

Oracle Application Server 10g R2: Administration I

Student Guide • Volume 1

D16508GC21

Edition 2.1

November 2005

D22658

ORACLE®

Author

Saurabh Banerjee

Editors

Aju Kumar
Navratan Singh

Technical Contributors and Reviewers

Martin Alvarez	Bruce Lowenthal
Nick Angelis	Russ Lowenthal
Celia Antonio	Ramaa Mani
Don Bates	Michael Mesaros
Gaurav Bhatia	Priya Nathan
Martijn Van Der Bruggen	Paul Needham
Paul Burgess	Vishal Parashar
Vince Casarez	Nagavalli Pataballa
Fermin Castro	Chirag Patel
Greg Cook	Bharat Peramur
Ellen Desmond	William (Cas) Prewitt
Lypp-Tek Khoo-Ellis	Srinivas Putrevu
Larry Frazier	Sandhya Rajput
Viresh Garg	Shankar Raman
Philip Garm	Holger Dindler Rasmussen
Usha George	Vasudeva Rayakuntapalle
Helen Grembowicz	Joe Roch
Ric Goell	Mary Beth Roeser
Shalendra Goel	Wes Root
Ellen Gravina	Shaibal Saha
Helen Grembowicz	David Saslav
Nicole Haba	Nachiketa Shukla
Kurt Heiss	Mohit Singh
Taj-ul Islam	Navneet Singh
Vijay Jayasheelan	Uma Sivakumar
Julia Johnson	Richard Leslie Smith
Tim m Kelly	Jayanthan Thomas
Teria Kidd	Dr Volker Zell
Nirguna Kota	Rob Zic
Mathias Kullberg	portaled_us
John Lang	portalpm_us
Peter Laquerre	svrcurr_us
	as10inst_ww

Publisher

Sujatha Nagendra

Copyright © 2005, Oracle. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Contents

Preface

1 Introduction

- Objectives 1-2
- Course Objectives 1-3
- Course Units 1-4
- Unit 1: Product Overview 1-5
- Unit 2: Installation 1-6
- Unit 3: Basic Management and Configuration 1-7
- Basic Management and Configuration 1-8
- Unit 4: Application Deployment 1-9
- Unit 5: Managing Access Control 1-10
- Unit 6: Availability 1-12
- Summary 1-13

2 Oracle Application Server 10g: Key Components and Features

- Objectives 2-2
- Challenges of Creating and Maintaining E-Business Applications 2-3
- Oracle Application Server: Overview 2-4
- Multitiered Model 2-7
- Oracle Application Server 10g Architecture 2-8
- Oracle Application Server Terminology 2-9
- Oracle Application Server Components and Solutions 2-11
- Oracle Application Server Integration Solutions 2-14
- Oracle Application Server Products 2-16
- Oracle Application Server Installation Types 2-17
- Oracle Application Server Management 2-18
- OracleAS Infrastructure 2-20
- OracleAS Infrastructure Components 2-21
- Oracle Internet Directory and Security 2-23
- Securing the Web Infrastructure 2-24
- OracleAS Infrastructure Installation Types 2-25
- Installation Types That Require Infrastructure 2-26
- Services and Component Matrix for OracleAS Infrastructure 2-27
- OracleAS Middle Tier Components 2-28
- Oracle HTTP Server 2-30
- OracleAS Web Cache 2-31
- Enhancing Performance with Caching 2-33
- OracleAS Containers for J2EE (OC4J) 2-34
- OracleAS Web Services 2-35
- OracleAS Enterprise Portal 2-36
- Wireless-Enabled Applications 2-38
- Mobile Portal Architecture 2-40
- OracleAS Developer Kits 2-41
- Summary 2-42

3 Installing OracleAS Infrastructure

- Objectives 3-2
- OracleAS Infrastructure Components 3-3
- Order of Installing OracleAS Infrastructure Components 3-5
- OracleAS Infrastructure Installation: Overview 3-6
- Minimum Requirements for OracleAS Infrastructure 3-7
- Setting Up the Environment 3-9
- OracleAS Infrastructure: Installation Steps 3-11
- Starting the Installation 3-13
- Oracle Universal Installer 3-14
- First Installation of an Oracle Product 3-15
- Specify File Locations Window 3-16
- Select a Product to Install 3-17
- Select Installation Type 3-18
- Select Configuration Options 3-20
- Specifying Port Configuration Options 3-22
- Specify Namespace in Internet Directory 3-23
- Oracle Application Server Certificate Authority 3-24
- Specify Database Configuration Options 3-25
- Specify Database Schema Passwords 3-26
- Specify Instance Details 3-27
- Summary of Installation 3-28
- End of Installation Window 3-29
- Postinstallation Tasks 3-30
- Accessing the OracleAS Infrastructure Instance 3-31
- Application Server Control 3-32
- Verifying Oracle Internet Directory Server 3-33
- Oracle Enterprise Manager 10g Database Control 3-34
- Managing OracleAS Metadata Repository Database 3-35
- Accessing the SSO Server 3-36
- Starting and Stopping OracleAS Infrastructure 3-37
- Summary 3-38

4 Installing OracleAS Middle Tier

- Objectives 4-2
- OracleAS Middle Tier Installation Phases: Overview 4-3
- Preinstallation: OracleAS Middle Tier Requirements 4-4
- Preinstallation: Setting Up the Environment 4-5
- Installation: Starting the Installer 4-6
- Specifying File Locations 4-7
- Selecting a Product 4-8
- Selecting an Installation Type 4-9
- Middle-Tier Installation Options 4-10

Installer: Selecting Component Configuration 4-12
Selecting Configuration Options 4-13
Selecting Repository Type 4-14
Specifying Metadata Repository for Database-Managed Farm 4-15
Selecting File-Based Farm Repository 4-16
Specifying Port Configuration Options 4-17
Registering with Oracle Internet Directory 4-18
Using Metadata Repository 4-19
Instance Name and the `ias_admin` Password 4-20
Installer: Summary 4-21
Installer: End of Installation 4-22
Accessing Application Server Control 4-23
Accessing the Welcome Page 4-24
OracleAS Portal 10.1.4: New Features 4-25
Upgrading OracleAS Portal Middle Tier from 10.1.2.0.2 to 10.1.4 4-27
Preinstallation: Setting Up the Environment 4-28
Performing the Upgrade 4-30
Postinstallation: Setting Up the Environment 4-31
Accessing the OracleAS Portal Welcome Page 4-32
Summary 4-33

5 Using Oracle Application Server Management Tools

Objectives 5-2
Management Controls 5-4
Comparing System Management Tools 5-5
System Management Tools 5-6
Grid Control: Overview 5-7
Grid Control Architecture 5-8
Application Server Control: Overview 5-10
Oracle Application Server: Overview 5-11
Application Server Control Architecture 5-12
Application Server Control 5-14
Using Application Server Control 5-15
Application Server Control Pages 5-16
OracleAS Farm Page 5-17
Topology Viewer: Overview 5-18
Accessing the Topology Viewer 5-20
Benefits of Using the Topology Viewer 5-21
Navigating Around the Topology 5-22
Oracle Application Server Instance Home Page 5-24
Starting, Stopping, and Restarting Oracle Application Server Instances 5-25
Oracle Application Server Component Home Pages 5-26
Starting, Stopping, and Restarting the Components 5-27

