information that appears in each listing.
- Portal administrators can also replace the default Advanced Search page with a custom page. The custom page is called whenever the Advanced Search link is clicked.

To access these new features, click **Search Settings** in the Services portlet. By default, the Services portlet is located on the Oracle Portal home page's Administer tab.

6.2 Prerequisites

You must be logged on as an Oracle Portal administrator to configure *interMedia* Text, and create, alter, update, and drop *interMedia* Text indexes.

Before using *interMedia* Text in Oracle Portal, perform the following tasks:

- Install and configure the Oracle8*i* database with the *interMedia* Text option by running the Oracle Universal Installer (OUI) and the Database Configuration Assistant. See the Oracle8*i* installation documentation for more information about installing and configuring an Oracle8*i* database to use *interMedia* Text.

Important: If you are planning to enable *interMedia* Text in Oracle Portal on Windows NT/2000, the following requirements apply:

- Oracle Portal must be installed in an Oracle 8.1.7 database.
 - Disable connection pooling from the Database Access Descriptor page. For more information, see *Using the PL/SQL Gateway* which is provided as part of the Oracle9i Application Server documentation set.
-
-
- Install Oracle Portal according to the instructions provided in the *Oracle9i Application Server Installation Guide* for your particular operating system.
 - Create an Oracle Portal content area using the Content Area Creation Manager. See the Oracle Portal Online Help topic: *Creating a content area*.
 - If this database was an existing ConText site, make sure to remove `text_enable` from the `init.ora` file. It is no longer used in Oracle8i, and actually prevents Oracle8i from operating properly. The error, "cannot find package DR_REWRITE" may appear.

See also: For a complete list of system requirements, see [Chapter 1, "Verifying Requirements"](#).

6.2.1 *interMedia* Text on UNIX

For the *interMedia* Text feature to work, set the following UNIX environment variables before starting the Oracle8i database and Net8 Listener.

Korn Shell (ksh)

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/ctx/lib
export LD_LIBRARY_PATH
```

C-Shell (csh)

```
setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${ORACLE_HOME}/ctx/lib
```

6.3 Searching in Oracle Portal

The main points to know when searching in Oracle Portal content areas include:

- *interMedia* Text searching in Oracle Portal is available in content areas only. Searches on Oracle Portal pages is performed from the Oracle Portal Navigator and searches on applications (and content areas) is performed with the Oracle Portal Search portlet. See the Oracle Portal Online Help topic: *Adding search capabilities on your page*.
- When search is enabled in Oracle Portal, search is enabled for *all* content areas created in your Oracle Portal installation. It cannot be enabled on one content area and disabled on another content area. Search results, however, are restricted to the content area on which the search is performed.
- There are two levels and two modes of searching in Oracle Portal content areas depending on the type of search you use: basic search, advanced search, and *interMedia* Text search. The distinctions are summarized in the following table:

Table 6–1 *interMedia* Text search types

	Basic Search	Advanced Search
Availability	From Search field on Navigation Bar	From Advanced Search screen
<i>interMedia</i> Text is disabled	<ul style="list-style-type: none"> ■ Searches on the item attributes such as display name, description, and keywords of items. ■ Automatically uses wildcards. <p>Example: If you search on "sing," the results of the search include: sing, single, tossing.</p>	
<i>interMedia</i> Text is enabled	<ul style="list-style-type: none"> ■ Searches on the item attributes such as display name, description, and keywords of items. ■ Searches in the content of documents and URLs. ■ Uses STEM searching. <p>Examples:</p> <ul style="list-style-type: none"> ■ If you search on "distinguish," the results of the search include: distinguish, distinguished, distinguishes. ■ If you search on "sing," the results of the search include: sing, sang, sung. 	

6.3.1 Basic Search

This type of search looks for the specified words in the item attributes such as the display name, description, and keywords of items, as well as the display name and description of folders, categories, and perspectives are searched. If *interMedia* Text

is enabled, basic search also looks in the content of documents and URLs. A search results page displays all items with this criteria.

Typically, a Search field appears on the Navigation bar enabling end users to perform a basic search as shown below:

Figure 6–1 Basic search screen in Oracle Portal



See also: The Oracle Portal Online Help topics: *Performing a basic search*, *Setting up the search feature*, and *Editing navigation bars*.

6.3.2 Advanced Search

With advanced search (which is always enabled), you can:

- Find content that contains any of the specified words.
- Search across a content area other than the current content area, or across all content areas.
- Restrict the search to a particular folder, category, perspective, item type, or attribute.
- If *interMedia* Text is enabled, searches for near, soundex, and fuzzy items are also searched.

Figure 6–2 Advanced search screen in Oracle Portal

Operator Search For
 Contains All basic Submit Reset

For advice on how to search, read the [Search Tips](#).
 For a simple search, use [Basic Search](#).

Search In Content Area Sample Content Area Folder Include Sub-folders
 Language English
 Restrict To Category All Perspective All
 Return Only Type All Folders Items of Type All Item Types
 Search Attributes Match All Any

Attribute Name	Operator	Value
<input type="text"/>	Contains	<input type="text"/>
<input type="text"/>	Contains	<input type="text"/>
<input type="text"/>	Contains	<input type="text"/>
<input type="text"/>	Contains	<input type="text"/>

More Attributes

See also: The Oracle Portal Online Help topic: *Performing an Advanced Search*.

6.3.3 *interMedia* Text Search

As discussed earlier, Oracle Portal has built-in support for *interMedia* Text indexing. It is worth repeating that when search is enabled in Oracle Portal, search is enabled for *all* content areas created in your Oracle Portal installation. It cannot be enabled on one content area and disabled on another content area. Search results, however, are restricted to the content area on which the search is performed. The search is performed on the actual content in documents such as PDF, PowerPoint, and Word as well as the contents on URL pages, text, and HTML.

If *interMedia* Text is *not* enabled, end users can always perform a basic or advanced search in the content area.

6.3.4 Viewing *interMedia* Text Search Results

If themes and gists are enabled from the Search Settings page (see [Figure 6-4](#)), then you can access the themes and gists for documents returned by a search from the search results. You can:

- View an HTML version of the file.
- View an HTML version of the file with search terms highlighted in any color or font set by the Oracle Portal administrator.
- View major themes in a chart (document theme analysis).
- View a short summary about the file's content (gist).

See also: [Section 6.5.2, "Step 2: Enable *interMedia* Text Searching"](#)

6.4 *interMedia* Text Performance

interMedia Text performance may be affected by the following query, indexing, and update considerations:

6.4.1 Query Considerations

How does the size of my data affect queries?

The speed at which the text index can deliver ROWIDs is not affected by the actual size of the data, but by the size of the Token Table which holds the list of words, and information about the rows in which they appear. Text query speed will be related to the number of rows that must be fetched from this Token Table, and the length of each row.

Thus, it should be nearly as fast to find a rare word in a large document set as it is to find a common word (or many uncommon words) in a smaller document set.

How does the source type of my data affect queries?

The format of the documents (for example, plain ASCII text, HTML or Microsoft Word) should make no difference to query speed. The documents are filtered to plain text at indexing time, not query time.

The "cleanliness" of the data makes a difference. Spell-checked and sub-edited text for publication tends to have a much smaller total vocabulary (and therefore size of

token table) than informal text such as e-mails, which contain many spelling errors and abbreviations to bloat the token table.

6.4.2 Indexing Considerations

How long should indexing take?

Indexing text is a resource-intensive process. Obviously, the speed of indexing depends on the power of the hardware involved, but you should expect somewhere between 50MB per hour on workstation-class NT machine (approximately 400MHz CPU, 128MB memory) to more than 1GB per hour on a large multi-CPU, multi-gigabyte server machine. The latter figure assumes you are using parallel indexing on a partitioned table (a new option for 8.1.6).

For most real-life systems, the time to index a complete table of documents will be measured in hours, and in some cases days.

How do I track the progress of the indexing process?

You can use the `ctx_output.start_log <filename>` command to log output from the indexing process. The `<filename>` will normally be written to `$ORACLE_HOME/ctx`, but you can change the directory using the `log_directory` parameter in `ctx_adm.set_parameter`.

Otherwise, for a course-grained answer, you can count the number of rows in the `DRxxxK` table. There will be one row in here for each row that has been indexed. However, these rows are only committed when the indexing process runs out of indexing memory and does a "flush" to the database. It is even possible that this will never happen until indexing is complete.

How much disk space overhead will the indexing require?

The overhead (the amount of space needed for the `DR$` index tables) varies between approximately 25% of the original text volume, and 100%. Generally, the larger the total amount of text, the smaller the overhead, but many small records will use more overhead than fewer large records. Also, "clean" data (such as published text) will require less overhead than "dirty" data such as e-mails or discussion notes, since the "dirty" data is likely to include many unique words from misspellings, abbreviations, and so on.

Theme indexes are generally much smaller than text indexes. Creating a theme index only will generally require very little storage, but creating a text index only will not save you much space over a combined index, though it is likely to be significantly faster.

How does the format of my data affect indexing?

Looking at indexing overhead, you can expect much lower overheads for formatted documents (for Microsoft Word files) since such documents tend to be very large compared to the actual text held in them.

So 1GB of Word documents might only require 50MB of index space, whereas 1GB of plain text might require 500MB, since there is ten times as much "plain text" in the latter set.

Indexing time is harder to determine. Although the reduction in the amount of text to be indexed will have an obvious effect, we must balance this out against the cost of filtering the documents. In general, these will roughly cancel out, so the time to index 1GB of formatted docs will be about the same as to index 1GB of plain text, although it may be a little longer.

6.4.3 Update Considerations

How often should I index new or updated records?

How often do you need to? The less often you run re-indexing (via the command `ALTER INDEX indexname REBUILD ONLINE PARAMETERS('SYNC')`) then the less fragmented your indexes will be, and the less you will need to optimize them. However, this means that your data will become progressively more out of date, which may be unacceptable for your users.

Many systems can handle overnight indexing. This means data that is less than a day old is not searchable. Other systems use hourly, ten minute, or five minute updates.

Note: The Context Server (ctxsrv) has been deprecated and should no longer be used. Use `drbgdml.sql` instead.

How can I tell when my indexes are getting fragmented?

The best way is to time some queries, run index optimization, then time the same queries. You must restart the database to clear the SGA each time. If the queries speed up significantly, then optimization is worthwhile.

A more scientific method involves counting the number of rows for each term in the `DRxxxI` table:

```
SELECT AVG(COUNT(*)) FROM DR$index_name$I
      GROUP BY TOKEN_TEXT HAVING COUNT(*) > 1;
```

Note: Ignore all words with only a single row in the index table.

A value greater than 10 from this query may indicate the need to optimize the index, but experimentation should yield the best value in any particular circumstances. Very large tables will inevitably have a lot of rows where the `TOKEN_INFO` data overflows the 4K internal limit, so you would expect the average to be greater on large systems.

For more information: *interMedia* Text Performance FAQ at:
<http://technet.oracle.com/products/text/>

6.5 Setting up *interMedia* Text Searching

There are four main steps for setting up *interMedia* Text in Oracle Portal:

- [Step 1: Set up the Global Page Settings](#)
- [Step 2: Enable *interMedia* Text Searching](#)
- [Step 3: Create an *interMedia* Text Index](#)
- [Step 4: Maintain an *interMedia* Text Index](#)

6.5.1 Step 1: Set up the Global Page Settings

The first step requires you to configure the global page settings in the following way:

1. In the **Services** portlet, click **Global Settings**. By default, the Services portlet is located on the Oracle Portal home page's **Administer** tab.

Figure 6–3 Global page settings - Proxy Server

Proxy Server
Enter the proxy server settings. The proxy server is used to connect outside of a firewall.

HTTP Server	<input type="text"/>
HTTP Server Port	<input type="text"/>
No Proxy Servers for Domains beginning with	<input type="text"/>
URL Connection Time-Out (seconds)	<input type="text"/>

2. In the **Proxy Server** section:
 - a. Enter the host name of your proxy server for the HTTP Server.
Note: Do not prefix `http://` to the proxy server name.
 - b. Enter the domains that you do not want redirected to the proxy server.
 - c. Enter the maximum amount of time (in seconds) that a connection should be attempted before timing out.
3. Click **OK**.

6.5.2 Step 2: Enable *interMedia* Text Searching

Before creating the *interMedia* Text indexes, configure the *interMedia* Text settings in Oracle Portal in the following way:

1. In the **Services** portlet, click **Search Settings**. By default, the Services portlet is located on the Oracle Portal home page's **Administer** tab.

Figure 6–4 Services portlet - interMedia Text properties

Specify *interMedia* Text Properties

Select whether or not to enable *interMedia* Text searching and themes and gist for your content areas. *interMedia* Text searching enables users to search the actual content of items as well as their titles, descriptions and keywords. Themes and gists provide additional contextual information about the items returned by a search. You can also choose the color and style to use highlight search words in the items returned. For *interMedia* Text searching to work, *interMedia* Text must be supported and installed on your database and you must create an index by clicking Create Index. If the button reads Drop Index, an index already exists which you can drop if desired.

Enable *interMedia* Text Searching

Enable Themes And Gists

Highlight Text Color:

Highlight Text Style:

Note: If you see the message, "*interMedia* Text is not installed", *interMedia* Text was not installed with the database and is not available for your content areas. Arrange with your database administrator to have *interMedia* Text installed. After it is installed, you need to run the following command in SQL*Plus:

```
inctxgrn.sql
```

This file is located in the <ORACLE_HOME>/portal30/src/wws directory.

Log on using the user name and password for the schema that owns the Oracle Portal content area. For example, if the schema name is "SCOTT", log on with the user name "SCOTT" and the appropriate password.

2. Select **Enable *interMedia* Text Searching** to make *interMedia* Text searching available in your content areas.
3. Select **Enable Themes And Gists** to create a theme and gist for each item returned by the search.

- A theme shows the nouns and verbs that occur most frequently within the item.
- A gist displays a brief summary of the item, derived from how frequently those nouns and verbs appear.

Note: Themes and gists are not available for all languages.

4. In the **Highlight Text Color** list, choose the color to highlight the search words in the HTML renditions of the items returned by the search.
5. In the **Highlight Text Style** list, choose the style to apply to the search words in the HTML renditions of the items returned by the search.
6. Click **OK**.

Now that you've enabled *interMedia* Text, you'll need to create a new *interMedia* Text index.

6.5.3 Step 3: Create an *interMedia* Text Index

To create an *interMedia* Text index:

1. Make sure that you have configured the *interMedia* Text settings in Oracle Portal as discussed in the previous section, [Section 6.5.2, "Step 2: Enable *interMedia* Text Searching"](#).
2. In the **Specify *interMedia* Text Search Properties** section, as shown on the previous page, click **Create Index**.
 - If properly run, the message "Index created" appears and the *interMedia* Text index is created on the server.
 - If you fail to create the index, check that your system has met all the requirements in [Section 6.2, "Prerequisites"](#).

Note: You can also run the following command in SQL*Plus:

```
ctxcrind.sql
```

This file is located in the <ORACLE_HOME>/portal30/src/wws directory.

Log on using the user name and password for the schema that owns the Oracle Portal content area. For example, if the schema name is "SCOTT", log on with the user name "SCOTT" and the appropriate password.

The following *interMedia* Text indexes are created:

Table 6–2 *interMedia* Text indexes created

Index Name	Description
WWSBR_CORNER_CTX_INDX	Index based on folders.
WWSBR_DOC_CTX_INDX	Index based on the content of uploaded documents.
WWSBR_PERSP_CTX_INDX	Index based on perspectives.
WWSBR_THING_CTX_INDX	Index based on items.
WWSBR_TOPIC_CTX_INDX	Index based on categories.
WWSBR_URL_CTX_INDX	Index based on the objects' Uniform Resource Locator (URL).

Note: The time required for creating indexes varies depending on the number of items you have in your content area.

6.5.3.1 Creating indexes in SQL*Plus

You can also create indexes in SQL*Plus as follows:

1. On the database where Oracle Portal is installed, log on to SQL*Plus with the appropriate username and password for the portal schema. For example:

```
sqlplus portal30/portal30
```

2. Enter the following command:

```
set serverout on
declare
  l_message varchar2(2000);
begin
  wwv_context.createindex(p_language => 'us', p_message => l_message);
  dbms_output.put_line(l_message);
end;
/
```

When the index is created, the message, "*interMedia* Text indexes created successfully" appears.

See also: If you encounter problems creating an *interMedia* Text index, see "[Problem: Unable to create interMedia Text indexes.](#)" in [Chapter 8, "Troubleshooting"](#).

6.5.4 Step 4: Maintain an *interMedia* Text Index

interMedia Text lets you create a text index (an inverted index) on documents stored in the database. Updating an inverted index requires heavy processing, so changes to a text column are queued and processed in batch. The process of updating the inverted index based on the queue is referred to as "synchronizing" the index.

The second aspect of maintaining your *interMedia* Text index is optimizing. As your index is synchronized, it grows in such a way as to consume more disk space than necessary and reduces the efficiency of queries.

Optimizing your index works differently depending on the mode you select. Optimizing in FAST MODE works on the entire index and compacts fragmented rows, but does not remove old data. FULL MODE permits optimization of the whole index or a portion of the index and both compacts fragmented rows and removes old data. For more information, see the *Oracle8i interMedia Text Reference* for the ALTER INDEX command.

interMedia Text gives you full control over how often each text index is synchronized. You can choose to synchronize every five seconds, for example, if it is important for your application to reflect text changes quickly in the index. You can choose to synchronize once a day, for more efficient use of computing resources and a more optimal index.

After creating your *interMedia* Text index, you need to consider a strategy for maintaining the index. For example, if you have many inserts, updates, or deletes

throughout the day, consider synchronizing the *interMedia* Text index on a daily basis.

6.5.4.1 Synchronize the *interMedia* Text Index

The following example assumes that you installed `ctx_schedule`.

Note: `ctx_schedule` is provided as an example, and is not supported as a component of either *interMedia* Text or Oracle9iAS Portal. For more information, see note 132689.1 on Oracle MetaLink at: <http://metalink.oracle.com>.

1. Log on to SQL*Plus with the appropriate user name and password.
2. Enter the following commands:

```
exec ctx_schedule.startup ( 'ex1_index', 'SYNC', 1 ) ;  
exec ctx_schedule.startup ( 'ex1_index', 'OPTIMIZE FAST', 120 ) ;
```

In this example, the index `ex1_index` is synchronized every minute, and optimized every two hours. This is true even if the database is shut down and restarted.

6.5.4.2 Stop Index Maintenance

The following example assumes that you installed `ctx_schedule`.

Note: `ctx_schedule` is provided as an example, and is not supported as a component of either *interMedia* Text or Oracle9iAS Portal. For more information, see note 132689.1 on Oracle MetaLink at: <http://metalink.oracle.com>.

1. Log on to SQL*Plus with the appropriate user name and password.
2. Enter the following commands:

```
exec ctx_schedule.stop ( 'ex1_index' ) ;  
exec ctx_schedule.stop ( 'ex1_index', 'OPTIMIZE FAST' ) ;
```

`ctx_schedule.stop` assumes that the operation to be stopped is "SYNC", unless you specify otherwise.

Note: To have new text searched immediately (every five seconds), consider using the `drbgdml.sql` script located in:

```
<ORACLE_HOME>/ctx/sample/script/drbgdml.sql
```

6.6 Setting up *interMedia* Text Indexes

In troubleshooting, you may need to reinstall *interMedia* Text, or you may need to recreate the `ctxsys` schema. In both of these cases, you need to run the following script in SQL*Plus to reset the Oracle Portal *interMedia* Text environment:

```
inctxgrn.sql
```

This file is located in the `<ORACLE_HOME>/portal30/src/wws` directory.

Log on using the user name and password for the schema that owns the Oracle Portal content area. For example, if the schema name is "SCOTT", log on with the user name "SCOTT" and the appropriate password.

6.7 Dropping an *interMedia* Text Index

Dropping an index is a very time-consuming and resource-intensive operation so plan this task during non-business hours.

You would drop an *interMedia* Text index in the following situations:

- You want to disable *interMedia* Text and switch back to a basic and advanced search only.
- You know that a significant amount of new content has been added to your content area. First, drop your index, then, recreate your index.

You can drop *interMedia* Text indexes in the following ways:

1. In the **Services** portlet, click **Search Settings**. By default, the Services portlet is located on the Oracle Portal home page's **Administer** tab.
2. In the *interMedia* Text **Properties** section, click **Drop Index**.

The *interMedia* Text index is dropped from the server.

Note:

- You can also drop an *interMedia* Text Index, by running the following script in Oracle SQL*Plus and logging in as the Oracle Portal schema owner:

```
ctxdrind.sql
```

6.8 Setting up Your Environment for *interMedia* Text

For Oracle8i Release 8.1.6.2 on UNIX, configure the `listener.ora` and `tnsnames.ora` files in your Oracle8i Home directory to support these external procedures. On Windows NT/2000, you require an Oracle 8.1.7 database if you want to use *interMedia* Text with Oracle Portal.

Note: If you are running Oracle8i Release 8.1.7, ignore this section as the database does not use external procedures to perform document filtering.

The `listener.ora` and `tnsnames.ora` files on your computer should look similar to the following examples:

6.8.1 listener.ora

The following lines define a complete listener definition in your `listener.ora` file.

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC0))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = oracle) (PORT=1521))
      )
    )
  )
```

6.8.1.1 System ID

If you want to add a system identifier (SID) name of PLSExtProc and a program name of EXTPROC in the server's `listener.ora` file, insert the following in the `SID_LIST_LISTENER` definition:

```

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC = ...

    (SID_DESC =
      (SID_NAME=PLSExtProc) (ORACLE_HOME=/oracle/db/dev118)
      (PROGRAM=extproc)

```

Notes:

- EXTPROC0 and PRESENTATION=RO ends with the letter "o" not the number zero.
 - SID=PLSExtProc is non-negotiable.
-
-

6.8.2 tnsnames.ora

Add the following lines to the end of your `tnsnames.ora` file, to add a net service name description entry for `EXTPROC0` in the server's `tnsnames.ora` file, using `SID` rather than `SERVICE_NAME` in the `CONNECT_DATA` section. For example:

```

extproc_connection_data =
  (DESCRIPTION=
    (ADDRESS_LIST =
      (ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC0))
    )
    (CONNECT_DATA=
      (SID=PLSExtProc)
      (PRESENTATION = RO)
    )
  )

```

Notes:

- Do not substitute your SID anywhere.
 - Do not change the case of any text.
 - EXTPROCO ends with the letter "o" not the number zero.
 - Start the Net8 Assistant to help you configure the `tnsnames.ora`.
 - *See also Net8 Administrator's Guide* for more information about creating these configuration files.
-
-

6.9 Multilingual Functionality (Multilexer)

Note: You may need to increase your tablespace to at least 20 MB to support multilexer.

Oracle8i (8.1.6 and above) provides multilexer which allows you to use language-specific features on documents of different languages stored in the same table. Multilexer is a feature of the index and is configured during index creation. Multilexer requires an extra column in your table, which identifies the language of each document.

At query time, the multilexer chooses a language-specific lexer to lex the query tokens. This is based on the `NLS_LANG` setting for the query session. Thus, a query session in the FRENCH language uses the lexer for FRENCH.

During installation of Oracle Portal, the `sbrimtlx.sql` script creates the language-specific lexer preferences and gathers them under a single multilexer preference.

6.10 *interMedia* Text-related Procedures Created in Oracle Portal

The Oracle Portal installation creates the following procedures in the `ctxsys` schema. These procedures are created to support the user datastores that are used in Oracle Portal content areas for *interMedia* Text indexing.

- `WWSBR_CORNER_CTX_<n>`
- `WWSBR_DOC_CTX_<n>`
- `WWSBR_PERSP_CTX_<n>`

- WWSBR_THING_CTX_<n>
- WWSBR_TOPIC_CTX_<n>

where <n> is the `user_id` of the Oracle Portal schema which may be different for each database. This value is the `user_id` column value from `all_users`.

Building Oracle Portal Reports and Forms with Rich Content

Oracle Portal lets you integrate *interMedia* rich content with Oracle Portal reports and Oracle Portal forms. This is possible via the Oracle Portal framework which leverages *interMedia*'s rich content services.

interMedia rich content includes objects such as graphics, audio, images, and video. These objects are stored in the Oracle8i database, thus eliminating the possibility of lost objects, as is the case when using the file system as the content repository.

Furthermore, storing multimedia content anywhere on the database in any schema is possible. Storage is not limited to a single schema and existing content does not have to be moved; it can remain in a separate schema.

This chapter contains the following sections:

- [interMedia Object Types](#)
- [Rich Content in Oracle Portal Reports](#)
- [Rich Content in Oracle Portal Forms](#)
- [Browsing Tables with the Oracle Portal Navigator](#)
- [Known Issues](#)

7.1 *interMedia* Object Types

The following *interMedia*-specific object types, defined in the ORDSYS schema, are contained in database tables and can be displayed in your Oracle Portal reports and forms:

Table 7–1 *interMedia object types*

Object Type	Description
ORDAUDIO	Supports the storage and management of audio data such as MP3, AU, WAV, MPEG.
ORDIMAGE	Supports the storage, management, and manipulation of image data such as GIF, JPEG, BMP.
ORDVIDEO	Supports the storage and management of video data such as REAL, QuickTime 3/4, AVI, MPEG.

Currently, *interMedia* rich content support includes *interMedia* images, audio, and video. It does not apply to *interMedia* Locator or *interMedia* Text.

For more information about these object types, refer to the *Oracle8i interMedia Audio, Image, and Video User's Guide and Reference* provided with your Oracle database documentation set.

7.2 Rich Content in Oracle Portal Reports

You can build reports with *interMedia* rich content by using either of the following wizards:

Table 7–2 *Oracle Portal Report wizards*

Report	Description
Reports From Query Wizard	This wizard guides you through the steps for creating a report, including creating the SQL query that selects the data displayed in the report. If you are unfamiliar with SQL or are new to Oracle Portal, you may want to create your report using this wizard.
Query By Example Report (QBE)	This wizard lets you create a report for querying, inserting, updating, and deleting data from the table or view on which the report is based. In the QBE report build wizard, you choose which data to display in the report. Or, allow end users to make their own queries in the QBE report's customization form.

When creating an Oracle Portal report with the Report Wizard, choose from the following three layouts: Tabular (default), Form, and Custom.

Important:

Before building a Oracle Portal report or form on *interMedia*-based tables, make sure that you have a table with columns of type ORDIMAGE, ORDAUDIO, or ORDVIDEO that you can use to upload your rich content. This is done outside of Oracle Portal. Consult your DBA if necessary to create such a table in your database.

For *interMedia* Text technical papers, training materials, code samples, and so on, visit:

<http://technet.oracle.com/products/intermedia>

7.2.1 Object Attribute Display

In addition to having these *interMedia* object type columns displayed in the Oracle Portal report or form, object attributes can be displayed in join conditions, formatting, column conditions, and so on. For example, display a video clip's size or duration in an Oracle Portal report and set a condition that only objects modified after a certain date display in the report.

7.2.2 Display Options

These audio, image, and video object types can be rendered in the Oracle Portal report in the following ways:

- Embedded in the report (inline).
- As icons with links pointing to the content (default). When the user clicks the link, the content is displayed on a new Web page or handled by the associated source application; for example, RealPlayer, in a separate window.

Notes:

- These display options apply to all columns containing *interMedia* content in the Oracle Portal report and can be selected by clicking the "Display Options" tab in the Oracle Portal report wizard.
 - Although *interMedia* supports a variety of content types and formats, the browser you are using must natively support the MIME type or have a plugin installed to display rich content that is typically not supported on the Web. For example, most browsers can natively display GIF and JPEG images, but TIFF images are *not* displayed without an installed plugin.
-
-

7.2.3 Building the Report

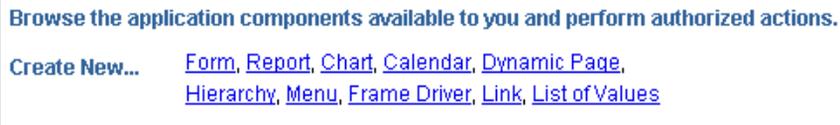
Follow these steps to build an Oracle Portal report that contains rich content:

1. In the Oracle Portal Navigator, click the **Applications** tab.
2. In the **Name** column, the names of all applications in which you have privileges are listed.

Click the name of the application for which you want to build the new component.

3. Beside **Create New**, click the component type you want to build. In this case, click the **Report** link.

Figure 7–1 Application component links on the Navigator



4. Choose the type of report you want to create from these choices: Query By Example (QBE) Reports or Reports From Query Wizard.
5. In the report wizard, enter the appropriate information requested on each page. Click **Next** to proceed to the next wizard page.

6. When building your report with the Oracle Portal report wizards, by default, the *interMedia* rich content is represented by icons that the user can click to view the actual content. However, if you want to embed the content in the Oracle Report page, select the "Embed *interMedia* Rich Content In The Report" check box.

This option is located below the **Display Options** section in the Report Wizard, Step 6 of 9.

Figure 7–2 Oracle Portal Reports Wizard - common options

Common Options

Choose common options that affect the appearance of the QBE

Show Total Row Count Show NULL Values as

Enable update link in the report output

Embed *interMedia* rich content in the report

7. Click **Finish** when you are done.

Figure 7–3 Oracle Portal Reports example with *interMedia* rich content

Audio	Video	Id	Name	Photo	Photo.Height
		22	Bartman		71
		1	test		122

When the user clicks the icon representing the audio, video, or photo object, the actual rich content is displayed.

8. If you want to make any Oracle Portal report, including reports containing *interMedia* rich content, available to Oracle Portal so that other application

providers can use it, make sure that you select the **Publish To Portal** check box which appears on the report wizard's **Access** tab. This tab is displayed after you have entered all the required information in the report wizard.

Figure 7-4 Portal Access tab - Publish to Portal

Portal Access

Check the following option to make the portlet version of this component available to the portal.

Publish to Portal

Note: If you do not see the Publish to Portal option, make sure that the application is exposed as a provider which is set from the application's Access tab. For more information, see the Oracle Portal Online Help topic: *Making an application a portlet provider*.

7.2.4 QBE Reports - Parameter Entry Form

Different users can specialize the same QBE report after it is created by using the report's parameter entry form. However, with reports on tables that contain *interMedia*-based columns, performing insert, update, or delete operations on the table from QBE reports is not possible. Also, object type attributes won't display in the report. Attempting to provide a value for an *interMedia*-based column in the parameter form results in an error. In short, you cannot specialize the report using the object type columns.

7.3 Rich Content in Oracle Portal Forms

With Oracle Portal, uploading images, audio and video clips from your desktop directly into any table in your database via an Oracle Portal form is possible.

The advantage of uploading rich content into *interMedia*-based columns over uploading content into BLOB columns is that the data is automatically parsed to extract several attributes such as MIME-type, length, and any user-defined meta data that might be included in the original media file.

For example, a QuickTime file might contain close captioning or a RealVideo file might have copyright information that can be automatically extracted and stored in the *interMedia* object for indexing and querying.

Creating a form on a table with *interMedia*-based columns is the same as creating a form on any other table. Content that is uploaded into tables in the database can be easily downloaded or made available in the portal framework by building a form on the above tables.

interMedia also provides plugins to allow streaming servers to access content from the database and deliver it to thin clients, and several thick and thin utilities to make uploading and downloading of rich content easy.

7.3.1 Building the Form

Follow these steps to build an Oracle Portal form that contains rich content:

1. In the Oracle Portal Navigator, click the **Applications** tab.
2. In the **Name** column, the names of all applications in which you have privileges are listed.
Click the name of the application for which you want to build the new component.
3. Beside **Create New**, click the component type you want to build. In this case, click the **Form** link.
4. Choose the type of report you want to create from these choices: Form based on table or view or Master-detail form.
5. In the form wizard, enter the appropriate information requested on each page.
Click **Next** to proceed to the next wizard page.
6. On Step 4 of 7, click the name of the *interMedia*-based column from the left frame.
7. In the right frame for the **Item Type** list, choose the File Upload (*interMedia*) option.