Obtaining Common Metrics About Oracle Application Server 5-28
Obtaining Information About the Host Computer 5-29
Oracle Application Server Host Home Page 5-30
Viewing Performance Metric Details 5-31
Application Server Ports Page 5-33
Storing Log Information in the Diagnostic Message Database Repository 5-34
Log Viewer 5-36
emctl Utility 5-37
Enabling SSL for Application Server Control 5-38
Changing Oracle Enterprise Manager 10g Port Values 5-40
Oracle Process Manager and Notification Server (OPMN) 5-42
opmnctl Command 5-43
Typical Startup Sequence 5-45
Typical Shutdown Sequence 5-47
Distributed Configuration Management 5-48
DCM and Metadata Repository 5-49
Using dcmctl 5-50
Using dcmctl in Batch Mode 5-51
Management Tasks: Tools 5-52
Summary 5-53

6 Configuring and Managing Oracle HTTP Server

Objectives 6-2
Introduction to Oracle HTTP Server 6-3
Oracle HTTP Server Modules 6-4
Oracle HTTP Server Processing Model 6-6
Managing Processes and Connections 6-7
Controlling the Number of Processes and Connections 6-8
Starting, Stopping, and Restarting Oracle HTTP Server 6-9
Starting and Stopping Oracle HTTP Server Manually 6-10
Directory Structure 6-11
Oracle HTTP Server Configuration Files 6-12
Specifying File Locations 6-14
Oracle HTTP Server Home Page 6-16
Configuring Oracle HTTP Server 6-17
Controlling Access to the Application Server 6-18
Setting Server and Administrator Functions 6-19
Modifying Server Properties 6-20
Specifying a Listener Port 6-21
Administrative Directives 6-22
Server Logs 6-23
Configuring and Using Server Logs 6-24

LogLevel Directive 6-25
Log Formats 6-26
Changing Error Log Properties 6-27
Adding an Access Log File 6-28
Log Rotation 6-29
Managing Client Requests and Connection Handling 6-31
Advanced Server Properties 6-32
Editing Server Configuration Files 6-33
Getting the Server Status 6-34
Monitoring Oracle HTTP Server 6-35
Using Stand-Alone Oracle HTTP Server Based on Apache 2.0 6-36
What Is mod_security? 6-38
Installing and Configuring mod_security 6-39
Summary 6-42

7 Configuring Directives and Virtual Hosts

Objectives 7-2
Configuration Contexts 7-3
Container Directives 7-5
Block Directives 7-6
Specifying the Location of the Directives 7-7
Where the Directives Can Be Specified 7-8
<Directory> Directive 7-9
<Files> and <Location> 7-10
Container Directives: <Files> and <Location> 7-11
Defining Virtual Hosts 7-12
Using IP-Based Virtual Hosts 7-14
Using Name-Based Virtual Hosts 7-15
Configuring Virtual Hosts 7-16
Controlling Allowed Features 7-18
Options Parameters 7-19
Using Options 7-21
Overriding Directives with the Per-Directory Configuration 7-22
Controlling Overrides with AllowOverride 7-23
Directory Indexing 7-24
DirectoryIndex Directive 7-25
Controlling Directory Listings with IndexIgnore 7-26
Error and Response Handling 7-27
Expires Header 7-29
Alias, AliasMatch, and ScriptAlias 7-31
Summary 7-32

8 Configuring and Managing OracleAS Web Cache

Objectives 8-2

What Is OracleAS Web Cache? 8-3

How Does OracleAS Web Cache Work? 8-5

OracleAS Web Cache Concepts 8-6

Administering OracleAS Web Cache 8-8

Managing OracleAS Web Cache with Application Server Control Console 8-10

Using Application Server Control to Start and Stop OracleAS Web Cache 8-11

Using `opmnctl` to Start and Stop OracleAS Web Cache 8-12

OracleAS Web Cache Home Page 8-13

Application Server Control Console: Web Cache Performance Page 8-15

Modifying Security Settings 8-16

Configuring Listening Ports for Requests 8-18

Changing Operations Ports 8-19

Specifying Origin Server Settings 8-20

Site Definitions 8-22

Configuring Site Definitions and Mapping to the Origin Server 8-23

Configuring Site Definitions 8-24

Caching Rules: Overview 8-25

Surrogate-Capability 8-26

Predefined Caching Rules 8-28

Rules for Caching, Personalization, and Compression 8-29

Streamed Delivery of Compressed Content 8-30

Streaming of Compressed Content by OracleAS Web Cache 8-31

Creating Caching Rules 8-32

Edit Caching Rules 8-34

Caching Dynamic and Partial Pages 8-36

Expiration Rules 8-38

Defining Expiration Rules 8-39

Performance Assurance and Surge Protection 8-40

Invalidation Messages 8-41

Basic Content Invalidation 8-43

Logging Events and Accessing Information 8-45

Configuring Access Log 8-46

Configuring Event Log 8-48

Event Log with Startup Entries: Example 8-50

Configuring Rollover Frequency 8-51

Manual Rollover of Logs 8-52

Summary 8-53

9 Configuring and Managing OC4J

Objectives 9-2

Java 2 Platform, Enterprise Edition (J2EE): Overview 9-3

J2EE Platform 9-4
Client-Tier Components 9-5
Web-Tier Components 9-6
Business-Tier Components 9-7
Enterprise JavaBeans (EJB) 9-8
Comparing JAR, WAR, and EAR Files 9-9
Compare JAR, WAR, and EAR Files 9-10
Oracle Application Server Containers for J2EE (OC4J) 9-11
Managing OC4J 9-12
Creating an OC4J Instance 9-13
Creating an OC4J Instance Using `dcmctl` 9-14
Application Server Control: OC4J Home Page 9-15
Starting and Stopping OC4J Instance 9-16
Starting and Stopping OC4J Instances Using OPMN 9-17
Disabling OC4J Instances 9-18
Enabling OC4J Instances 9-19
OC4J Configuration Basics 9-20
OC4J Instance Configuration Files 9-21
Relationship of Configuration Files 9-22
Sample `server.xml` File 9-23
Sample `default-web-site.xml` File 9-24
Configuring OC4J Using Application Server Control 9-25
Server Properties Page: General Section 9-26
Web Site Properties 9-27
JSP Properties 9-28
Advanced Properties 9-30
Application Deployment 9-31
OC4J Applications Page 9-32
Maintaining Applications 9-33
Maintaining Web Modules 9-34
Summary 9-35

10 Deploying Java 2, Enterprise Edition (J2EE) Applications

Objectives 10-2
J2EE Architecture 10-3
Databases and J2EE 10-4
Enterprise JavaBeans 10-5
EJB Structure 10-6
EJB and OC4J 10-7
EJB Module 10-8
Deploying Web Application Modules Using Application Server Control 10-9
Deploying Web Application Modules Using `dcmctl` 10-11

Data Sources and the Deployer Role 10-12
Specifying Data Sources 10-13
Obtaining Data Source Information 10-14
Sample `data-sources.xml` File 10-16
Creating a Data Source: General 10-17
Creating a Data Source: Username and Password 10-18
Creating a Data Source: JNDI Locations 10-19
Creating a Data Source: Connection Attributes and Properties 10-20
Specifying CMP Data Source 10-21
Binding EJBs to Existing Tables 10-22
Deploying J2EE Applications Using Application Server Control 10-23
Monitoring J2EE Applications 10-27
Deploying J2EE Applications Using `dcmctl` 10-29
Summary 10-30

11 Oracle Application Server Security Services

Objectives 11-2
Security Risks in an Internet Environment 11-3
Security Services in an Internet Environment 11-5
Addressing the Security Challenges 11-6
Oracle Application Server Security Architecture 11-7
What Is SSL? 11-9
Private and Public Key Cryptography 11-11
Public Key Infrastructure (PKI) 11-13
Public Key Infrastructure 11-14
Storing Secure Credentials 11-15
OracleAS Web Cache Security 11-16
Oracle HTTP Server Security 11-17
J2EE Security and JAAS 11-18
Oracle Identity Management Security Solution 11-21
OracleAS Portal Security Architecture 11-23
Summary 11-25

12 Configuring Oracle Application Server Components in Oracle Internet Directory

Objectives 12-2
Identity Management: Overview 12-3
Benefits of Identity Management 12-4
Oracle Identity Management 12-5
Oracle Internet Directory 12-7
Security Benefits of Oracle Internet Directory 12-8
Oracle Application Server Components and Oracle Internet Directory 12-9
Default Schema and Directory Information Tree (DIT) 12-11
Identity Management Realm-Specific Common Entries 12-13

Default Identity Management Realm Configuration 12-15
Starting and Stopping by Using OPMN 12-16
Oracle Internet Directory Server Processes 12-18
Starting and Stopping the `oidmon` Process 12-19
Starting and Stopping an Oracle Internet Directory Server Instance 12-21
Starting Oracle Internet Directory Server Instance 12-22
Using Bulk Tools 12-23
Using LDAP Command-Line Tools 12-24
Using Oracle Directory Manager 12-25
Connecting to the Oracle Internet Directory Server 12-26
Oracle Directory Manager Connect Dialog Box 12-27
ODM Connect Dialog Box 12-28
Disconnecting from the Oracle Internet Directory Server 12-29
Oracle Application Server Bootstrap Model 12-30
Oracle Internet Directory Administration Delegation Flow 12-31
Delegated Directory Administration 12-32
Directory Roles 12-33
Oracle Application Server Administration Model 12-34
User Administration 12-36
Group Administration 12-37
Administrative Groups 12-38
Storage of User Credentials 12-40
Modifying Password Policies by Using ODM 12-42
Modifying the Oracle Internet Directory Administrator Password 12-44
Modifying the Realm-Specific Administrator Password 12-45
Modifying the Administrator Password 12-46
Relationship Between OracleAS Portal and Oracle Internet Directory 12-47
OracleAS Portal Directory Entries in Oracle Internet Directory 12-48
Configuring Oracle Internet Directory Settings in OracleAS Portal 12-50
Caching Oracle Internet Directory Information in OracleAS Portal 12-51
Synchronizing Cached Oracle Internet Directory Information in OracleAS Portal 12-52
Enabling Directory Synchronization in the OracleAS Portal Instance 12-53
Summary 12-54

13 Managing the OracleAS Portal

Objectives 13-2
OracleAS Portal Administrative Services: Overview 13-3
Managing the OracleAS Portal Instance by Using Application Server Control 13-4
OracleAS Portal Instance Home Page 13-5
Monitoring the OracleAS Portal Instance 13-6
Managing the OracleAS Portal Instance by Using Administrative Portlets 13-8
Default Portal Users 13-9
Default Portal Groups 13-10

OracleAS Portal Schemas	13-12
Managing Passwords for the OracleAS Portal Schemas	13-13
Managing Portal Users and Groups	13-14
Creating Portal Users	13-15
Editing Portal User Profiles	13-16
Mapping Portal Users to a Custom OracleAS Portal Access Schema	13-18
Creating Portal Groups	13-19
Editing Portal Group Profiles	13-20
Assigning Privileges to OracleAS Portal Users and Groups	13-21
User and Group Lists of Values	13-22
What Is Portlet Repository?	13-23
Accessing the Portlet Repository	13-24
Managing the Portlet Repository	13-25
Refreshing the Portlet Repository	13-27
Displaying the Portlet Repository Page Group	13-28
Organizing the Portlet Repository Page Group	13-29
Securing the Portlet Repository Page Group	13-30
Registering a Provider	13-31
Updating the Provider Registration Information	13-32
Database Providers and PL/SQL Portlets	13-34
Installing the Database Provider and Its PL/SQL Portlets	13-35
Registering the Database Provider with OracleAS Portal	13-36
Using a WSRP Provider	13-38
Registering a WSRP Provider	13-39
What are Web Providers?	13-41
Accessing Web Providers	13-42
Testing Web Providers	13-43
Registering Web Providers: Provider Information	13-44
Adding the Portlet to a Portal Page	13-45
Exporting and Importing Objects in OracleAS Portal	13-46
Creating a New Transport Set	13-48
Editing a Saved Transport Set	13-49
Exporting a Transport Set	13-50
Importing the Transport Set	13-52
Browsing Transport Sets	13-54
Summary	13-55

14 Configuring OracleAS Portal

Objectives	14-2
OracleAS Portal Configuration Tasks: Overview	14-3
Self-Registration Feature in OracleAS Portal	14-4
Configuring the Self-Registration Feature in OracleAS Portal	14-5
Enabling the Self-Registration Feature in the Login Portlet	14-6

OraDAV Architecture 14-7
Configuring OraDAV Support for OracleAS Portal Access 14-8
Configuring Language Support 14-9
Setting Language for a Portal Session 14-11
Configuring OracleAS Portal Dependencies 14-12
Portal Dependency Settings File 14-13
The Portal Dependency Settings File 14-14
Portal Dependency Settings Tool 14-15
ptlconfig Modes 14-16
Configuring Portal Web Cache Settings 14-18
Configuring Virtual Hosts 14-19
Configuring Multiple Middle Tiers 14-22
Configuring a Dedicated OracleAS Web Cache 14-24
Configuring OracleAS Portal Search Portlets 14-26
Configuring OracleAS Portal Search Options 14-27
Administering Web Clipping 14-28
Administering OmniPortlet 14-32
Summary 14-34