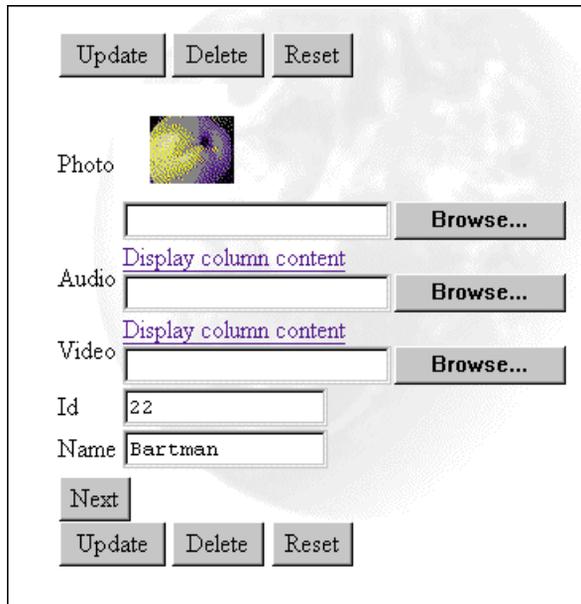
Figure 7-5 Oracle Portal Form - Item Level option



Item Level Options for EMPNO
Item Type File Upload (interMedia) ▼

8. Click **Finish** when you are done.

Figure 7-6 Oracle Portal Form example with interMedia rich content



Update Delete Reset

Photo 

Browse...

[Display column content](#)

Audio Browse...

[Display column content](#)

Video Browse...

Id

Name

Next

Update Delete Reset

9. If you want to make any Oracle Portal form, including forms containing *interMedia* rich content, available to Oracle Portal so that other application providers can use it, make sure that you select the **Publish To Portal** check box which appears on the form wizard's **Access** tab. This tab is displayed after you have entered all the required information in the form wizard.

Note: If you do not see the Publish to Portal option, make sure that the application is exposed as a provider which is set from the application's Access tab. For more information, see the Oracle Portal Online Help topic: *Making an application a portlet provider*.

See also: For more details on *interMedia*, and for useful scripts and tips on using Oracle Portal with *interMedia*, see the Oracle Technology Network at the following URL:
<http://technet.oracle.com/products/intermedia/>

7.4 Browsing Tables with the Oracle Portal Navigator

You can use the Oracle Portal Navigator to browse tables that contain *interMedia* content. However, you cannot upload rich content from the Navigator.

The following is a list of issues with using the Navigator to access Oracle Portal reports and forms containing rich content:

- For columns containing *interMedia* rich content, the Navigator displays icons with links; it does not embed the content on the same page.
- You cannot perform insert, update, or delete operations from the Navigator for reports or forms that contain *interMedia*-based columns nor will object type attributes display.
- You can download or view images from object columns.
- Do not enter text in the text fields for *interMedia*-based columns.
- Since updates are not supported on the Navigator and QBE forms, use Oracle Portal forms instead.

See: The Oracle Portal Online Help topics: *Using the Navigator* and *Accessing the Navigator*.

7.5 Known Issues

The following lists the known issues for support of rich content in *interMedia*-based tables in Oracle Portal reports and forms:

- You may not be able to display *interMedia* rich content in Oracle Portal reports on certain views. This occurs for reports on multi-table views that do not have an implicit ROWID column. This note applies only to views on tables where at least one of the tables contains *interMedia*-based columns and is not a general restriction of Oracle Portal reports. The following error message displays in this case:

ORA-01445 cannot select ROWID from a join view without a key-preserved table.

Cause:

A SELECT statement tried to select ROWIDs from a view derived from a join operation. Because the rows selected in the view do not correspond to underlying physical records, no ROWIDs can be returned.

Workaround:

You can try to access media content directly from the underlying tables using the Reports Wizard and joining relevant tables.

- This integration does not apply to text columns (usually VARCHAR, CLOB, BLOB, or spatial data).

See also: For more information about *interMedia* Text and *interMedia* Locator, see the *interMedia* documentation set provided with Oracle8i.

Troubleshooting

This chapter lists possible solutions to errors that you may encounter while installing or using Oracle Portal.

Specific topics covered include:

- [Verify System Requirements](#)
- [Identify the Component Causing the Problem](#)
- [Troubleshooting Connection Problems with the Diagnostics Tool](#)
- [Configuration Control Points and File Locations](#)
- [Troubleshooting Oracle Portal](#)
 - [Installation and Configuration Problems](#)
 - [Problems Logging on to Oracle Portal](#)
 - [Problems Running Oracle Portal](#)
 - [Miscellaneous Issues Using Oracle Portal](#)

8.1 Verify System Requirements

If you are having any problems installing Oracle Portal, make sure that your system meets the system requirements in [Chapter 1, "Verifying Requirements"](#).

8.1.1 Check Installation Log

Always check the installation session log that describes the actions performed and the components created upon installation.

The log file is located in:

<ORACLE_HOME>/assistants/opca/install.log

Note: The log file includes results from the diagnostic tool.

8.2 Identify the Component Causing the Problem

To troubleshoot any issue, first identify which component of Oracle Portal may be causing the problem. The following is a quick checklist for identifying the component where the problem may likely be occurring:

- Try to access `http://host.domain:port` in your Web browser. Failure to access indicates an issue with the Oracle HTTP Server *powered by Apache* listener. For more information, see [Section A.1.1, "Oracle HTTP Server Configuration File \(httpd.conf\)"](#).
- Confirm that the Oracle HTTP Server is started. Check the listener log file for more details. Specifically, look at the `httpd_error.log` file. Note that externally, the server is addressed with the default port 80; however, internally, the `server.company.com` is listening on port 7777. For more information, see *Using the PL/SQL Gateway* which is provided as part of the Oracle9i Application Server documentation set.
- Try to access `http://host.domain:port/pls/admin_/` in your Web browser. Failure to access indicates an issue with the PL/SQL Gateway (Apache `mod_plsql`) or its configuration. Check the DAD settings in `mod_plsql` and verify the username, password, and connect strings for both the Oracle Portal DAD and the Login Server (SSO) DAD. See [Section A.1.3, "Database Access Descriptor \(DAD\) Configuration File \(wdbsvr.app\)"](#).
- Access `http://host.domain:port/servlet/IsItWorking` in your Web browser. Failure to access indicates an issue with the Apache JServ. Most internal server errors are related to Apache JServ's failure to start due to a port conflict. Check the Apache JServ log files for more details. The log files are located in `<ORACLE_HOME>/Apache/Jserv/logs`.

8.2.1 Location of Apache Log Files

The Apache log files are located in the following directories:

Table 8–1 Apache log file locations

Apache listener log file	<code><ORACLE_HOME>/Apache/Listener/logs</code>
--------------------------	---

Table 8–1 Apache log file locations

Apache JServ log file	<ORACLE_HOME>/Apache/Jserv/logs
-----------------------	---------------------------------

where <ORACLE_HOME> is the location of your Oracle9i Application Server.

Try accessing Oracle Portal as described in [Section 2.4, "Accessing Oracle Portal in Your Browser"](#). If you still cannot connect to Oracle Portal and have just completed the installation, use the diagnostics tool or see [Section 8.5, "Troubleshooting Oracle Portal"](#).

8.3 Troubleshooting Connection Problems with the Diagnostics Tool

The diagnostics tool locates any Oracle Portal configuration problems involving a single installation. Thus, this tool is *not* useful in a distributed Oracle Portal environment with multiple nodes.

Many of the portal connection problems occur because the `ssodatan` script did not populate the configuration information in the Login Server when run.

To use this tool, verify that the following conditions exist:

- Oracle Portal was installed successfully and basic network connections such as TCP/IP and SQL*Net are properly configured. Perform a test to connect to the Oracle Portal schema from SQL*Plus. Fix any connection failures before running the diagnostics tool.
- The Oracle Portal schema and the Login Server schema exist in the same database instance or machine.
- Oracle Portal uses the HTTP protocol.
- The owner of the Oracle Portal schema is able to connect to the location of the Oracle Portal installation: either `webdb30/admin/plsql` (Oracle Portal version 3.0.6 and below) or `portal30/admin/plsql` (version 3.0.6 and above) to run this tool. The owner must also have file creation privileges in the directory containing the diagnostics tool.

8.3.1 Problems Detected by the Diagnostics Tool

Many Oracle Portal connection problems arise due to a misconfiguration in the Login Server. During an Oracle Portal installation, the `ssodatan` script is responsible for associating the Oracle Portal installation node with the appropriate Login Server and populating the enabler tables. However, if this script fails, the

diagnostics tool can read the configuration information in both the Oracle Portal schema and the Login Server schema. A diagnostic report provides any missing information in these tables. In addition, this tool reads the configuration information for the partner applications from the Login Server.

For the URLs stored in both the partner enabler configuration information and the partner application configuration information, it performs the following checks:

- Verifies that the correct protocol exists. Currently, the diagnostics tool works only for HTTP.
- Verifies the host name and port number by establishing a connection with Oracle Portal.
- Verifies that the PL/SQL Gateway (modplsql) is running.
- Verifies that JServ is working.
- Reads the Database Access Descriptor (DAD) information and retrieves the schema name, connect string, and the authentication mode used by this Oracle Portal installation. Any problems reading the DAD information are reported.
- Reads the DAD name from the preference store and compares it with the DAD name obtained from the Oracle Portal URL and reports any differences.

8.3.2 Problems Not Detected by the Diagnostics Tool

The following is a list of problems that the diagnostics tool does not detect:

- Installation problems including those encountered while loading various database objects such as tables, indices, packages, Java classes, and so on.
- Database problems.
- Oracle HTTP Server powered by Apache server configuration problems.
- Problems in an installation where Oracle Portal and the Login Server are installed in different database instances.
- Problems in an Oracle Portal installation that uses HTTPS.

8.3.3 Running the Diagnostics Tool

In Oracle Portal 3.0.8.5.3 and above, when the diagnostics tool is running, the Java class, `Diagnose.class`, and the PL/SQL package, `wwsec_diagnostics`, are loaded into the database.

If you are running an earlier Oracle Portal version, download the Java archive file, `diag.jar`, from the Oracle Technology Network to your Oracle Portal installation location (`portal30/admin/plsql`) and run the following command to extract the files (you can also use WinZip):

```
jar -xvf diag.jar
```

The `wvc` directory is created if it does not already exist.

Note: If you are a UNIX user, execute the `chmod +x diag.cmd` before you run the diagnostics tool.

You run this tool from the command line:

UNIX: `diag.csh`

Windows NT/2000: `diag.cmd`

Example

```
diag.csh -s portal30_schema -p portal30_schema_password -c connect_string
```

where:

Table 8–2 *Diagnostics tool's diag parameters*

Parameter	Description
<code>portal30_schema</code>	Is the name of the database schema that contains the Oracle Portal product. The default schema name is <code>portal30</code> .
<code>portal30_schema_password</code>	Is the password for to the Oracle Portal schema.
<code>connect_string</code>	Is the connect string for the database in which Oracle Portal is installed. The default value is <code>orcl</code> .

The diagnostics tool also provides any recommendations to the user based on these tests.

8.3.4 Sample Diagnostics Report

The diagnostics report, `diag.txt`, is created when the diagnostics command is run. Below is a sample report output.

Diagnostics Report v 1.0: Oracle Portal v 3.0.8.6.5

As of 14-Dec-2000 15:33:01 Schema Name: SML SSO Schema Name: sml_SSO

SML.wwsec_enabler_config_info\$

Login Server URL : http://host.domain.com:3000/pls/sml_sso/sml_SSO.wwssso_app_admin.ls_login

DAD : sml_sso

Host connection : successful.

mod_plsql : working.

JServ : working.

Schema name : sml_sso

Connect string : orcl

Authentication mode : Single Sign-On

sml_sso.wwsec_enabler_config_info\$

Login Server URL : http://host.domain.com:3000/pls/sml_sso/sml_SSO.wwssso_app_admin.ls_login

DAD : sml_sso

Host connection : successful.

mod_plsql : working.

JServ : working.

Schema name : sml_sso

Connect string : orcl

Authentication mode : Single Sign-On

Partner Application Information

**** Oracle Portal (sml) ****

Home URL : http://host.domain.com:3000/pls/sml/sml.home

Success URL : http://host.domain.com:3000/pls/sml/sml.wwsec_app_priv.process_signon

DAD : sml

Host connection : successful.

mod_plsql : working.

JServ : working.

Schema name : sml

Connect string : orcl

Authentication mode : Single Sign-On

**** The Login Server (sml_SSO) ****

Home URL : http://host.domain.com:3000/pls/sml_sso/sml_SSO.home

Success URL : http://host.domain.com:3000/pls/sml_sso/sml_SSO.wwssso_home.process_signon

DAD : sml_sso

```

Host connection : successful.
mod_plsql : working.
JServ : working.
Schema name : sml_sso
Connect string : orcl
Authentication mode : Single Sign-On

*****

Diagnostics completed successfully!

```

8.4 Configuration Control Points and File Locations

When you are planning an installation or troubleshooting an Oracle Portal configuration problem, be aware of the various configuration control points which are discussed in [Appendix A, "Oracle9i Application Server Configuration Files"](#). For your convenience, the following table is provided below:

Configuration File/table	Location or Description
Oracle HTTP Server	<ORACLE_HOME>/Apache/Apache/conf/httpd.conf
Apache JServ	<ORACLE_HOME>/Apache/Jserv/etc/zone.properties
PL/SQL Gateway	<ORACLE_HOME>/Apache/modplsql/cfg/wdbsvr.app
Database Connection	<ORACLE_HOME>/network/admin/tnsnames.ora
Login Server enabler table	Oracle Portal and Login Server's configuration table, WWSEC_ENABLER_CONFIG_INFO\$.
Login server configuration table	Login Server's Partner Applications configuration table, WWSSO_PAPP_CONFIGURATION_INFO\$.
Local HOSTS file	This file resolves IP names to IP addresses. On Windows NT, this file is at \Winnt\system32\drivers\etc\hosts. On Unix, it is at /etc/hosts.
winstall, linstall, ssodatan, ssodatx scripts	<ORACLE_HOME>/portal30/admin/plsql/ For more information, see Appendix B, "Oracle Portal Installation and Configuration Scripts" .

where <ORACLE_HOME> is the location of your Oracle9i Application Server installation.

8.4.1 Oracle Portal Installation Directory Name Change

Depending on your Oracle Portal version, the default location of your installation files is as follows:

Table 8–3 Oracle Portal installation directory structure

Oracle Portal 3.0.6	<ORACLE_HOME>/webdb30
Oracle Portal 3.0.7 and above	<ORACLE_HOME>/portal30

8.5 Troubleshooting Oracle Portal

The following lists configuration errors and problems.

Installation and Configuration Problems

Table 8–4 List of installation and configuration problems

Problem or Error
1. Problem: The Oracle Portal Configuration Assistant displays one or more of the following errors, prompting you for the SYS password and database connect information.
2. Error: The allocated SHARED_POOL_SIZE parameter for the database is insufficient for the Oracle Portal installation.
3. Error: The allocated JAVA_POOL_SIZE parameter for the database is insufficient for the Oracle Portal installation.
4. Error: The default tablespace selected requires 150 MB of free space for the Oracle Portal installation. Increase the tablespace size to proceed with the installation.
5. Problem: The Oracle Portal Configuration Assistant continues to display one or more of the errors listed above after the init.ora settings have been corrected.
6. Problem: The Temporary Tablespace drop-down list for the Oracle Portal schema or Login Server schema is disabled in the Oracle Portal Configuration Assistant.
7. Problem: Oracle Portal Configuration Assistant displays one or more of the following errors after completing the configuration of Oracle Portal.
8. Error: The enabler configuration table, WWSEC_ENABLER_CONFIG_INFOS, does not have any entries.
9. Error: The Oracle Portal schema user was not created.
10. Error: The Login Server user was not created.
11. Error: There are invalid packages in the Oracle Portal schema.

Problems Logging on to Oracle Portal

Table 8–5 *List of problems logging on to Oracle Portal*

Problem or Error
1. Error: There are invalid packages in the Login Server schema.
2. Error: Database Login Failure” while trying to connect to Oracle Portal.
3. Error: Preference path not found error
4. Problem: Receive the error: "You cannot login because there is no configuration information stored in the enabler configuration table (WWC-41439)" when trying to log on to Oracle Portal.
5. Problem: Receive the error "Proxy log on failed" together with the message "TNS could not resolve service name" when trying to connect or log into Oracle Portal.
6. Problem: Received the error “HTTP 400 - Bad Request /Malformed Host Header.”
7. Problem: Cannot log on to Oracle Portal due to an incorrect Proxy Setting.