15 Administering the OracleAS Single Sign-On Server

Objectives 15-2
OracleAS Single Sign-On Server: Overview 15-3
Single Sign-On Components 15-4
Authentication Flow for OracleAS Single Sign-On 15-6
Starting and Stopping OracleAS Single Sign-On Components 15-8
OracleAS Single Sign-On Administrator's Role 15-10
OracleAS Single Sign-On Administration Pages 15-12
Configuring the OracleAS Single Sign-On Server 15-13
Partner Application: Overview 15-14
Registering mod_osso 15-15
Creating and Editing a Partner Application 15-17
Administering External Applications 15-20
Adding External Applications 15-21
Adding an External Application 15-22
Accessing an External Application and Storing Its Credentials 15-23
Monitoring the OracleAS Single Sign-On Server 15-24
Accessing SSO Server from OracleAS Portal 15-26
Accessing External Applications from OracleAS Portal 15-27
Summary 15-28

16 Managing Access Using Oracle Delegated Administration Services

Objectives 16-2
Oracle Delegated Administration Services 16-4
Benefits of Oracle Internet Directory Self-Service Console 16-6

How Oracle Delegated Administration Services Works 16-7
How DAS Works 16-8
Oracle Delegated Administration Services Proxy User 16-9
Starting and Stopping 16-10
Verifying That Oracle Delegated Administration Services Is Running 16-11
Configuring the Default Identity Management Realm–Specific Context 16-13
Configuring User Entries 16-15
Managing Users, Groups, and Subscribers 16-17
Searching for User and Group Entries 16-18
Maintaining User Entries 16-19
Changing Passwords 16-20
Changing Another User's Password 16-21
Creating Group Entries 16-22
Modifying and Deleting Group Entries 16-24
Assigning Privileges to Users and Groups 16-25
Managing Services 16-26
Managing Accounts 16-27
Creating Identity Management Realms 16-28
Accessing Oracle Delegated Administration Services from OracleAS Portal 16-30
Granting Privileges to OracleAS Portal Users by Using Roles 16-31
Disabling the Privilege Assignment Section 16-33
Summary 16-34

17 Managing and Configuring OracleAS Certificate Authority

Objectives 17-2
OracleAS PKI Components 17-3
How SSL Works 17-5
OracleAS Certificate Authority 17-7
Oracle Application Server Certificate Provisioning 17-9
OCA Functional Structure 17-10
OCA Single Sign-On Authentication 17-12
OCA Configuration Elements 17-13
Starting and Stopping OCA 17-15
Accessing the OCA Interface 17-16
Details Required to Obtain a Certificate 17-17
Requesting the Web Administrator Certificate 17-18
Managing Certificates 17-19
Viewing, Approving, and Rejecting Certificates 17-20
Viewing / Approving / Rejecting Certificates 17-21
Revoking and Renewing Certificates 17-22
Updating the Certificate Revocation List (CRL) 17-23
Configuring OCA Server 17-25
Summary 17-26

18 Securing OracleAS Components by Using SSL

- Objectives 18-2
- Accessing the End-User Interface 18-3
- User Certificates 18-4
- Single Sign-On Authentication 18-5
- Requesting Other User Certificates 18-7
- Managing User Certificates 18-8
- Obtaining a Server Certificate 18-9
- What Is Oracle Wallet Manager? 18-10
- OWM Functions 18-11
- Creating a New Wallet 18-12
- Managing User Certificates 18-13
- Adding a Certificate Request 18-14
- Exporting a User Certificate Request 18-15
- Importing the User Certificate to the Wallet 18-16
- Managing Trusted Certificates 18-17
- Importing/Exporting a Trusted Certificate 18-18
- Exporting a Wallet 18-19
- Uploading Wallets 18-20
- Downloading Wallets 18-22
- Requesting a Server Certificate 18-23
- Requesting a Subordinate CA Certificate 18-24
- Importing and Downloading a CRL 18-26
- Configuring Browser to Trust OCA 18-27
- Enabling Oracle HTTP Server to Use SSL 18-29
- SSL Configuration Tool 18-30
- Configuring Oracle HTTP Server for SSL Certificates 18-32
- Adding User Certificates to Oracle Internet Directory 18-34
- Configuring OracleAS Web Cache to Use SSL 18-36
- Securing OracleAS Portal 18-37
- Securing the Parallel Page Engine 18-38
- Associating OracleAS Portal with OracleAS SSO in SSL Mode 18-39
- Associating OracleAS Portal with OracleAS SSO 18-40
- Securing Calls to DAS from OracleAS Portal 18-41
- Summary 18-42

19 Backing Up and Restoring Oracle Application Server

- Objectives 19-2
- Backup and Recovery Features 19-3
- Roadmap for Backup and Recovery 19-5
- Concept of Oracle Application Server Backup and Recovery 19-6
- Philosophy of Oracle Application Server Backup and Recovery 19-7
- Terminology 19-8

Performing a Complete Oracle Application Server Environment Backup	19-10
Performing Online Backups	19-11
Performing a Backup After a Major Change	19-12
OracleAS Backup and Recovery Tool	19-13
Preparing to Configure the Tool	19-14
Configuring the OracleAS Backup and Recovery Tool	19-16
Using <code>bkp_restore.sh</code>	19-19
Using <code>bkp_restore.sh</code> : Examples	19-20
Using Oracle Application Server Control for Backup and Recovery	19-23
Configuring Backup/Recovery Settings	19-24
Backup Procedures	19-25
Creating a Record of the Configuration	19-26
Performing a Complete Backup	19-27
Step 1: Shut Down the Oracle Application Server Environment	19-28
Step 2: Back Up the Infrastructure	19-30
Step 3: Back Up the Middle Tiers	19-32
Step 4: Back Up Your Oracle System Files	19-33
Step 5: Start Your Oracle Application Server Environment	19-34
Restore Procedures	19-36
Restoring OracleAS Infrastructure to a New Host	19-38
Restoring OracleAS Infrastructure Configuration Files	19-41
Restoring Middle Tier to the Same Host	19-42
Restoring Middle Tier to a New Host	19-43
Restoring Middle-Tier Configuration Files	19-46
Summary	19-47

Appendix A: Practices

Appendix B: Solutions

Appendix C: Configuring `mod_rewrite`

Appendix D: Deploying PL/SQL Applications

Appendix E: Introduction to Linux

Appendix F: Introduction to OracleAS Portal 10.1.4

Index

Preface

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Profile

Before You Begin This Course

Before you begin this course, you should have the following qualifications:

- Working experience with LINUX operating system

How This Course Is Organized

Oracle Application Server 10g R2: Administration I is an instructor-led course featuring lecture and hands-on exercises. Online demonstrations and written practice sessions reinforce the concepts and skills introduced in this course.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Related Publications