Problems Running Oracle Portal

Table 8–6 *List of problems running Oracle Portal*

Problem or Error
1. Problem: The database and/or TNS listener crashes when running Oracle Portal on Windows NT/2000.
2. Problem: Apache Listener crashes frequently.
3. Problem: Receive the error 'Call to utl_http failed' when clicking on a URL item link rendered "in - place."
4. Problem: Occasionally receive the error "Timeout for content={0}" in one or more portlets.
5. Problem: Receive the error "The listener returned the following message: 503 Service Temporarily Unavailable" intermittently when running Oracle Portal.
6. Problem: Receive the error “Internal Server Error” consistently when trying to access any page in Oracle Portal.
7. Problem: Receive the error “Internal Server Error” intermittently when trying to access Oracle Portal.
8. Problem: Receive “400 bad request” error or the Web browser hangs when trying to access Oracle Portal.
9. Error: The request for content either timed out, or produced an error, after 0 seconds.

Miscellaneous Issues Using Oracle Portal

Table 8-7 List of miscellaneous issues using Oracle Portal

Problem or Error	
1.	Problem: Unable to create interMedia Text indexes.
2.	Problem: Apache generates the following error on startup: "The procedure entry point snlpcgtsrvbynm could not be located in the dynamic link library oranl8.dll."
3.	Error: missing string (login link text) language(e) domain(wwc) sub_domain(sec) Missing string(pages) language(e) domain(wwc) sub_domain(pob).
4.	Error: PLS-00306: wrong number or types of arguments.

8.5.1 Installation and Configuration Problems

Problem: The Oracle Portal Configuration Assistant displays one or more of the following errors, prompting you for the SYS password and database connect information.

Error	The Java option is not enabled in the selected database. The Java option in the database must be enabled to install Oracle Portal.
Cause	Oracle Portal requires that the Oracle8i Java Virtual Machine (JVM) database option be installed and available in the database in which you are installing Oracle Portal. This error appears when the installer cannot find the Oracle8i JVM. The installer checks the existence of the Oracle8i JVM option with the following query: <pre>select count(object_name) from all_objects where object_type like 'JAVA%' and status='VALID'</pre>
Solution	Either install the Oracle8i JVM option by running the Oracle8i Database Configuration Assistant or specify a different Oracle8i database in the Oracle Portal installation.

Error: The allocated SHARED_POOL_SIZE parameter for the database is insufficient for the Oracle Portal installation.

Cause	<p>Oracle Portal requires that the SHARED_POOL_SIZE parameter be greater than 31457280. The installer determined that the SHARED_POOL_SIZE parameter for the database is set to a value less than 31457280. The installer checks for this requirement with the following query:</p> <pre>select value from v\$parameter where name like 'shared_pool_size'</pre>
Solution	<p>Increase the SHARED_POOL_SIZE allocation in the <code>init.ora</code> file for your database to continue the installation process. Shutdown and restart the database for your changes to take effect.</p>

Error: The allocated JAVA_POOL_SIZE parameter for the database is insufficient for the Oracle Portal installation.

Cause	<p>Oracle Portal requires that the JAVA_POOL_SIZE parameter be greater than 20971520. The installer determined that the JAVA_POOL_SIZE parameter for the database is set to a value less than 20971520. The installer checks for this requirement with the following query:</p> <pre>select value from v\$parameter where name like 'java_pool_size'</pre>
Solution	<p>Increase the JAVA_POOL_SIZE allocation in the <code>init.ora</code> file for your database to continue the installation process. The JAVA_POOL_SIZE setting must be increased to a value greater than 20971520. Shutdown and restart the database for your changes to take effect.</p>

Error: The default tablespace selected requires 150 MB of free space for the Oracle Portal installation. Increase the tablespace size to proceed with the installation.

Cause	<p>Oracle Portal requires at least 150 MB of free space in the DEFAULT tablespace that is specified for the Oracle Portal schema. The Configuration Assistant detected that the DEFAULT tablespace you selected has less than 150 MB of free space available. The installer checks for this requirement with the following query:</p> <pre>select sum(bytes)/1024 from DBA_FREE_SPACE where tablespace_name like '" + <UserSelectedTablespace> + "'</pre>
Solution	<p>Specify a different DEFAULT tablespace for the Oracle Portal schema or increase the amount of free space available in the tablespace you have selected.</p> <p>The <code>alter database datafile</code> command achieves this. You should set the <code>autoextend</code> option when altering the tablespace size.</p> <p>See your Oracle database documentation for details.</p>
Tip	<p>On Windows NT/2000, for a default installation of the Oracle8i database and an Oracle9i Application Server, you can resize your tablespace for a faster installation of Oracle Portal in the following way:</p> <ol style="list-style-type: none"> 1) Install a default Oracle8i database. <ol style="list-style-type: none"> a) Measure its tablespace size. 2) Install a default installation of Oracle9i Application Server (HTTP only install option). <ol style="list-style-type: none"> a) Measure its tablespace size. 3) Subtract 1a from 2a. <p>Note: Use only the datafile changes; do not use the tablespace and/or index changes.</p> <pre>alter database datafile 'C:\oracle\oradata\orcl\system01.dbf' resize 510M; alter database datafile 'C:\oracle\oradata\orcl\users01.dbf' resize 140M;</pre>

Problem: The Oracle Portal Configuration Assistant continues to display one or more of the errors listed above after the `init.ora` settings have been corrected.

Cause	Based on the queries that the Configuration Assistant uses, the settings still appear to be incorrect.
Solution	Verify that you have entered the new values in the <code>init.ora</code> file as a valid number of bytes without using any abbreviated notations (for example, 60M as an abbreviation for 60000000). Since the Configuration Assistant compares the settings as numbers, all the digits must be entered without using abbreviated notation. Also, you must shutdown and restart your database anytime changes to the <code>init.ora</code> settings are made.

Problem: The Temporary Tablespace drop-down list for the Oracle Portal schema or Login Server schema is disabled in the Oracle Portal Configuration Assistant.

Cause	In version 3.0.6 of Oracle Portal, the Configuration Assistant populates the Temporary Tablespace list with those tablespaces that are of type "TEMPORARY." If your database does not have any of these tablespaces, then the drop-down list is disabled. In versions 3.0.7 and above, this problem has been fixed.
Solution	This problem must be fixed before continuing with the installation. Define at least one tablespace in your database that is of type "TEMPORARY."

Problem: Oracle Portal Configuration Assistant displays one or more of the following errors after completing the configuration of Oracle Portal.

Error	The SSOHash class has not been loaded into the database.
Cause	The Oracle Portal Configuration Assistant could not find this class after the configuration was complete. To enable Single Sign-On, Oracle Portal installs the SSOHash Java class during the configuration process. This class is necessary to log on to Oracle Portal.

Solution Check the Oracle Portal installation and configuration log file for other errors. Additional errors in the log file are usually an indication that there is a more fundamental problem with the way the database was configured.

If there are no other errors, the SSOHash class can be manually installed by entering the following command from the <ORACLE_HOME>/portal30/admin/plsql/wwc directory.

```
loadjava -resolve -user <PORTAL_
SCHEMA> /<PORTAL_SCHEMA>@<CONNECT>
SSOHash.class
```

where <PORTAL_SCHEMA> is the database schema name where Oracle Portal is installed and <CONNECT> is the TNS connect string for your database.

Error: The enabler configuration table, WWSEC_ENABLER_CONFIG_INFO\$, does not have any entries.

Cause The Oracle Portal Configuration Assistant did not detect any entries in the WWSEC_ENABLER_CONFIG_INFO\$ table after the configuration was complete. Oracle Portal uses the WWSEC_ENABLER_CONFIG_INFO\$ table when contacting the Login Server. This table must have at least one entry for Oracle Portal to function properly.

Solution Check the Oracle Portal installation and configuration log file for other errors. Additional errors in the log file are usually an indication that there is a more fundamental problem with the way the database is configured.

If there are no other errors, fix this problem by running the ssodatan script. See [Section B.4, "Configuring a New Oracle Portal Instance and Login Server with the ssodatan Script"](#).

Error: The Oracle Portal schema user was not created.

Cause The Oracle Portal Configuration Assistant did not find the Portal schema in the database after the configuration was complete.

Solution	<p>Check the Oracle Portal installation/configuration log file for other errors. Additional errors in the log file are usually an indication that there is a more fundamental problem with the way the database is configured. The log file is located in <ORACLE_HOME>/assistants/opca/install.log. When the Portal schema doesn't get created, the configuration process generates a large number of errors. The reason for the Portal Schema not being created should be close to the top of the log file. Once the problem has been determined and fixed, deinstall Oracle Portal and rerun the Configuration Assistant.</p>
-----------------	--

Error: The Login Server user was not created.

Cause	<p>The Oracle Portal Configuration Assistant did not find the Login Server schema in the database after the configuration was complete.</p>
Solution	<p>Check the Oracle Portal installation and configuration log file for other errors. Additional errors in the log file are usually an indication that there is a more fundamental problem with the way the database is configured.</p> <p>When the Login Server schema is not created, the configuration process generates a large number of errors which typically appear at the top of the log file.</p> <p>Once the problem has been determined and fixed, deinstall Oracle Portal and rerun the Configuration Assistant. See Section 2.7, "Deinstalling Oracle Portal".</p>

Error: There are invalid packages in the Oracle Portal schema.

Cause	<p>One or more errors were generated during the configuration of Oracle Portal causing some of the PL/SQL packages to be invalid. Invalid packages may be caused by some Oracle Portal database objects failing while being created.</p> <p>Invalid packages may also result if required dependencies are not installed in the database. Some of the required dependencies include the PL/SQL Web Toolkit (OWA Packages), as well as the standard PL/SQL packages available in the SYS schema.</p>
--------------	--

Solution	<p>Check the Oracle Portal installation and configuration log file for other errors. Additional errors in the log files are usually an indication that there is a more fundamental problem with the way the database is configured.</p> <p>Verify that you have installed the PL/SQL Web Toolkit that ships with your version of Oracle Portal. Oracle Portal makes heavy use of the PL/SQL Web Toolkit and requires that the latest version be installed.</p>
-----------------	--

Error: There are invalid packages in the Login Server schema.

Cause	<p>One or more errors were generated during the configuration of Oracle Portal causing some of the PL/SQL packages to be invalid. Invalid packages may be caused by some Login Server objects failing while being created.</p> <p>Invalid packages may also result if required dependencies are not installed in the database. Some of the required dependencies include the PL/SQL Web Toolkit (OWA Packages), as well as the standard PL/SQL packages available in the SYS schema.</p>
--------------	--

Solution	<p>Check the Oracle Portal installation and configuration log file for other errors. Additional errors in the log files are usually an indication that there is a more fundamental problem with the way the database is configured.</p> <p>Verify that you have installed the PL/SQL Web Toolkit that ships with your version of Oracle Portal. Because of Oracle Portal's heavy use of the PL/SQL Web Toolkit, it is important that the latest version be installed.</p>
-----------------	---

Error: Database Login Failure” while trying to connect to Oracle Portal.

Cause	<p>The Database Access Descriptor (DAD) for Oracle Portal may be incorrect or the TNS names entry used in the DAD is not defined in the <code>tnsnames.ora</code> file located in the ORACLE_HOME location containing your Oracle9i Application Server files, including Oracle Portal. Additionally, it is possible that the installation and configuration process generated errors.</p>
--------------	---

Solution	<p>Verify the DAD configuration by entering the following URL: <code>http://host.domain:port/pls/admin_/gateway.htm</code></p> <p>For more information, see <i>Using the PL/SQL Gateway</i> which is provided as part of the Oracle9i Application Server documentation set.</p> <p>Make sure that the connect string information for the database is correct and the same when connecting through SQL*Plus.</p> <p>If you have multiple Oracle Homes, confirm that the appropriate connect string is added to <code><ORACLE_HOME>/network/admin/tnsnames.ora</code>. If in doubt, add the connect string to the <code>tnsnames.ora</code> files in all the Oracle Homes.</p> <p>Also, for UNIX, check the following file to verify that the appropriate ORACLE_HOME is being used: <code><ORACLE_HOME>/Apache/Apache/bin/apachectl</code></p> <p>Check the Oracle Portal installation and configuration log file for other errors. Additional errors in the log file are usually an indication that there is a more fundamental problem with the way the database is configured.</p>
-----------------	--

Error: Preference path not found error

Reason	<p>The preference path is part of the seed data. This error indicates an incomplete installation; one or more portal objects were not installed.</p> <p>The allocated <code>SHARED_POOL_SIZE</code> parameter for the database is not sufficient for the Oracle9iAS Portal.</p> <p>Oracle Portal requires that the <code>SHARED_POOL_SIZE</code> parameter be greater than 15728640. The installer determined that the <code>SHARED_POOL_SIZE</code> parameter for the database is set to a value less than 15728640.</p> <p>You can verify the size by issuing the following query:</p> <pre>select value from v\$parameter where name like 'shared_pool_size'</pre>
Solution	<p>From the Oracle8i database home location, edit the <code>init.ora</code>. Increase the <code>SHARED_POOL_SIZE</code> allocation to continue the installation process. Shutdown and restart your database for this value to take effect.</p> <p>Be sure to specify all the digits (for example, 1000000000) for the <code>SHARED_POOL_SIZE</code>; do not use abbreviations (1000M).</p>

8.5.2 Problems Logging on to Oracle Portal

Problem: Receive the error: "You cannot login because there is no configuration information stored in the enabler configuration table (WWC-41439)" when trying to log on to Oracle Portal.

There are several potential reasons for this error message. Each reason is listed below with a corresponding solution.

Reason 1	The hostname specified in the <code>httpd.conf</code> file was written using capital letters. This prevents the SSO subsystem from finding a matching entry in the enabler configuration table.
Solution	Modify the "servername" setting in the <code>httpd.conf</code> file so that the hostname uses all lowercase letters.
Reason 2	The SSOHash class has not been loaded into the database. Check this by running the following query in SQL Plus: <pre>select * from all_objects where object_type is like 'JAVA CLASS'</pre>
Solution	Manually load the SSOHash class using the following command: <pre>loadjava -resolve -user <PORTAL_ SCHEMA>/<PORTAL_SCHEMA>@<CONNECT> SSOHash.class</pre> <p>where <code><PORTAL_SCHEMA></code> is the database schema name containing your Oracle Portal installation and <code><CONNECT></code> is the TNS connect string for your database.</p>
Reason 3	The <code>JAVA_POOL_SIZE</code> parameter for your database is too small. Oracle Portal requires that the <code>JAVA_POOL_SIZE</code> is set to a value greater than 20971520.
Solution	Increase the <code>JAVA_POOL_SIZE</code> allocation to continue the installation process. The <code>JAVA_POOL_SIZE</code> value is set in the <code>init.ora</code> file for your database. Also, shutdown and restart your database anytime changes to the <code>init.ora</code> settings are made.
Reason 4	If you are installing or configuring Oracle Portal manually, then the URL may have been mistyped when running the <code>ssodatan</code> script.
Solution	Rerun the <code>ssodatan</code> script with the correct data. See Section B.5, "Updating an Existing Portal Instance with the ssodatax Script" .

Reason 5	An alias which is defined in the Apache configuration causes Apache to translate <code>host.domain.com</code> to just <code>host</code> . In this case, the login link shows only <code>host:port</code> (dropping the domain).
Solution	Remove all such aliases from your Apache configuration file, <code>http.conf</code> .
Reason 6	The default domain is not set in the Apache configuration. When this occurs, only the hostname is shown in the Login link and the domain is not included.
Solution	Define the default domain in the Apache configuration file, <code>http.conf</code> .
Reason 7	The default HTTP port (port 80) was specified during installation, configuration, or when running the <code>ssodatan</code> script. Unfortunately, Apache, and some browsers, drop the <code>:80</code> .
Solution	Run the <code>ssodatan</code> script without specifying port <code>:80</code> . Also, when accessing Oracle Portal through the browser, do not specify the port. See Section B.4, "Configuring a New Oracle Portal Instance and Login Server with the ssodatan Script" and Section 2.4, "Accessing Oracle Portal in Your Browser" .
Reason 8	Mismatch in the case of the hostname in Oracle Portal and Apache.
Solution	Change the hostname to all lowercase in: <code>Oracle9iAS_Home/Apache/Apache/conf/httpd.conf</code>

Problem: Receive the error "Proxy log on failed" together with the message "TNS could not resolve service name" when trying to connect or log into Oracle Portal.