Oracle Publications

Title	Part Number
<i>Oracle Application Server Installation Guide 10g Release 2 for Linux</i>	<i>B19310-03</i>
<i>Oracle Application Server Release Notes 10g Release 2 (10.1.2) for Linux x86</i>	<i>B19312-03</i>
<i>Oracle Application Server Concepts 10g Release 2 (10.1.2)</i>	<i>B13994-02</i>
<i>Oracle Application Server Quick Tour 10g Release 2 (10.1.2)</i>	<i>B13993-01</i>
<i>Oracle Application Server Administrator's Guide 10g Release 2 (10.1.2)</i>	<i>B13995-05</i>
<i>Oracle Application Server Security Guide 10g Release 2 (10.1.2)</i>	<i>B13999-03</i>
<i>Oracle Application Server Performance Guide 10g Release 2 (10.1.2)</i>	<i>B14001-02</i>
<i>Oracle HTTP Server Administrator's Guide 10g Release 2 (10.1.2)</i>	<i>B14007-03</i>
<i>Oracle Application Server Containers for J2EE User's Guide 10g Release 2 (10.1.2)</i>	<i>B14011-02</i>
<i>Oracle Application Server mod_plsql User's Guide 10g Release 2 (10.1.2)</i>	<i>B14010-02</i>
<i>Oracle Application Server Portal Configuration Guide 10g Release 2 (10.1.4)</i>	<i>B19305-02</i>
<i>Oracle Application Server Web Cache Administrator's Guide 10g Release 2 (10.1.2)</i>	<i>B14046-03</i>
<i>Oracle Enterprise Manager Concepts 10g Release 1 (10.1)</i>	<i>B12016-02</i>
<i>Oracle Application Server Single Sign-On Administrator's Guide 10g Release 2 (10.1.2)</i>	<i>B14078-02</i>
<i>Oracle Application Server Certificate Authority Administrator's Guide 10g Release 2 (10.1.2)</i>	<i>B14080-02</i>
<i>Oracle Internet Directory Administrator's Guide, 10g Release 2 (10.1.2)</i>	<i>B14082-02</i>
<i>Oracle Identity Management Concepts and Deployment Planning Guide 10g Release 2 (10.1.2)</i>	<i>B14084-02</i>

Additional Publications

- System release bulletins
- Installation and user's guides
- Read-me files
- International Oracle User's Group (IOUG) articles
- *Oracle Magazine*

Typographic Conventions

Typographic Conventions In Text

Convention	Element	Example
Bold	Emphasized words and phrases in Web content only	To navigate within this application, do not click the Back and Forward buttons.
Bold italic	Glossary terms (if there is a glossary)	The <i>algorithm</i> inserts the new key.
Brackets	Key names	Press [Enter].
Caps and lowercase	Buttons, check boxes, triggers, windows	Click the Executable button. Select the Registration Required check box. Assign a When-Validate-Item trigger. Open the Master Schedule window.
Carets	Menu paths	Select File > Save.
Commas	Key sequences	Press and release these keys one at a time: [Alt], [F], [D]

Typographic Conventions (continued)

Typographic Conventions In Text (continued)

Convention	Object or Term	Example
Courier New, case sensitive	Code output, SQL and PL/SQL code elements, Java code elements, directory names, filenames, passwords, pathnames, URLs, user input, usernames	Code output: <code>debug.seti('I',300);</code> SQL code elements: Use the <code>SELECT</code> command to view information stored in the <code>last_name</code> column of the <code>emp</code> table. Java code elements: Java programming involves the <code>String</code> and <code>StringBuffer</code> classes. Directory names: <code>bin</code> (DOS), <code>\$FMHOME</code> (UNIX) Filenames: Locate the <code>init.ora</code> file. Passwords: Use <code>tiger</code> as your password. Pathnames: Open <code>c:\my_docs\projects</code> . URLs: Go to <code>http://www.oracle.com</code> . User input: Enter <code>300</code> . Usernames: Log on as <code>scott</code> .
Initial cap	Graphics labels (unless the term is a proper noun)	Customer address (<i>but</i> Oracle Payables)
Italic	Emphasized words and phrases in print publications, titles of books and courses, variables	Do <i>not</i> save changes to the database. For further information, see <i>Oracle7 Server SQL Language Reference Manual</i> . Enter <u><i>user_id@us.oracle.com</i></u> , where <i>user_id</i> is the name of the user.
Plus signs	Key combinations	Press and hold these keys simultaneously: [Control] + [Alt] + [Delete]
Quotation marks	Lesson and chapter titles in cross references, interface elements with long names that have only initial caps	This subject is covered in Unit II, Lesson 3, “Working with Objects.” Select the “Include a reusable module component” and click Finish. Use the “WHERE clause of query” property.

Typographic Conventions (continued)

Typographic Conventions in Navigation Paths

This course uses simplified navigation paths, such as the following example, to direct you through Oracle Applications.

Example:

Invoice Batch Summary

(N) Invoice > Entry > Invoice Batches Summary (M) Query > Find (B) Approve

This simplified path translates to the following:

1. (N) From the Navigator window, select Invoice > Entry > Invoice Batches Summary.
2. (M) From the menu, select Query > Find.
3. (B) Click the Approve button.

Notation:

(N) = Navigator	(I) = Icon
(M) = Menu	(H) = Hyperlink
(T) = Tab	(B) = Button

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

1

Introduction

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this course, you should be able to do the following:

- **Describe the role of a Web administrator**
- **Describe the architecture and components of Oracle Application Server**
- **Install OracleAS Infrastructure and OracleAS Middle Tier**
- **Configure and manage OracleAS Middle Tier components**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Course Objectives

This course describes how to perform Oracle Application Server administration tasks, such as:

- Installing and configuring Oracle Application Server components
- Deploying applications
- Implementing access control and security
- Monitoring the performance and availability of Oracle Application Server and the deployed applications

Course Objectives

- **Configure and manage OracleAS Infrastructure components, such as:**
 - Oracle Internet Directory
 - OracleAS Single Sign-On
- **Manage and configure Oracle Application Server Certificate Authority**
- **Deploy and manage Web applications**
- **Describe backup and recovery solutions for OracleAS Infrastructure and OracleAS Middle Tier**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Course Units

This course is divided into the following units:

- 1. Product Overview**
- 2. Installation**
- 3. Basic Configuration and Management**
- 4. Application Deployment**
- 5. Managing Access Control**
- 6. Performance and Availability**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Course Units

This course is divided into six units. Each unit addresses a major task that an Oracle Application Server administrator is expected to perform.

The tasks are related to the administration of the default installation of OracleAS Infrastructure and OracleAS Middle Tier with the Portal and Wireless installation type.

The units are described on the following pages.