Cause	This is likely due to a Net8 configuration problem.
Solution	Verify that the <code>ORACLE_HOME</code> where the listener was started is pointing to the same home where the Oracle9i Application Server <code>tnsnames.ora</code> file is located. Verify that the <code>tnsnames.ora</code> file that exists, is valid, and that it contains the service name associated with your DAD. Refer to the Net8 documentation provided with your Oracle8i database documentation library.

Problem: Received the error “HTTP 400 - Bad Request /Malformed Host Header.”

Cause	This situation can happen when the hostname on the machine where Apache is running contains the underscore ‘_’ character. Underscores are invalid in URLs.
Solution	Remove any underscores in the name of the host or access the machine by its IP Address.

Problem: Cannot log on to Oracle Portal due to an incorrect Proxy Setting.

Cause	If the Proxy Setting is incorrect or invalid, Oracle Portal cannot process logins.
Solution	<p>Using SQL*Plus, log on as the Oracle Portal schema owner (default is PORTAL30) and issue the following statements:</p> <pre> begin wwpre_api_value.set_value_as_varchar2(p_path=> 'oracle.portal.proxy', p_name=> 'name', p_level_type =>wwpre_api_value.SYSTEM_LEVEL_ TYPE, p_level_name => null, p_value => NULL); wwpre_api_value.set_value_as_number(p_path=> 'oracle.portal.proxy', p_name=> 'port', p_level_type => wwpre_api_value.SYSTEM_LEVEL_TYPE, p_level_name => null, p_value=> NULL); end; / </pre> <p>After issuing these statements, the value of the proxy setting is NULL.</p>

8.5.3 Problems Running Oracle Portal

Problem: The database and/or TNS listener crashes when running Oracle Portal on Windows NT/2000.

Cause	This is most likely happening because you are running a version of the UTL_HTTP package that shipped with early versions of the Oracle 8.1.6 database on Windows NT/2000.
Solution	<p>Download and install the Oracle8i database 8.1.6.2 patch from Metalink at:</p> <p>http://metalink.oracle.com</p> <p>Click patches, and then select "product: Oracle Server - Enterprise Edition", "platform: MS Windows NT/2000". You must be registered to use metalink.</p>

Problem: Apache Listener crashes frequently.

Cause	The most common problem with Apache stability is rooted in Oracle Portal's use of new 8.1.7 client libraries against an 8.1.6 database. The database team has discovered a protocol problem that exposed itself on threaded client applications like the Windows NT/2000 version of mod_plsql. This is not a problem on UNIX because mod_plsql and Apache are process-based. This patch is relevant if you are running a Windows NT/2000 middle-tier against a database on either Windows NT/2000 or UNIX.
Solution	<p>Download and install the Oracle8i database 8.1.6.2 patch from Metalink at:</p> <p>http://metalink.oracle.com</p> <p>Click patches, and then select "product: Oracle Server - Enterprise Edition", "platform: MS Windows NT". You must be registered to use metalink.</p>

Problem: Receive the error 'Call to utl_http failed' when clicking on a URL item link rendered "in - place."

Cause	An incorrect proxy value may be specified if running within a firewall.
--------------	---

Solution Verify that the proxy is properly set on the Global Settings page in Oracle Portal which is accessed from the Services portlet on the Administers tab page. Click the Administer tab on the Portal Home Page. In the Proxy Server section, provide appropriate values for the HTTP Server, HTTP Server Port, and No Proxy Servers for Domains beginning with fields.

See [Section 6.5.1, "Step 1: Set up the Global Page Settings"](#).

Problem: Occasionally receive the error "Timeout for content={0}" in one or more portlets.

Cause The Parallel Page Engine is timing out before the portlet has a chance to respond. The {0} was actually a bug in early versions of Oracle Portal that has since been fixed. The error in newer versions of Oracle Portal displays the timeout value that was exceeded.

Solution In the `zone.properties` file on your listener, set the following value to change the default timeout period. The value should be set high enough to allow the portlet time to respond.

```
servlet.page.initArgs=requesttime=40
```

The value "40" can be substituted with a higher or lower number, depending on your requirements.

For individual database portlets, you can also set the timeout in the portlet record. For Web portlets, increase the timeout in `provider.xml`.

Problem: Receive the error "The listener returned the following message: 503 Service Temporarily Unavailable" intermittently when running Oracle Portal.

Cause

This may occur when mod_plsql cannot connect to the database because the maximum number of database sessions has been reached. The database connection pool in mod_plsql is not shared across Apache processes meaning each process maintains its own pool. The total number of database connections pooled in Apache mod_plsql is directly related to the number of Apache processes that are spawned off and the number of DADs used to access different PL/SQL applications. The PL/SQL Gateway (mod_plsql) pools one database session per DAD per Apache process. Simply stated, the maximum number of database sessions that is pooled by mod_plsql is calculated as follows:

`NumberOfApacheProcesses*NumberOfDADs`

Currently on Windows NT/2000, since Apache is multi-threaded, all threads share the same database connection pool. The maximum number of database sessions that is pooled by mod_plsql on Windows NT/2000 is calculated as follows:

`MaximumNumberOfApacheThreadsEverActiveForEachDAD`

Ideally, every thread can take advantage of a database session created by another thread. Thus, on platforms where Apache is not multi-threaded, it is important that it be tuned carefully.

Solution The Apache process configuration requires tuning so that processes are not started up or shutdown heavily (each process takes down its connection pool, and a new process needs to replenish its pool). This tuning is governed by the load on the Web server.

The maximum number of database sessions needs to be setup according to the maximum number of Apache processes expected. Edit the following parameters in the `http.conf` file:

- `Set MaxRequests = MaxSpareServers`
- `MaxRequestsPerchild=HighNumber`
- `MinSpareServers=0`
- `KeepAlive off`
- `KeepAliveTimeOut`

This configuration ensures that Apache processes are very rarely shutdown and the overhead of creating an Apache process/new database connection is greatly reduced.

Solution Check the `processes` parameter in your database configuration file (`init$SID.ora => processes=NNN`). This number should be greater than or equal to the maximum number of Apache processes configured in the `httpd.conf` file:

`StartServers+MaxSpareServers`

One way to verify that you have this problem is to connect as SYS through SQL*Plus, and issue the query "select username from v\$session". If the count for the number of rows is almost the same as the value of the `processes` parameter, then you are likely exceeding the maximum number of processes.

Solution Configure a separate Oracle HTTP Server *powered by Apache* listener to handle only PL/SQL requests. The main Apache listener can be used to redirect all PL/SQL requests to the new listener. For the new Oracle9i Application Server listener, specify a low number for the Apache processes parameter since it only handles PL/SQL requests. Thus, the database session numbers are kept to a minimum.

See [Section 5.6, "Tuning the Oracle HTTP Server"](#).

Problem: Receive the error “Internal Server Error” consistently when trying to access any page in Oracle Portal.

Cause	This problem is most likely due to the Parallel Page Engine or the Apache JServ Engine having gone down or been misconfigured.
Solution	<p>Ensure that the Apache JServ process has been started and is configured correctly by trying to access the following URL from any browser:</p> <pre>http://host.domain:port/servlet/IsItWorking</pre> <p>If this does not work, then Apache JServ is failing. Check the Apache JServ log files for more details. The log files are located in <ORACLE_HOME>/Apache/Jserv/logs.</p>

Problem: Receive the error “Internal Server Error” intermittently when trying to access Oracle Portal.

Cause	This problem is most likely due to the Apache JServ crashing periodically and then being restarted by Apache.
Solution	Check the JServ and Apache log files to determine what is causing the JServ process to crash. The log files are located in <ORACLE_HOME>/Apache/Jserv/logs.

Problem: Receive “400 bad request” error or the Web browser hangs when trying to access Oracle Portal.

Cause	If Secure Socket Layer (SSL) has been configured, the most likely reason for this error is because the Parallel Page Engine is trying to talk HTTP over an HTTPS port.
Solution	<p>Add the following configuration line in the <code>zone.properties</code> file to instruct the Parallel Page Engine to use https whenever it talks to this port.</p> <pre>servlet.page.initArgs=httpsports=<HTTPS port></pre>

Error: The request for content either timed out, or produced an error, after 0 seconds.

Cause

This is not a time out of the content being returned from the request. This message occurs when a connection problem occurs and may appear for any of the following reasons:

A connection is refused due to a server being down, overloaded, or a machine not found, and the like.

A connection is closed during communication due to some type of instability with the machine, network, or listener.

A connection takes too long to establish due to a DNS lookup, slow network, slow listener, and the like.

The latter is usually the cause of this error. However, the first reason occurs occasionally with Web providers that do not having their listener up and running.

Solution

There is a parameter that can be set in the JServ configuration file, `zone.properties`, that can resolve the third problem. Edit the following `zone.properties` parameter as required:

```
servlet.page.initArgs=stall=<time in sec>
```

The time in seconds serves as the a stalling mechanism for the connection. The default in the code is 10 sec, which may be insufficient if the DNS is taking too long. Normally, a connection is established almost immediately. However, if more time is required to establish the connection, a higher value can be entered. For example, entering a value of 20 sec or more may get things running. However, the higher the stall time set, the lower the performance.

If the problem persists, locate the underlying reason for the connection failure. In the case of DNS, it may be a faster DNS server, or a bigger cache on the machine. It could be a port problem where there are not enough ports available for function. This may be related to File Descriptors on a UNIX box.

See also: [Section A.1.2, "JServ Configuration File \(zone.properties\)"](#).

8.5.4 Miscellaneous Issues Using Oracle Portal

Problem: Unable to create *interMedia* Text indexes.

If you encounter any errors while creating an *interMedia* Text index, first check the following:

- Your system should meet all the requirements described in [Section 1.1.8, "interMedia Text Requirements"](#). Your system requires an Oracle8i database with the *interMedia Text* option installed. On Windows NT/2000, you require an 8.1.7 release of the Oracle8i database.
- Also, *interMedia Text* must be installed in the same Oracle Home as the database. It is not installed if you choose to perform a Minimal installation of the Oracle8i database.
- If you are planning to enable *interMedia Text* in Oracle Portal on Windows NT/2000, the following requirements apply:
 - Oracle Portal must be installed in an Oracle 8.1.7 database.
 - Disable connection pooling from the Database Access Descriptor page. For more information, see *Using the PL/SQL Gateway* which is provided as part of the Oracle9i Application Server documentation set.

See also: [Chapter 6, "Setting up the Search Feature in Oracle Portal Content Areas"](#).

Cause	<p>If any of these errors appear in the installation log, a problem creating <i>interMedia Text</i> indexes occurred:</p> <ul style="list-style-type: none"> ■ Cannot grant CTXAPP Role to PORTAL30. ■ ERROR: Creating datastore procedures in CTXSYS. ■ ERROR: Setting up interMedia Text data stores. ■ An unexpected error has occurred (WWS-32100)
--------------	--

Solution	<p>Choose one of the following options to resolve this issue:</p> <ul style="list-style-type: none"> ■ Access the database server and log on using the user name and password for the schema that owns the Oracle Portal content area. For example, if the schema name is "SCOTT", log on with the user name "SCOTT" and the appropriate password. <p>Start SQL*Plus and execute the <code>inctxgrn.sql</code> script. This script is located in <code><ORACLE_HOME>\portal30\admin\plssql\wws</code>. Running this script creates the <i>interMedia</i> preferences required for Oracle Portal.</p> <ul style="list-style-type: none"> ■ If you do not have access to the database server, but you do have a copy of the <code>sbrimtlx</code> script, you can connect to the database using SQL*Plus as the schema owner and run the following commands: <pre>set serveroutput on size 10000 begin wwv_context_util.grantCtxRole(user); end; @@sbrimtlx</pre> <p>Replace (user) with the Oracle Portal schema owner, for example, portal30.</p>
-----------------	---

Problem: Apache generates the following error on startup: "The procedure entry point snlpcgtsrvbynm could not be located in the dynamic link library oranl8.dll."

Cause	<p>The primary ORACLE_HOME contains Oracle 8.1.6 client libraries. This most likely happened because an 8.1.6-based Oracle product was installed after installing Oracle9i Application Server and the primary ORACLE_HOME was changed during the installation. This problem has been confirmed when installing Oracle Internet Directory (OID).</p>
Solution	<p>Change the values of the PATH variable so that it points to the Oracle9i Application Server Oracle Home before the OID home. If the DLLs are forward-compatible, then both Oracle9i Application Server and OID should work with this change. Alternatively, use the Oracle Home Selector utility to switch between different Oracle Homes. This utility solves incompatibilities and switches between different Oracle Homes. See the installation guide for your Oracle8i database for details.</p>

Error: missing string (login link text) language(e) domain(wwc) sub_ domain(sec) Missing string(pages) language(e) domain(wwc) sub_ domain(pob).

Cause	This error may display when you try to access the Oracle Portal home page and National Language Support (NLS) files are missing.
Solutions	<p>You can solve in any of the following ways:</p> <ul style="list-style-type: none"> ▪ Check to make sure your browser language preference is set to the language you've installed with Oracle Portal. Oracle Portal is supported on 24 different languages with English as the default language. See Section 2.6, "Installing Language Support in Oracle Portal". ▪ Make sure that your browser is set to accept Java and JavaScript. ▪ Check the installation log file to see if <code>wwvcbus.ctl</code> and <code>wwcus.ctl</code> were loaded into Oracle Portal's NLS table, <code>wnls_strings\$</code>.

Error: PLS-00306: wrong number or types of arguments.

Cause	This error occurs when you are installing Oracle Portal on a machine with an existing Oracle WebDB 2.2. The installed Oracle Portal synonym (<code>webdb30.wwv_utl_api_types</code>) causes the WebDB 2.2 components to become invalid and you'll experience problems with your pre-existing components and creating new components in WebDB 2.2.
Solutions	<p>You'll need to drop the WebDB 2.2 synonyms as follows:</p> <ol style="list-style-type: none"> 1. Log on to SQL*Plus as the SYS user with the appropriate password. 2. Enter the following commands: <pre>drop public synonym wwv_utl_api_types; create public synonym wwv_utl_api_types for <webdb22schema>.wwv_utl_api_types</pre>

Oracle9i Application Server Configuration Files

This appendix provides information about the configuration files which can affect the connection to and the behavior of the Oracle9i Application Server and its components in the middle-tier as well as on other machines to which it is connecting.

Specific topics covered include:

- **Control Points and File Locations**
 - Oracle HTTP Server Configuration File (`httpd.conf`)
 - JServ Configuration File (`zone.properties`)
 - Database Access Descriptor (DAD) Configuration File (`wdbsvr.app`)
 - Oracle Database Connection File (`tnsnames.ora`)
 - Login Server Configuration Table
 - Login Server's Partner Application Table
 - Local HOSTS File

A.1 Control Points and File Locations

[Table A-1](#) lists the important Oracle9i Application Server configuration control points and their locations.

Table A-1 List of Oracle Portal Control Points

Configuration File/table	Location or Description
Oracle HTTP Server	<ORACLE_HOME>/Apache/Apache/conf/httpd.conf
Apache JServ	<ORACLE_HOME>/Apache/Jserv/servlets/zone.properties
PL/SQL Gateway	<ORACLE_HOME>/Apache/modplsql/cfg/wdbsvr.app
Database Connection	<ORACLE_HOME>/network/admin/tnsnames.ora
Login Server enabler table	Oracle Portal and Login Server's configuration table, WWSEC_ENABLER_CONFIG_INFO\$.
Login server configuration table	Login Server's Partner Applications configuration table, WSSO_PAPP_CONFIGURATIION_INFO\$.
Local HOSTS file	This file resolves IP names to IP addresses. On Windows NT, this file is located at \winnt\system32\drivers\etc\hosts. On Unix, this file is at /etc/hosts.

<ORACLE_HOME> is the location of your Oracle9i Application Server installation.

A.1.1 Oracle HTTP Server Configuration File (httpd.conf)

The Oracle HTTP Server *powered by Apache* configuration file, `httpd.conf`, contains configuration information for running the Oracle HTTP Server. The contents of this file includes information about listening ports, server names, virtual hosts, proxy configurations, and the like. Also, configuring Secure Sockets Layer (SSL) support by defining information such as certificates and other HTTPS configuration directives is done in this file.

<ORACLE_HOME>/Apache/Apache/conf/httpd.conf

See: [Section 2.11, "Enabling Secure Socket Layer \(SSL\)".](#)

A.1.2 JServ Configuration File (zone.properties)

The `zone.properties` file contains information specific to the servlets being run under the Apache JServ environment. This is the environment where the Parallel Page Engine servility runs. This file sets up specific parameters which the servlets use for initialization.