Unit 1: Product Overview

This unit covers the following lesson:

- **Oracle Application Server: Key Components and Features**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Unit 1: Product Overview

You learn about the Oracle Application Server solution areas and product components, and the installation type that is necessary for your business goals. The following are the key solution areas addressed by Oracle Application Server:

- Deploying and managing J2EE applications
- Deploying and managing portals and wireless-enabled applications
- Accelerating performance with caching
- Managing and securing the Web infrastructure

Unit 2: Installation

This unit covers the following lessons:

- **Installing OracleAS Infrastructure**
- **Installing OracleAS Middle Tier (Portal and Wireless)**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Unit 2: Installation

Each installation of Oracle Application Server depends on the options chosen at the time of installation; however, there are common elements to each installation. You learn about:

- The components that form the core of the Oracle Application Server architecture and are common to most installations
- The components that enhance the Oracle Application Server architecture
- The request and the communication flow involved in providing services to clients

The middle tier contains software that enables you to deliver Web content, host Web applications, connect to back-office applications, and access your data on wireless devices. You learn how to choose the installation option that is appropriate for your needs, perform the installation tasks, and verify whether the installation is successful.

OracleAS Infrastructure is a prerequisite for many middle-tier installations. The components in OracleAS Infrastructure act as service providers for the middle tier. You learn how to configure and enable these components to best suit the middle-tier architecture in your environment.

Unit 3: Basic Management and Configuration

This unit covers the following lessons:

- **Using Oracle Application Server Management Tools**
- **Configuring and Managing Oracle HTTP Server**
- **Configuring and Managing OracleAS Web Cache**
- **Configuring and Managing OC4J**
- **Configuring OracleAS Portal**
- **Managing OracleAS Portal**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Unit 3: Basic Management and Configuration

Oracle Application Server provides flexibility in managing your Oracle Application Server environment. You learn how to use the Web-based Application Server Control to manage Oracle Application Server instances. You also learn to perform basic management tasks, such as starting and stopping OracleAS Infrastructure and middle-tier components by using command-line interfaces.

Oracle HTTP Server is a core component of Oracle Application Server. OracleAS Web Cache accelerates static and dynamic content delivery. You learn:

- About the configuration of Oracle HTTP Server, and how to start and stop Oracle HTTP Server
- To configure WebDAV support in Oracle HTTP Server for OracleAS Portal access
- To use Application Server Control to start, stop, restart, and configure OracleAS Web Cache, and also to obtain status information
- To create, modify, and delete caching rules and to apply invalidation mechanisms

Unit 3: Basic Management and Configuration (continued)

Oracle Application Server Containers for J2EE (OC4J) is the basis for all the J2EE services that are provided by Oracle Application Server. You learn about the architecture of OC4J, and how to configure and manage OC4J.

OracleAS Portal is installed as part of the Oracle Application Server Middle Tier installation. OracleAS Portal supports a wide variety of topologies and configuration options. You learn to:

- Manage the default OracleAS Portal schemas, users, and groups
- Configure the self-registration feature
- Administer the portlet repository
- Migrate your portal content to another portal instance

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Unit 4: Application Deployment

This unit covers the following lesson:

- **Deploying J2EE Applications**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Unit 4: Application Deployment

OC4J is a core component of Oracle Application Server and is installed with the goal of managing J2EE enterprise systems. You learn to:

- Deploy J2EE applications to Oracle Application Server
- Create and manage the database connectivity for J2EE applications
- Deploy an application by using the Deployment Wizard of Application Server Control
- Create a new data source, and configure and use the new data source in your J2EE application

Unit 5: Managing Access Control

This unit covers the following lessons:

- **Providing Basic Security Services**
- **Configuring Oracle Application Server Components in Oracle Internet Directory**
- **Administering the OracleAS Single Sign-On Server**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Unit 5: Managing Access Control

You learn:

- About how the architecture and functions of Oracle Application Server provide basic security services
- About the concepts of Identity Management
- To manage users and groups in Oracle Internet Directory by using Oracle Directory Manager
- About how user passwords are managed for different components

OracleAS Single Sign-On is designed to work in an environment where multiple Web-based applications are accessible through a portal. You learn to configure and administer OracleAS Single Sign-On by using graphical user interface (GUI) and command-line interfaces.

Unit 5: Managing Access Control

This unit also covers the following lessons:

- **Managing Access to Oracle Application Server Using Oracle Delegated Administration Services**
- **Managing and Configuring Oracle Application Server Certificate Authority**
- **Securing OracleAS Components by Using SSL**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Unit 5: Managing Access Control (continued)

Oracle Delegated Administration Services enables end users to modify their own passwords without the intervention of an administrator. You learn how to start and stop Oracle Delegated Administration Services, and implement security for OracleAS Portal and portlets.

Oracle Application Server Certificate Authority can seamlessly provision new digital certificates. You learn to access the GUI-based tools to create and administer certificates.

You also learn to create and maintain wallets by using Oracle Wallet Manager and enable SSL for Oracle HTTP Server, OracleAS Portal, and OracleAS Web Cache.

Unit 6: Availability

This unit covers the following lesson:

- **Backing Up and Restoring Oracle Application Server**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Unit 6: Availability

You learn how to restore Oracle Application Server instances from a backup. You also learn how to configure backup and recovery.

Summary

In this introductory lesson, you should have learned about the course units and lessons.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Oracle Application Server 10g: Key Components and Features

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this course, you should be able to do the following:

- **Describe the solution areas addressed by Oracle Application Server**
- **Describe the key components of Oracle Application Server**
- **Explain the different installation options for Oracle Application Server**
- **Explain the installation dependencies of Oracle Application Server components**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Challenges of Creating and Maintaining E-Business Applications

The common challenges when creating and maintaining e-business applications are:

- **Development related**
- **Deployment related**

You can meet the challenges mentioned above by leveraging an integrated, complete, and open e-business platform.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

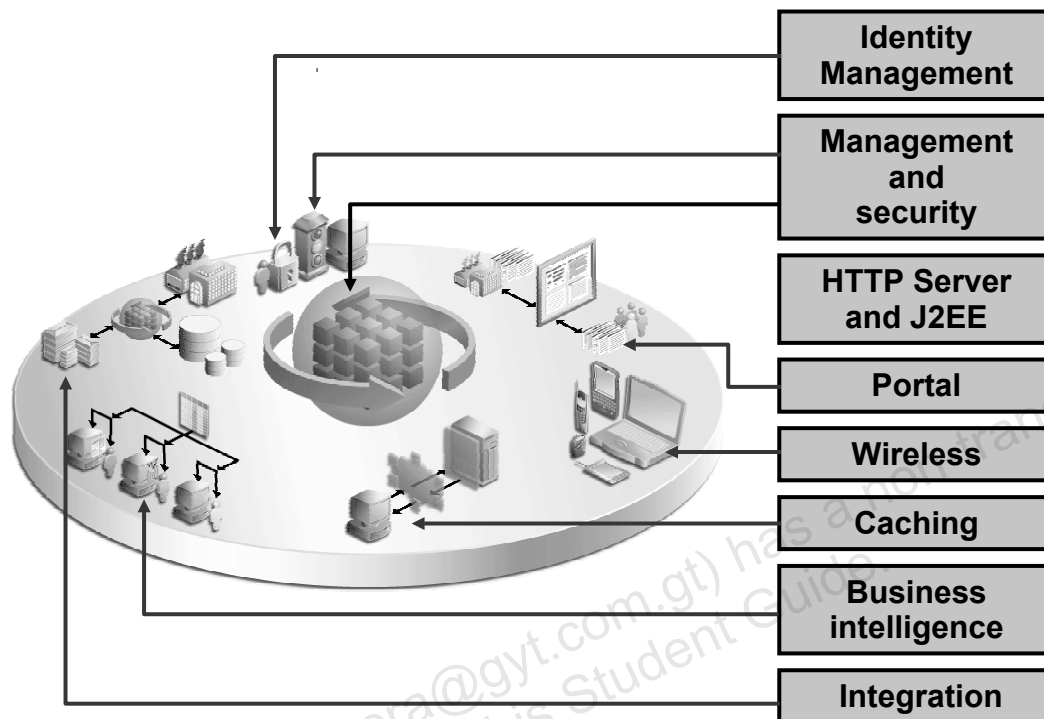
Challenges of Creating and Maintaining E-Business Applications