This file is used by the servlets at initialization time. The Parallel Page Engine uses this file to get certain required information for it to run properly. If configuring the Portal to run in SSL mode, one of the items needed here is to tell the Parallel Page Engine which ports are secure ports. This way it knows what protocol should be used on different ports. You can add as many ports as needed for secure communication by separating the port numbers with a colon ":".

See also: [Section 2.10.2, "Securing Ports to Use Certificates and HTTPS"](#)

A.1.3 Database Access Descriptor (DAD) Configuration File (wdbsvr.app)

The `wdbsvr.app` file contains the definition of the Database Access Descriptors (DADs) which define the connection information that the PL/SQL Gateway (`mod_plsql`) uses to connect to a particular database schema.

The DAD is the entity which informs the middle-tier which database to connect to, and the username and password for connecting to that database. By specifying the appropriate connect string, the middle-tier can connect to a database on any remote host, or may connect to a database on the local machine.

See also: *Using the PL/SQL Gateway* which is provided as part of the Oracle9i Application Server documentation set.

[Table A-2](#) lists the configuration parameters for setting up a DAD for use with the Login Server or Oracle Portal.

Table A-2 Database Access Descriptor (DAD) configuration parameters

Parameter	Description
User Name	<p>Name of the Oracle Portal product schema.</p> <p>Default = PORTAL30</p> <p>The user name and password specifies the "trusted" user for the entire portal instance. When the Oracle9i Application Server middle-tier connects to Oracle Portal, it initially connects as the "trusted" user specified in the DAD. Once the connection is established, the trusted user becomes the actual database user for the portal user (i.e. portal30_public), using the connect through functionality supported in the database. As a result, all database users on which you want to map portal users must be granted connect through privileges through the trusted user.</p> <p>If you create the new schema in Oracle Portal, select the "Use this schema for Portal users" check box in the Create Schema dialog box. Oracle Portal performs this grant for you automatically.</p>
Password	<p>Password to the Oracle Portal product schema.</p> <p>Note: Any changes to this password from the database (for example, using "alter user"), should be updated in the DAD as well. Otherwise, a "Proxy Logon Failed" error may occur.</p>
Connect String	<p>The connect string defines where the schema specified in User Name resides.</p> <p>The connect string can be specified as:</p> <p><code><hostname>:<port>:<sid></code></p> <p>where: <i>hostname</i> is where the database is installed, <i>port</i> is the database port, and <i>sid</i> is the database instance.</p> <p>The connect string can also be specified in the tnsnames.ora file associated with the <ORACLE_HOME> of the Oracle9i Application Server installation.</p>
Authentication Mode	<p>For Oracle Portal and the Login Server, set the mode to "Single Sign-On."</p> <p>This mode provides for session creation and uses N-Tier authentication to access the database.</p>
Default Page	<p>Set the default page to the home procedure in the Oracle Portal product schema.</p> <p>Default = PORTAL30.home</p>

See also: *Using the PL/SQL Gateway* which is provided as part of the Oracle9i Application Server documentation set.

A.1.4 Oracle Database Connection File (tnsnames.ora)

The `tnsnames.ora` file defines the entries that can be used as connect strings in the DADs.

Also, the `tnsnames.ora` file in the Oracle Home location containing your Oracle9i Application Server must have a connect string entry pointing to the database where your Oracle Portal installation is located.

In the C shell, for example, type the following at a command line prompt:

```
setenv TNS_ADMIN path
```

`path` points to the `tnsnames.ora` file. This command differs depending on the shell.

See also: *Net8 Administrator's Guide* in the Oracle8i database documentation library.

A.1.5 Login Server Configuration Table

The `WWSEC_ENABLER_CONFIG_INFO$` table is the configuration table for the Single Sign-on enabler stack.

Each Partner Application to the Login Server has such a table for configuration information. As such, one such table exists in the Oracle Portal schema as well as the Login Server schema, since the Login Server application is a Partner Application as well. This table defines the login URL for the Login Server which this Partner Application is associated with.

It is important to understand how the `LSNR_TOKEN` is used in the enabler configuration table in order to 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 purposes of this discussion, the main flows include:

- Initial request to the requested URL.

- Redirect to the Login Server for authentication.
- Redirect to Oracle Portal's success URL (`wwsec_app_priv.process_signon`).
- Redirect back to the requested URL.

The Login Server (SSO) partner enabler APIs read the `WWSEC_ENABLER_CONFIG_INFO$` table for configuration information. Similarly, in the Login Server, the Login Server's private APIs read the `WWSSO_PAPP_CONFIGURATION_INFO$` table. In the latter table, the URL that should be redirected to each Partner Application.

Since each Partner Application's success URL is stored in the Login Server'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 Login Server 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 Oracle Portal to be accessible from `http://www.xyz.com` as well as `http://www.abc.com`. In this case, two Partner Applications must be registered in the Login Server. 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/pls/portal30/portal30.wwsec_app_priv.process_signon` for the `www.xyz.com` partner
- `http://www.abc.com/pls/portal30/portal30.wwsec_app_priv.process_signon` for the `www.abc.com` application

Each of these Partner Application entries on the Login Server would have a distinct site id, site token, and encryption key. Oracle Portal's enabler configuration table has one row for each Partner Application, for example:

LSNR_TOKEN	SITE_ID	LS_LOGIN_URL	...
<code>www.xyz.com</code>	<code>1321</code>	<code>https://www.login.com/pls/...</code>	
<code>www.abc.com</code>	<code>1322</code>	<code>https://www.login.com/pls/...</code>	

See also: For more information, see the *Oracle9iAS Single Sign-On Application Developer's Guide* included in the Oracle9i Application Server documentation library.

A.1.6 Login Server's Partner Application Table

The configuration table on the Login Server's side is the Partner Application Table, `WSSO_PAPP_CONFIGURATION_INFO$`. Maintenance of this table is typically done using the Login Server application's user interface for Adding or Editing Partner Applications.

For an initial installation on a single database instance, the `ssodatan` script populates both the Login Server's partner configuration table as well as Oracle Portal's enabler configuration table.

If running the `linstall` script to install a standalone Login Server, this script populates the Login Server's enabler configuration table and Partner Application configuration table as part of the installation. See [Section B.3, "Manually Installing a Login Server with the linstall Script"](#).

A.1.7 Local HOSTS File

The HOSTS file on a network host defines mappings of IP names to IP addresses. Normally, the association of IP name to IP address is provided by a Domain Name Server (DNS). In some of the configurations described in [Chapter 3, "Basic Oracle Portal Configurations"](#), 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.

Oracle Portal Installation and Configuration Scripts

After installing Oracle Portal with the Oracle9i Application Server installation, several scripts are made available for post-installation configuration. For example, you may want to customize the configuration and install additional components, such as a standalone Login Server, additional Oracle Portal nodes, or load additional language translations into Oracle Portal.

Specific topics covered include:

- [Overview](#)
- [Manually Installing Oracle Portal with the wininstall Script](#)
- [Manually Installing a Login Server with the linstall Script](#)
- [Configuring a New Oracle Portal Instance and Login Server with the ssodatan Script](#)
- [Updating an Existing Portal Instance with the ssodatax Script](#)
- [Modifying the Scope of the Portal Session Cookie](#)
- [Managing the Session Cleanup Job](#)

See:

- For information about the Oracle Portal import and export scripts, see the Oracle Portal Online Help topic: *Exporting and importing in Oracle Portal*.
- For information about Oracle Portal upgrade scripts, visit:

<http://technet.oracle.com/products/iportal>

B.1 Overview

For purposes of configuring Oracle Portal, the following scripts are useful, and are described in this appendix:

Table B-1 Oracle Portal installation scripts

Script	Description
winstall	<p>This script allows you to perform a manual installation of Oracle Portal. If you already have a Login Server installed, install a single portal node without having to install another Login Server.</p> <p>See Section B.2, "Manually Installing Oracle Portal with the winstall Script".</p>
linstall	<p>This script allows you to perform a manual installation of a standalone Login Server, without a corresponding Oracle Portal installation.</p> <p>See Section B.3, "Manually Installing a Login Server with the linstall Script".</p>
ssodatan	<p>This script is responsible for setting up the configuration information associated with a newly-installed Oracle Portal node with a Login Server.</p> <p>Run this script on a newly-installed host in which the Login Server and Oracle Portal are on the same database instance and are being setup for the first time.</p> <p>See Section B.4, "Configuring a New Oracle Portal Instance and Login Server with the ssodatan Script".</p>
ssodatax	<p>This script is responsible for setting up the configuration information associated with an Oracle Portal node with a Login Server.</p> <p>This script is also used to add configuration entries to support server aliases or virtual host names. A different partner configuration entry is required for each alias to be used. This script must also be used in any instance when the Oracle Portal node is on a separate database instance from the Login Server.</p> <p>Section B.5, "Updating an Existing Portal Instance with the ssodatax Script".</p>

Note: These scripts are located in the following directory upon installation:

<ORACLE_HOME>/portal30/admin/plsql/

B.2 Manually Installing Oracle Portal with the winstall Script

You can perform a manual installation of Oracle Portal by running the `winstall` script or you can create a Login Server for the Oracle Portal by invoking this script.

To manually install Oracle Portal with the `winstall` script, complete the following:

- [Step 1: Manually Install Oracle Portal](#)
- [Step 2: Verify the Manual Install](#)
- [Step 3: Run the ssodatan script](#)
- [Step 4: Edit wdbsvr.app](#)
- [Step 5: Stop and Restart the Oracle HTTP Server](#)

Note: A manual installation of Oracle Portal completes in 1 1/2 - 3 hours.

Step 1: Manually Install Oracle Portal

Follow these steps to manually install Oracle Portal:

1. Make sure that your Oracle8i database is up and running.
2. Verify that the Net8 listener is running by issuing the following command:

```
lsnrctl start listener
```
3. Ensure that all your ORCL database environment variables are set accordingly. See [Chapter 1, "Verifying Requirements"](#).
Ensure that your ORCL `<ORACLE_HOME>/bin` is set in the `PATH` variable *before* any other `ORACLE_HOME`. Otherwise, you may encounter LoadJava or JDBC type errors.
4. Ensure you have a working TNS Alias that points to your ORCL database. Test your connection with SQL*Plus.
5. Ensure that your ORCL `<ORACLE_HOME>/bin/loadjava` works correctly. Type `loadjava -help` to test it and display the help usage messages.
6. (*optional*) If you successfully installed a previous Oracle Portal schema into this database either using the OUI or manually, skip this step.

Otherwise, change the directory to the following to run the following onetime script:

```
<ORACLE_HOME>/portal30/admin/plsql
```

In a command box, run the onetime script file to install the necessary Oracle Portal packages as follows:

```
onetime [-p sys_password] [-l logfile] [-c connect_string]
```

Example `onetime -p change_on_install -l onetime.log -c orcl`

Note: Run this command once for each database where you want to install Oracle Portal. Ignore any ORA messages about dropping objects that do not exist.

7. In a command box, run the Oracle Portal installation script, `winstall`, from the following location to install the Oracle Portal product.

```
<ORACLE_HOME>/portal30/admin/plsql
```

Run the Oracle Portal installation script parameters as follows:

Syntax - Usage 1

```
winstall <-s portal_schema> <-p sys_password> <-u default_tablespace>  
<-t temporary_tablespace> <-d document_tablespace> <-l logging_tablespace>  
<-w workflow_schema> <-o sso_schema> <-i pstore_password> <-r random_seed>  
<-c connect_string>
```

Syntax - Usage 2

```
winstall list_tablespaces [sys_password] [connect_string]
```

Syntax - Usage 3

```
winstall uninstall sys_password portal_schema [sso_schema][connect_string]
```

Example

```
winstall -s portal30 -p change_on_install -u users -t temp -d users -l users  
-o portal30_sso -i portal30_sso_ps -r 12345 -c orcl > winstall.log
```

The parameter descriptions are provided in the `winstall` file which opens with any text editor. All arguments are validated before the installation starts.

By default, five schemas are created: `portal30`, `portal30_sso`, `portal30_sso_public`, `portal30_public`, `portal30_demo`. The default base schema name and password is `portal30` which you can change at installation time. For more information, see [Section 2.1, "Oracle Portal Default Schemas"](#).

Install an Oracle Portal node without an associated Login Server by invoking `winstall` with the following usage:

```
winstall -s newportalnode -nosso
```

In the preceding example, executing this command installs an Oracle Portal node in a schema named `newportalnode` without loading an associated Login Server schema. Usage of this command is useful for installing nodes for a distributed Oracle Portal installation as discussed in [Chapter 4, "Distributed Oracle Portal Installations"](#).

Step 2: Verify the Manual Install

You should check the installation session log that describes the actions performed and the components created upon installation. GREP the log file for "ORA-", "PLS-", and "ERROR:" that may have occurred during installation. The log file is located in:

```
<ORACLE_HOME>/assistants/opca/install.log
```

Note: If you need to abort or re-run the `winstall` script in case of failure, first drop all Oracle Portal schemas that were created using SQL*Plus. For example, enter:

```
SQLPLUS > drop user <schema> cascade;
```

You might need to stop and restart the database to release any defunct or runaway database processes that are locking your Oracle Portal schema before re-running `winstall`.

Proceed with the next step only if you successfully ran the `winstall` script and Oracle Portal was installed properly.

Step 3: Run the `ssodatan` script

If you install an Oracle Portal node with the `winstall` script, you need to run the `ssodatan` script afterwards to establish the linkage between the Portal node and the Login Server.

However, if the installation of the Oracle Portal node does *not* include the Login Server, or you are linking to an existing Login Server, then run the `ssodatax` script to perform the linkage. See [Section B.5, "Updating an Existing Portal Instance with the `ssodatax` Script"](#).

In a command box, run the `ssodatan` script from this location:

```
<ORACLE_HOME>/portal30/admin/plsql
```

This script configures your Single Sign-On (SSO) login via the Login Server, which is installed as part of Oracle Portal.

For example, if your settings are as follows:

```
DAD                = simpledad
;(from <ORACLE_HOME>/Apache/modplsql/cfg/wdbsvr.app )
SSO DAD            = ssodad
;(from <ORACLE_HOME>/Apache/modplsql/cfg/wdbsvr.app )
Portal schema     = portal30
Host name          = myhost
IAS/Apache port   = 7777
TNS Alias          = ORCL
```

You would run the `ssodatan` script as follows in one continuous line. If you use port 80, then omit the colon ':' and the port number altogether:

Also, the hostname must match the `ServerName` parameter in your `<ORACLE_HOME>/Apache/Apache/conf/httpd.conf` file.

```
./ssodatan -w http://hostname:7777/pls/simpledad/ -l
http://hostname:7777/pls/ssodad/ -s portal30 -o portal30_sso -c orcl >
ssodatan.log
```

This script completes in a few seconds. Verify that there were no "ORA-" or "PLS-" type errors in your `ssodatan.log` file. If there were such errors, make sure you correct these before re-running the `ssodatan` script.