When you create e-business Web sites, some common challenges that you might encounter are:

- **Development related:** When you create applications, depending on the requirements, you must ensure that these applications:
 - Meet Java 2 Platform, Enterprise Edition (J2EE) standards
 - Interact directly with other software applications by using Web services
 - Can be wireless enabled for rapid access through mobile devices
 - Can be integrated with new business processes
- **Deployment related:** After your application is deployed to the server, it is important that the server can deliver the appropriate content to users quickly and reliably. Some of the challenges here include availability, scalability, performance, caching, systems management, and user and security management.

It is, therefore, very important to leverage a complete and integrated e-business platform for building an e-business solution.

Oracle Application Server: Overview



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server: Overview

Oracle Application Server is a standards-based application server that offers a fully integrated platform to develop, deploy, and administer Internet-based applications. The following section describes the solutions that Oracle Application Server provides to address the development and deployment challenges.

HTTP Server, J2EE, and Web Services

- Oracle HTTP Server functions as the HTTP interface for all Oracle Application Server components.
- Oracle Application Server is built on the J2EE framework. It enables you to design, develop, and deploy dynamic Web sites, portals, and transactional applications by using familiar languages and technologies.
- Oracle Application Server also provides comprehensive Web services to expose business functions to authorized parties over the Internet from any Web device.

Portals

- You can use Oracle Application Server to build, deploy, and maintain self-service and integrated enterprise portals. Oracle Application Server enables self-service content management and publishing, wizard-based development, and deploying, publishing, and consuming Web services on an extensible framework.

Oracle Application Server: Overview (continued)

Wireless

- OracleAS Wireless provides a simplified development and deployment of applications in a wireless environment. In addition, OracleAS Wireless includes wireless services (such as e-mail) and location-based services that simplify wireless-enabling of applications and portals.

Caching

- Oracle Application Server provides a Web-caching solution with the unique capability of caching both static and dynamically generated Web content. OracleAS Web Cache significantly improves the performance and scalability of heavily loaded Web sites. In addition, OracleAS Web Cache provides a number of features to ensure consistent and predictable responses. These features include page-fragment caching, Edge Side Includes (ESI) and Edge Side Includes for Java (JESI) support, compression, dynamic content assembly, Web-server load balancing, Web Cache clustering, and failover.

Business Intelligence

- Using the Oracle Application Server business intelligence features, you can dynamically serve personalized content recommendations to both registered and anonymous visitors as they browse your site; perform dynamic, ad hoc query reporting and analysis using a standard Web browser; and publish high-quality, dynamically generated reports on a scalable, secure platform.

Integration

- Using Oracle Application Server, you can integrate enterprise applications, trading partners, and Web services, and provide query and transaction access to many non-Oracle data sources.

Availability and Scalability

- Oracle Application Server provides a flexible deployment model that you can use to design your system for high availability and scalability.

Management and Security

- Oracle Application Server provides a set of management facilities to simplify Web site administration. You can:
 - Use Application Server Control to configure and monitor individual Oracle Application Server instances to optimize them for performance and scalability. When you need to use command-line interfaces, you can use `dcmctl` to perform configuration management, and `opmnctl` to perform process management.
 - Use encrypted secure sockets layer (SSL) connections, user- and client certificate-based authentication, and single sign-on across all applications
 - Implement an LDAP directory that provides a single repository and administration environment for user accounts

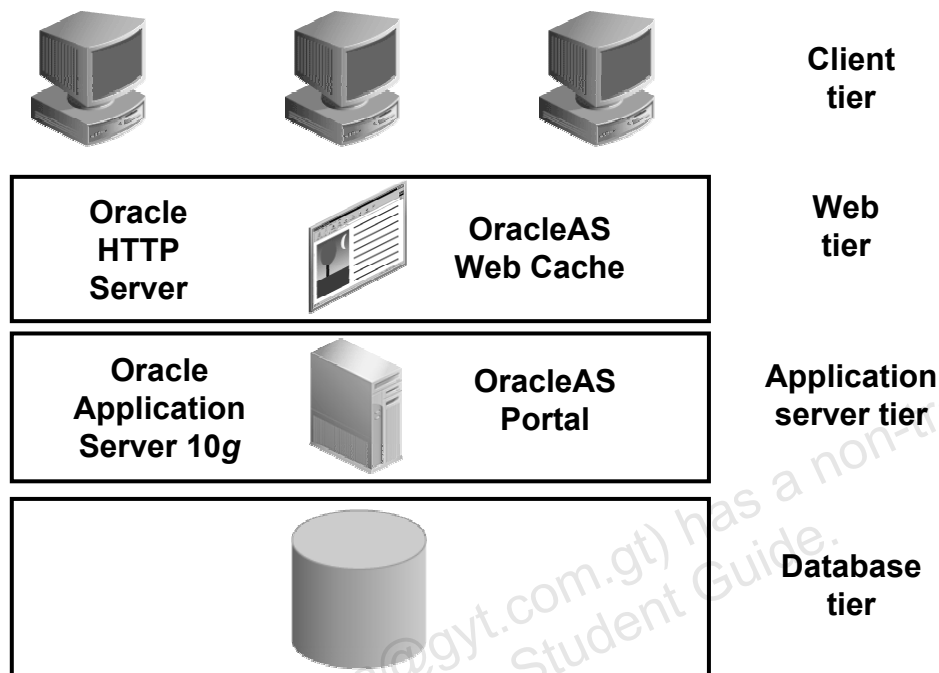
Oracle Application Server: Overview (continued)

Identity Management

- Using Identity Management, you can centralize user-management tasks and manage password policies in an enterprise setup. Identity Management is a viable solution for secure deployment of third-party applications in an Oracle environment. These deployments are based on integrating a secure framework between Oracle products and multiple third-party applications. In addition, you can independently deploy Identity Management environments in existing Oracle products, such as the Oracle database, Oracle Collaboration Suite, and Oracle E-Business Suite.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Multitiered Model



Copyright © 2005, Oracle. All rights reserved.