See also:

- [Appendix A, "Oracle9i Application Server Configuration Files"](#)
 - [Section B.4, "Configuring a New Oracle Portal Instance and Login Server with the `ssodatan` Script"](#)
-
-

Step 4: Edit wdbsvr.app

1. Go to the <ORACLE_HOME>/Apache/modplsql/cfg directory.
2. Create a backup copy of the PL/SQL Gateway configuration file, wdbsvr.app.
3. Edit the wdbsvr.app file by removing all existing entries, and then adding these initial settings (using the example values):

```

;
[WVGATEWAY]
defaultDAD = simpledad
administrators = all
adminPath = /admin_/
admindad =
;upload_as_long_raw =
;upload_as_blob =
;debugModules =
;
[DAD_simpledad]
connect_string = orcl
password = portal30
username = portal30
default_page = portal30.home
document_table = portal30.wwdoc_document
document_path = docs
document_proc = portal30.wwdoc_process.process_download
upload_as_long_raw =
upload_as_blob = *
reuse = Yes
connmax = 10
enablesso = Yes
pathalias = url
pathaliasproc = portal30.wwpth_api_alias.process_download
;name_prefix =
;always_describe =
;after_proc =
;before_proc =
;
[DAD_%sso_DAD%]
connect_string = orcl
password = portal30_sso
username = portal30_sso
default_page = portal30_sso.wwsso_home.home
document_table = portal30_sso.wwdoc_document
document_path = docs
document_proc = portal30_sso.wwdoc_process.process_download

```

```
upload_as_long_raw =
upload_as_blob = *
reuse = Yes
connmax = 10
enablesso = Yes
pathalias = url
pathaliasproc = portal30_sso.wvpth_api_alias.process_download
;name_prefix =
;always_describe =
;after_proc =
;before_proc =
;
[DAD_sample]
connect_string = sample-tcp
password = sample
username = sample
default_page = sample.home
document_table = sample.wvdoc_document
document_path = docs
document_proc = sample.wvdoc_process.process_download
upload_as_long_raw =
upload_as_blob = *
reuse = Yes
connmax = 10
enablesso = Yes
pathalias = url
pathaliasproc = sample.wvpth_api_alias.process_download
;name_prefix =
;always_describe =
;after_proc =
;before_proc =
;
```

Step 5: Stop and Restart the Oracle HTTP Server

Stop and restart the Oracle HTTP Server *powered by Apache* with the following commands:

- `<ORACLE_HOME>/Apache/Apache/bin/apachectl stop`
- `<ORACLE_HOME>/Apache/Apache/bin/apachectl start`

With Secure Sockets Layer, the start command is:

- `<ORACLE_HOME>/Apache/Apache/bin/apachectl startssl`

Note: Access the Gateway Configuration Menu page by pointing your browser to the following location:

- `http://hostname:7777/pls/simplicated/admin_/gateway.htm`

For more information, see *Using the PL/SQL Gateway* which is provided as part of the Oracle9i Application Server documentation set.

B.2.1 Manually Installing Online Help

The Oracle Portal Online Help Content Area contains conceptual, getting started, step-by-step, and troubleshooting Help topics. If you manually install Oracle Portal, choose to install the Oracle Portal online help.

In the `<ORACLE_HOME>/portal30/admin/plsql/wwu` directory, run the following script (in one continuous line):

Syntax

```
./contimp.csh -s portal30 -p portal30 -o portal_help -m reuse -u  
database_user -d ../../../../doc/site/hlp30ca.dmp -c orcl
```

Notes:

- Enter the "-o", "-m", "-u", "-d" static values.
 - Change the "-s", "-p", "-c" options as appropriate to your environment.
-
-

This process takes about three minutes to complete. Ignore the following messages after the script is run:

```
security.dmp: No such file or directory  
pobpage.dmp: No such file or directory  
Unable to set user acl for:
```

See also: For additional information, see the appropriate manual installation document at:

<http://technet.oracle.com/products/iportal>

B.3 Manually Installing a Login Server with the linstall Script

To install an instance of the Login Server which resides in a database instance without an associated Oracle Portal, execute the `linstall` script.

This script installs the Login Server schema and also automatically sets the configuration information that registers the Login Server application as a Partner Application to the Login Server itself.

Syntax

```
linstall <-o sso_schema> <-i pstore_password> <-s login_server_url> <-r random_seed> <-p sys_password> <-u default_tablespace> <-t temporary_tablespace> <-d document_tablespace> <-l logging_tablespace> <-c connect_string>
```

Example

```
linstall -o portal30_sso -i portal30_sso_ps -s
http://server.oracle.com:3000/pls/portal30_sso/ -r 12345 -p change_on_install
-u users -t temp -d users -l users -c orcl
```

where

Table B-2 *linstall* script parameters

Parameter	Description
<code>-o sso_schema</code>	Oracle database schema containing the Login Server installation (database objects). Default = <code>portal30_sso</code>
<code>-i pstore_password</code>	The password for the password store. The password store schema name is <code><sso_schema>_PS</code> .
<code>-s login_server_url</code>	URL that points to the Login Server host and DAD. Include the full domain name with the host, <code>http://</code> prefix in your URL, and end the URL with a forward slash (/). Note: If you are not using the default port number (for example, 443), specify it in the Login Server URL.

Table B-2 *install script parameters*

Parameter	Description
-r random_seed	Optional specification of a random string to be used to seed the random number generator for initializing encryption keys.
-p sys_password	Oracle database password for the SYS schema. Default = change_on_install
-u default_tablespace	Default = USERS
-t temporary_tablespace	Default = TEMP
-d document_tablespace	Default = <default_tablespace>
-l logging_tablespace	Default = <default_tablespace>
-c connect_string	Optional connect string to a remote database.
-casesensitive	Enforce case-sensitivity on passwords.

B.4 Configuring a New Oracle Portal Instance and Login Server with the ssodatan Script

The `ssodatan` script sets up a fresh Oracle Portal and a new Login Server. Running this script completely removes any pre-existing configuration information in the Login Server configuration and replaces it with the information specified in the most recent invocation of this script.

Alternatively, if you are associating a Portal node with an *existing* Login Server that contains configuration information you want to retain, do not use this script. Instead, invoke the `ssodatax` script which is described in the next section, [Section B.5, "Updating an Existing Portal Instance with the ssodatax Script"](#).

The `ssodatan` script can only be used on an Oracle Portal and Login Server which reside on the *same* database instance. If you are setting up a configuration with separate database instances for the Portal and the Login Server, then use a combination of the `linstall` and `ssodatax` script to set this up.

In the case of the single instance, where you want to associate a fresh association of Oracle Portal with Login Server, invoke the `ssodatan` script as follows:

Syntax (full usage)

```
ssodatan <-w portal_url> <-l login_server_url> <-s portal_schema> <-p portal_
password> <-o sso_schema> <-d sso_password> <-e pstore_schema> <-c portal_
connect_string>
```

Syntax (Login Server only)

```
ssodatan -loginserver <-l login_server_url> <-o sso_schema> <-d sso_password>
<-c portal_connect_string>
```

Example (full usage)

```
ssodatan -w http://webdbsvr.us.oracle.com:3000/pls/portal/ -l
http://webdbsvr.us.oracle.com:3000/pls/portal_sso/ -s portal30 -p portal30 -o
portal30_sso -d portal30_sso -e portal30_sso_ps -c orcl
```

Example (skip optional parameters)

```
ssodatan -w http://webdbsvr.us.oracle.com:3000/pls/portal/ -l
http://webdbsvr.us.oracle.com:3000/pls/portal_sso/ -s portal30
```

Example (standalone Login Server)

```
ssodatan -loginserver -l http://webdbsvr.us.oracle.com:3000/pls/portal_sso/ -o
portal30_sso -d portal30_sso -c orcl
```

Note: This shell script installs the appropriate seed data to set up a connection between Oracle Portal and a newly-installed Login Server in SQL*Plus.

Table B-3 lists the ssodatan script parameters.

Table B-3 ssodatan script parameters

Parameter	Description
-w portal_url	URL that points to the Oracle Portal node DAD. Include the full domain name with the host, http:// prefix in your URL, and end the URL with a forward slash (/). Note: If you are not using the default port number (for example, 80), specify it in the Oracle Portal URL.

Table B-3 *ssodatan script parameters*

Parameter	Description
-l login_server_url	URL that points to the Login Server host and DAD. Include the full domain name with the host, http:// prefix in your URL, and end the URL with a forward slash (/). Note: If you are not using the default port number (for example, 443), specify it in the Login Server URL.
-s portal_schema	Oracle database schema containing the Oracle Portal installation (database objects). Default = portal30
-p portal_password	Oracle database password for the Oracle Portal schema. Default = <portal_schema>
-o sso_schema	Oracle database schema for Login Server objects. Default = <portal_schema>_SSO
-d sso_password	Oracle database password for Login Server schema. Default = <sso_schema>
-e pstore_schema	Oracle database password for access to password store objects.
<-c portal_connect_string>	Connect string for the database in which the Oracle Portal schema is installed. You need to provide the connect string only if the Oracle Portal schema is located on a remote database. Default = null
-loginserver	Sets up the enabler configuration entry for the standalone Login Server. Do not use the -w, -s, -p, and -e options with this parameter.

Warning: If you are using port 80, which is the default HTTP port number, do not include the port number in the Portal URL prefix or the Login Server URL prefix.

Also, do not specify the port in the URL that is used to access the Web site since some browsers exclude the port in the URL if it is the default port, resulting in a problem selecting the appropriate enabler configuration entry.

B.5 Updating an Existing Portal Instance with the ssodatax Script

The `ssodatax` script updates the Partner Application's enabler configuration table, `WSSO_PAPP_CONFIGURATION_INFO$`. However, before running this script, first update the Login Server's Partner Application configuration table to create the entries on the Login Server side in the following way:

1. On the Oracle Portal Home Page, click the **Administer** tab.
2. In the **Services** portlet, click **Login Server Administration**.
3. Click **Administer Partner Applications**.
4. Click **Add Partner Application**.
5. In the **Partner Application Login** section, enter the Partner Application's name, the URL to the application's home page, and a success URL.
6. In the **Valid Login Timeframe** section, enter the dates when users can log on to the application through the Login Server. If you leave the **End Date** field blank, users can log on to the application indefinitely. In the **Application Administrator** section, enter an e-mail address and other information for the application's contact person or administrator.
7. Click **OK**.

The new Partner Application appears in the Edit/Delete Partner Application list on the Partner Application page. After adding a Partner Application entry on the Login Server, the Login Server generates a site id, site token, and encryption key for the new Partner Application. These are used as input when invoking the `ssodatax` script.

In the case where you want to add or update entries in the enabler configuration table, invoke the `ssodatax` script as follows:

Syntax (full usage)

```
ssodatax <-i portal_site_id> <-t portal_site_token> <-k encryption_key> <-w portal_url> <-l login_server_url> <-s portal_schema> <-p portal_password> <-v cookie_version> <-o sso_schema> <-e pstore_schema> <-r pstore_password> <-b pstore_dbblink> <-c connect_string> <-n ps_connect_string>
```

Syntax (remove enabler configuration entry)

```
ssodatax <-s portal_schema> <-p portal_password> <-remove portal_host> <-b pstore_dbblink> <-c connect_string>
```

Example (full usage)

```
ssodatx -i 1234 -t A1B2C3 -k X9Y8Z7 -w
http://webdbsvr.us.oracle.com:3000/pls/portal30/ -l
http://webdbsvr.us.oracle.com:3000/pls/portal30_sso/ -s portal30 -p portal30 -v
v1.1 -o portal30_sso -e portal30_sso_ps -r portal30_sso_ps -b portal30_dblink -c
orcl -n orcl01
```

Example (skip optional parameters)

```
ssodatx -i 1234 -t A1B2C3 -k X9Y8Z7 -w
http://webdbsvr.us.oracle.com:3000/pls/portal30/ -l
http://webdbsvr.us.oracle.com:3000/pls/portal30_sso/ -s portal30
```

Example (deletion)

```
ssodatx -s portal30 -remove webdbsvr.us.oracle.com:3000 -b portal30_dblink
```

Table B-4 lists the ssodatx script parameters.

Table B-4 *ssodatx script parameters*

Parameter	Description
-i portal_site_id	The ID is automatically set when a Partner Application (in this case, Oracle Portal installation) is added. It is used by the Login Server to identify the Partner Application to this node.
-t portal_site_token	The token is automatically set when a Partner Application (in this case, Oracle Portal installation) is added. It is used by the Login Server to identify the Partner Application. The Partner Application must use the application token to identify itself to the Login Server to this node when requesting authentication.
-k encryption_key	When a user tries to log on to this Oracle Portal node using Single Sign-On, the Login Server generates a cookie that indicates a user's identity and whether the user has been authenticated. This key encrypts the login cookie.
-w portal_url	URL prefix to this Oracle Portal node and the DAD being accessed. Include the full domain name with the host, http:// prefix in your URL, and end the URL with a forward slash (/).
-l login_server_url	URL prefix to the Login Server host and DAD. Include the full domain name with the host, http:// prefix in your URL, and end the URL with a forward slash (/).
-s portal_schema	Oracle database schema containing the Oracle Portal installation (database objects). Default = portal30

Table B-4 *ssodatax script parameters*

Parameter	Description
-p portal_password	Oracle database password for the Oracle Portal schema. Default = <portal_schema>
-v cookie_version	Cookie version being used by the Login Server." Default = v1.1
-o sso_schema	Oracle database schema for Login Server objects. Default = <portal_schema>_SSO
-e pstore_schema	Oracle database schema used for access to the password store. Default = <sso_schema>_PS
-r pstore_password	Oracle database password for Password Store access schema. Default = <pstore_schema>
-b pstore_dblink	Name of the database link for connecting from the Oracle Portal schema to the Password Store access schema across database instances. Default = <portal_schema>_DBLINK
-c connect_string	Connect string for the database in which the Oracle Portal schema is installed. Provide the connect string only if the Oracle Portal schema is located on a remote database. Default = null
-n ps_connect_string	Connect string to connect to the Password Store access schema on a remote database. Default = null
-remove portal_host	Removes the enabler configuration entry associated with the specified portal host.

The `ssodatax` script removes entries from the enabler table with the `-remove` option. When you use this option, the following parameters are applicable:

- `-s portal_schema`
- `-p portal_password`
- `- remove portal_host`

The `portal_host` is the value of the `lsnr_token` to remove from the enabler table.

- `-c connect_string`

See also: If you have any problems starting Oracle Portal, see [Chapter 8, "Troubleshooting"](#) or run the diagnostics tool.

B.6 Modifying the Scope of the Portal Session Cookie

In cases where you want to install a distributed Oracle Portal environment, and you need to have more than one distinctly named middle-tier server, you need to define the scope of the Oracle Portal session cookie to be sent to all the middle-tier servers involved in the architecture. By default, the session cookie is scoped to the host from which it was generated which is typically the root path.

For example, if the cookie was generated from `www.oracle.com`, then the cookie domain is `www.oracle.com`. However, let's say that another server, `portal.oracle.com` is also a middle-tier server that needs to get access to that session cookie, then the cookie domain would need to be widened so that the `portal.oracle.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>/portal30/admin/plsql/wwc
```

2. On the database where your Login Server schema is installed, log on to SQL*Plus with the appropriate username and password. For example:

```
sqlplus nodea/nodea
```

3. Enter the following command:

```
SQL> @ctxckupd
Oracle Portal
Current Settings for Portal Session Cookie:
Cookie Domain : Only send cookie back to originating host:port
Enter the domain for the session cookie: .oracle.com
Settings changed to
Cookie Domain : .oracle.com
SQL>
```

This allows you to set the cookie domain for the session cookie. In the example above, the cookie domain is set to `.oracle.com`.

Tip:

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 Oracle Portal from two machines:

- machine1.us.oracle.com:3000
- machine2.us.oracle.com:4000

When running `ctxckupd.sql`, set the cookie domain to `.us.oracle.com`.

See: [Section 4.4.2, "Step 2: Create Same Cookie Domain"](#).

B.7 Managing the Session Cleanup Job

Oracle Portal and the Login Server perform session management similar to other web-based applications. Sessions are tracked using cookies. Session information is stored in a table in the Portal and Login Server 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 Oracle 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 Portal schema to manage Portal sessions, or in the Login Server schema to manage Login Server 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>/portal30/admin/plsql/wwc
```

2. On the database where the Portal or Login Server schema is installed, log on to SQL*Plus with the appropriate user name and password for that schema. For example:

```
sqlplus portal30/portal30
```

3. Enter the following command to get the current job information:

```
SQL> @ctxjget
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 command results in the display of the currently installed job information, as returned by the DBMS_JOB package. It indicates 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 cleanup active sessions which 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 needs to 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>/portal30/admin/plsql/wwc
```

2. On the database where the Portal or Login Server schema is installed, log on to SQL*Plus with the appropriate user name and password for that schema. For example:

```
sqlplus portal30/portal30
```

3. Enter the following command to submit new cleanup job information:

```
@ctxjsub <hours_old> <start_time> <time_format> <interval_hours>
```

Table B-5 lists the `ctxjsub` parameters.

Table B-5 *ctxjsub parameters*

Parameter	Description
<code>hours_old</code>	The age of an active session that should be deleted.
<code>start_time</code>	The time that the next job should run.
<code>time_format</code>	The time format string that specifies how <code>start_time</code> is formatted.
<code>interval_hours</code>	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
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, you need to resubmit the `start_time` desired using `ctxjsub.sql`.

Index

Numerics

503 error, 1-5
503 Service Temporarily Unavailable, 8-23

A

accounts
 letting users create own, 2-12
aliases, 8-19
 supporting multiple, 5-15
Apache, 8-21
 log file locations, 8-2
 multi-threaded on Windows NT, 8-23
 setting the number of requests that can be handled, 5-19
 tuning process configuration, 8-24
apachectl, 8-17
audience, xiii
audio
 adding interMedia rich content, 7-2
AUTHENTICATED_USERS
 default group created, 2-4

B

BLOB, 7-6, 7-10
browsers
 accessing Oracle Portal, 2-5
 communication with Web Server, 2-20
 cookies sent to portlets, 4-5
 simplifying Oracle Portal URL, 2-6
 supported by Oracle Portal, 1-4
 system requirements, 1-4

C

certificates
 using with HTTPS, 2-16
connect string, 3-8
 specified in wdbsvr.app, A-4
connection pooling, 1-6, 5-1, 6-3, 8-24, 8-27
content areas
 creating, 6-3
 part of Oracle Portal schema, 1-4
contimp.csh, B-9
cookie domains
 creating same across nodes, 4-8 to 4-9
 modifying the scope to send to all middle-tier servers, B-17
 node requirement, 4-5
ctx_schedule, 6-16
ctxckupd.sql, 4-6, 4-9, B-17
ctxcrind.sql, 6-14
ctxdrind.sql, 6-18
ctxsrv, 6-9
ctxsys, 6-20, 8-27

D

Database Access Descriptor (DAD)
 common cookie name required, 4-6
 configuring in middle-tier servers, 5-17 to 5-18
 database login failure, 8-16
 definition, A-3
 editing across nodes, 4-10 to 4-11
 relationship with database, 5-19
 specifying in browser, 2-6
DBA

- default group created, 2-4
- diag script, 8-5
- diag.jar, 8-5
- Diagnose.class, 8-4
- diagnostics tool, 8-3 to 8-7
- directives
 - to support multiple aliases, 5-15
- distributed environments
 - architecture, 4-2
 - benefits, 4-3 to 4-4
 - privileges required to configure, 4-7
 - setting up, 4-7 to 4-18
- domains
 - DNS, 8-26
 - DNS in HOSTS file, A-7
 - resolving in HOST file, 5-15
- drbgdml.sql, 6-17

E

- errors
 - 400 bad request, 8-25
 - 503, 1-5
 - 503 Service Temporarily Unavailable, 8-23
 - Apache listener crashes, 8-21
 - Call to utl_http failed, 8-21
 - cannot find package DR_REWRITE, 6-3
 - creating interMedia Text indexes, 8-26
 - creating Login Server, 8-15
 - Database Login Failure, 8-16
 - database objects failing to create, 8-15
 - finding schema, 8-14
 - HTTP 400, 8-20
 - in Oracle Portal Configuration Assistant, 8-10, 8-13
 - interMedia Text is not installed, 6-12
 - Internal Server Error, 8-25
 - invalid Login Server packages, 8-16
 - Java Virtual Machine (JVM) is not installed, 8-10
 - JavaScript, 1-4
 - list of installation and configuration problems, 8-8
 - list of miscellaneous problems using Oracle Portal, 8-9

- list of problems logging on to Oracle Portal, 8-9
- list of problems running Oracle Portal, 8-9
- logging on to Oracle Portal, 8-18, 8-20
- NLS language, 8-29
- ORA and PLS, B-5
- ORA-01445, 7-10
- PLS-00306, 8-29
- PL/SQL Web Toolkit (OWA), 8-16
- Proxy Log On Failed, 8-19, A-4
- received when manually installing, B-5
- tablespaces, 8-13
- Timeout for content={0}, 8-22
- TNS could not resolve service name, 8-19
- troubleshooting, 8-1 to 8-29
- WWS-32100, 8-27

exporting, B-1

F

- FAST MODE
 - interMedia Text mode, 6-15
- firewalls, 5-10, 5-16, 5-18
- forms
 - adding rich content, 7-6
 - building rich content, 7-7
 - known issues, 7-9
 - object attributes, 7-3
- FULL MODE
 - interMedia Text mode, 6-15

G

- gateway.htm, 8-17
- groups
 - Oracle Portal defaults, 2-4
- guides, xiii
 - conventions, xv
 - Net8 Administrator's Guide, 6-20
 - structure, xiii

H

- Help
 - documentation, xvii
 - manually installing, B-9

HOSTS file
 definition, A-7
 localhost entry, 3-2
 location of, 8-7
 resolving domain names, 5-15
HTTP, 8-3
 400 error, 8-20
 default port 80, 8-19
http.conf, 5-20
 configuring virtual hosts, 5-7
httpd_error.log, 8-2
httpd.conf, 2-18, 5-14, 5-20, 8-18
 definition, A-2
 file location of, 8-7, A-2
HTTPS, 2-18, 2-20, 8-25
 using with certificates, 2-16

I

images
 adding interMedia rich content, 7-2
importing, B-1
init.ora, 6-3, 8-11, 8-13
 required settings, 1-5
installation
 Oracle Portal components, 1-1
install.log, 8-2
interMedia Text
 creating, 6-13 to 6-15
 dropping index, 6-17
 enabling, 6-11 to 6-13
 error creating, 8-26
 getting other information about, 7-3
 maintaining and synchronizing, 6-15
 non-support for rich content, 7-2
 prerequisites, 6-2
 searching in Oracle Portal, 6-6
 setting listner.ora, 6-18
 setting tnsnames.ora, 6-19
 setting up environment variables, 6-18 to 6-19
 system requirements, 1-5

J

Java, 8-29

 error when JVM option is not installed, 8-10
 SSOHash class, 8-13
JAVA_POOL_SIZE, 1-5, 8-11, 8-18
JavaScript, 1-4, 8-29
JServ, 1-3, 8-2, 8-3, 8-25, 8-26

K

KeepAlive, 8-24

L

langinst, 2-7, 2-8
languages
 error, 8-29
 multilexers in interMedia Text, 6-20
 supported in Oracle Portal, 2-7 to 2-9
laptop
 Oracle Portal configuration, 3-1 to 3-3
LD_LIBRARY_PATH, 6-3
 setting for interMedia Text, 1-6
LDAP, 5-6, 5-18
linstall, 3-7, 5-4, A-7
 file location of, 8-7
 syntax, B-10
listener crashes, 8-21
listener.ora, 6-18
Login Server
 account created, 2-3
 associating nodes to same, 4-11 to 4-14
 associating Oracle Portal to, 5-2 to 5-5
 centralized for Single Sign-On, 4-1
 common on all nodes, 4-6
 configured with Oracle Internet Directory (OID), 5-5
 configuring distinct instance, 3-6 to 3-8
 configuring separate listener, 5-1 to 5-5
 creating user with administrator privileges, 4-14
 deinstalling, 2-11
 error creating schema, 8-15
 installed schema, 2-2
 invalid package errors, 8-16
 location of WWSSO_PAPP_CONFIGURATION_INFO table, 8-7

- manually installing, B-10
- misconfiguration, 8-3
- Partner Application table, A-7
- privileges required to edit settings, 4-7
- same cookie domain required, 4-6
- verifying DAD, 8-2

logs

- Apache listener, 8-2
- indication of problems, 8-15
- installation, 8-2
- location of Apache, 8-2
- ssodatan.log, B-6
- LSNR_TOKEN, A-5, B-16

M

- MAX_ENABLED_ROLES, 1-5
- MaxClients, 5-20
- MaxRequestsPerChild, 8-24
- Metalink, 8-21
- Microsoft Internet Explorer, 1-4
 - browsers
 - configuring HTTPS in Internet Explorer, 2-24
- middle-tier
 - configuring load balancing, 5-16
 - Oracle9i Application Server, 3-1, 3-4, 3-5, 3-6
- MIME types, 7-6
 - support, 7-4
- MinSpareServers, 8-24
- mod_jserv, 5-13
- mod_plsql, 5-13, 8-2, 8-21, 8-23
- mod_ssl, 2-20
- multilexer
 - supported in interMedia Text, 6-20

N

- Navigator
 - using to browse rich content, 7-9
 - using to search pages and content areas, 6-4
- Net8
 - Proxy Log On Failed, 8-19
- Netscape, 1-4
- nodes
 - creating, 4-8

- creating additional, 4-17
- definition, 4-2
- discovering names, 4-14
- refreshing portlet repository, 4-16
- registering between themselves, 4-15 to 4-16
- requirements, 4-4
- symmetric registration, 4-6
- nosso, 4-8, B-5

O

- onetime, B-4
- operating systems
 - requirements, 1-3
- ORA-01445, 7-10
- Oracle HTTP Server powered by Apache
 - configuring separate listeners on Login Server and Oracle Portal, 5-1
 - on one node or on multiple nodes, 4-5
 - resolving frequent crashes, 8-21
- Oracle Internet Directory (OID), 8-28
 - configured in Login Server, 5-5
- Oracle JServer option
 - system requirement, 1-3
- Oracle Portal
 - accessing in browser, 2-5
 - changes in directory structure between releases, 8-8
 - configuring distinct instance, 3-6 to 3-8
 - configuring Global Settings, 6-10
 - configuring language support, 2-7 to 2-9
 - default home page, A-4
 - default schemas, 2-2
 - deinstalling, 2-10
 - documentation, xv
 - enabling beta features, 2-14
 - group defaults, 2-4
 - identifying component causing problem, 8-2
 - installed components, 1-1
 - Oracle8i database requirement, 1-3
 - Partner Applications, 3-6 to 3-8
 - ServerName distinction in different versions
 - of, 5-12
 - setting up distributed environments, 4-7 to 4-18
 - system requirements, 1-3

- troubleshooting, 8-1 to 8-29
- Oracle Portal Configuration Assistant
 - default tablespace, 1-4
 - errors, 8-10, 8-13
- Oracle Portal Development Kit (PDK), 4-7
- Oracle Technology Network, xvii, 4-7
- ORACLE_HOME
 - convention used in this guide, xv
 - locations of various Oracle HTTP Server files, 8-7
 - Oracle 8.1.6 client error, 8-28
 - requirements, 1-4
 - setting for laptop configuration, 3-3
- Oracle8i database
 - login error, 8-16
 - rich content stored in, 7-1
 - system requirement, 1-3
 - UTL_HTTP missing, 8-21
- Oracle9i Application Server
 - architecture, 1-2
 - configuration files, A-1 to A-7
 - configuration points and file locations, 8-7
 - installed components, 1-1

P

- Partner Applications
 - adding, 4-12, B-14
 - configuring distinct Oracle Portal and Login Server instances, 3-6 to 3-8
 - configuring Oracle Portal as, 5-1
 - creating, 5-3
 - in Login Server configuration table, A-5
 - success URL, A-6
- passwords
 - changing account, 2-4
- PLS-00306, 8-29
- PL/SQL
 - database requirements, 1-3
 - packages failing to be created, 8-16
 - Web Toolkit (OWA) errors, 8-16
- PL/SQL Gateway
 - 503 error, 8-23
 - location of middle-tier software, 3-6
- plugins, 7-7

- PORTAL_ADMINISTRATORS
 - default group created, 2-4
- PORTAL_DEVELOPERS
 - default group created, 2-4
- portal30
 - Oracle Portal default schema name, 2-2
- portal30_sso
 - Login Server default schema name, 2-2
- portlet providers
 - shared across nodes, 4-3
- portlet repository
 - refreshing, 4-16
- PORTLET_PUBLISHERS
 - default group created, 2-4
- ports
 - 443, 2-23
 - conflicting on Apache JSev, 8-2
 - dropping port 80, 8-19
 - HTTPS, 2-18
 - not including if using 80, B-13
 - used to access Oracle Portal, 2-5
- Proxy log on failed, 8-20
- proxy server, 5-11, 5-16, 6-11

R

- redirect
 - Apache listener, 8-24
 - simplifying Oracle Portal URL, 2-6
- remove, B-16
- reports
 - building with rich content, 7-4
 - display options, 7-3
 - known issues, 7-9
 - object attributes, 7-3
 - QBE parameter entry form, 7-6
 - wizards, 7-2
- routers
 - configuring load-balancing, 5-16

S

- sbrimtlx.sql, 6-20
- scalability
 - solutions in a distributed Oracle Portal

- environment, 4-3 to 4-4
- schemas
 - ctxsys, 6-20
 - database objects failing to create, 8-15
 - not created in portal, 8-15
 - Oracle Portal default, B-5
 - Oracle Portal defaults, 2-2
 - storing rich content, 7-1
- scripts
 - location of, B-2
- Secure Socket Layer (SSL)
 - 400 bad request error, 8-25
 - enabling, 2-20
- security
 - changing passwords, 2-4
- selfreg, 2-12
- self-registration, 2-12
- ServerName, 2-5, 3-2, 3-4, 5-7, 5-12, A-6, B-6
 - distinction in Oracle Portal versions, 5-12
- servers
 - integrated Oracle Portal
 - configuration, 3-3 to 3-4
 - proxy, 5-11
- sessions
 - cookie, B-17
 - determining number, 5-19
- Set MaxRequests, 8-24
- SHARED_POOL_SIZE, 1-5, 8-11, 8-17
- SID
 - interMedia Text, 6-18
- Single Sign-On (SSO)
 - configuring distributed Oracle Portal
 - environment, 4-1
 - failing, 4-6
 - installed component, 1-1
 - registering Login Server, 5-8
 - setting authentication mode, A-4
- snlpcgtsrvbynm, 8-28
- SQL, 7-2, 8-3
- SQL*Net, 8-3
- ssodatan, 2-18, 8-3, 8-18, B-6
 - file location of, 8-7
 - syntax, B-12
- ssodatan.log, B-6
- ssodatax, 3-7, 4-13, 5-3, 5-5, 5-8

- file location of, 8-7
 - syntax, B-14
- SSOHash, 8-13, 8-14, 8-18
- Success URL, 5-3
- SYS user, 1-3, 8-10
- system requirements, 1-3, 8-1
 - interMedia Text, 6-2

T

- tablespaces, 8-12
 - errors in, 8-13
 - increase for multilexers, 6-20
 - minimum requirements, 1-4
 - resizing tip, 8-12
- TCP/IP, 2-25, 3-2, 5-7, 5-16
- terminal
 - required settings, 1-5
- themes and gists
 - enabling for interMedia Text, 6-12
 - viewing, 6-7
- three-tier architecture
 - configuring in Oracle Portal, 3-4 to 3-5
- time out, 8-26
 - error, 8-22
 - KeepAlive parameter, 8-24
- tnsnames.ora, 6-19, 8-17, A-5
 - file location of, 8-7, A-2
- troubleshooting, 8-1 to 8-29
 - diagnostics tool, 8-3
 - identifying component causing the problem, 8-2
- tuning
 - Apache process configuration, 8-24

U

- UNIX
 - interMedia Text variables, 6-3
 - terminal settings, 1-5
- URL
 - Call to utl_http failed error, 8-21
 - excluding port 443, 2-24
 - invalid underscore character, 8-20
 - Oracle Portal home, 3-7
 - Partner Applications stored in Login

- Server, A-6
- simplifying Oracle Portal address, 2-6
- stored in partner enabler configuration table, 8-4
- useCanonicalName on, 5-13
- UTL_HTTP, 2-22, 8-21

- file location of, 8-7, A-2

V

- Verisign, 2-19
- videos
 - adding interMedia rich content, 7-2
- virtual hosts
 - configuring, 5-6 to 5-8
 - configuring HTTPS, 2-24

W

- wdbsvr.app, B-7
 - definition, A-3
 - file location of, 8-7, A-2
- WebDB 2.2, 8-29
- webdb30.www_utl_api_types, 8-29
- Windows NT
 - FAT file system, 1-4
- winstall
 - file location of, 8-7
 - steps to manually install Oracle Portal, B-3
 - syntax, B-4
- WWC-41439, 8-18
- wwctx_api_vpd, 2-12
- WWS-32100, 8-27
- wwsec_app_priv.process_signon, 3-7, 5-3, A-6
- wwsec_diagnostics, 8-4
- WWSEC_ENABLER_CONFIG_INFOS, 8-7, 8-14,
A-2, A-5
- wwsso_api_user_admin, 2-12
- WWSO_PAPP_CONFIGURATION_INFOS, A-2,
A-7, B-14
- wwwseedus.sql, 2-7

Z

- zone.properties, 2-16, 8-22, 8-25, 8-26
 - definition, A-2