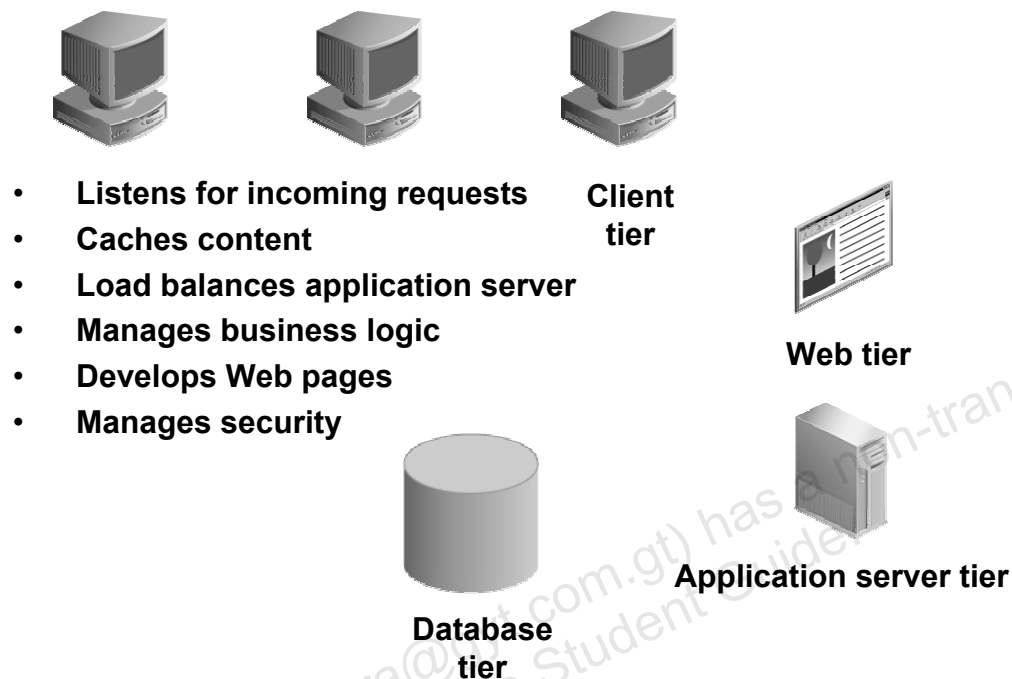
Multitiered Model

In a multitiered architecture, Oracle Application Server 10g components reside at different tiers or layers, which represent hardware layers, with each tier made up of one or more servers. In general, the number of tiers and number of servers in each tier vary depending on the Oracle Application Server 10g implementation.

The Oracle Application Server 10g architecture for a multitiered model consists of:

- **Client tier:** Containing the Web browsers for end users
- **Web tier:** Containing Oracle HTTP Server and OracleAS Web Cache
- **Application server tier:** Containing Oracle Application Server 10g
- **Database tier:** Containing the Oracle database, which can be a single instance or multiple instances managed by Real Application Clusters (RAC)

Oracle Application Server 10g Architecture



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server 10g Architecture

The functional architecture of Oracle Application Server 10g is as follows:

Web Tier

- The listener listens on a specific port for incoming requests.
- OracleAS Web Cache stores Web page components that are accessed frequently. In addition, it also load balances application servers, thus ensuring optimal allocation of computing resources.

Application Server Tier

- It controls all business logic and content assembly.
- OracleAS Portal defines Web page components.
- Oracle Single Sign-On controls security for the application server layer.

Database Tier

- It functions as a repository for the storage and retrieval of application data.
- It stores metadata.

Oracle Application Server Terminology

Oracle Application Server installation	Is the set of executables and configuration files that are created when installing Oracle Application Server
Oracle Application Server instance	Is an operational Oracle Application Server installation that runs some of the Oracle Application Server components, such as Oracle HTTP Server and Oracle Application Server Containers for J2EE (OC4J)
Oracle Application Server Infrastructure	Is a combination of OracleAS Metadata Repository and Identity Management
OracleAS Metadata Repository	Is a database of information required by Oracle Application Server instances, which are part of a farm.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Terminology

This slide gives a brief introduction to some of the key terms that are used in the administration of Oracle Application Server.

Oracle Application Server Terminology

OracleAS Farm	Is a collection of Oracle Application Server instances sharing the same configuration repository. The repository can be OracleAS Metadata Repository or a file-based repository.
OracleAS Cluster	Is a collection of Oracle Application Server instances in the same farm, with identical application deployments and functioning as a single unit
Oracle Enterprise Manager 10g Application Server Control	Manages individual Oracle Application Server instances
Oracle Enterprise Manager 10g Grid Control	Centrally manages all the components of your network and your enterprise

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Components and Solutions

J2EE and Internet Applications	Portal
Oracle HTTP Server OracleAS Containers for J2EE OracleAS TopLink Oracle Application Development Framework OracleAS Web Services Oracle XML Developer's Kit Oracle PL/SQL OracleAS MapViewer	OracleAS Portal OracleAS Portal Developer Kit
Business Intelligence and Forms	Wireless
Oracle Business Intelligence Discoverer Oracle Application Server Reports Services	OracleAS Wireless OracleAS Wireless Developer Kit Oracle Sensor Edge Server

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Components and Solutions

Oracle Application Server provides several components that help you develop, deploy, and administer your Internet-based applications. These components and the solution areas they address are highlighted in the slide. This lesson introduces some of the important components of Oracle Application Server, such as Oracle HTTP Server, OracleAS Containers for J2EE, and OracleAS Web Cache.

Oracle Application Server provides a fully integrated, J2EE-compliant platform to design, develop, and deploy dynamic Web sites, portals, and transactional applications.

With OracleAS Portal, you can personalize and secure portals, provide self-service content, and build new portlets to display custom information.

With OracleAS Wireless, you can access any corporate portal, application, or data, on any wireless device, on any network.

Using Oracle Business Intelligence Discoverer, you can access information from multidimensional OLAP or relational data sources including analytic workspaces, data warehouses, data marts, online transaction processing (OLTP) systems, and Oracle E-Business Suite.

Oracle Application Server Components and Solutions

System Management	Caching
Oracle Enterprise Manager 10g Oracle Application Server Control	OracleAS Web Cache
Identity Management and Security	Integration
OracleAS Single Sign-On Oracle Application Server Certificate Authority Java Authentication and Authorization Service Oracle Internet Directory	OracleAS Integration InterConnect OracleAS Integration B2B Oracle BPEL Process Manager OracleAS Integration Adapters

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Components and Solutions (continued)

With Oracle Application Server Forms Services, you can successfully deploy form-based applications through built-in services.

With Oracle Application Server Reports Services, you can develop and publish reports. These reports can be easily deployed on Oracle Application Server.

Oracle Application Server Components and Solutions (continued)

Later in the course, you also learn about Oracle Application Server components that are used to administer Oracle Application Server, such as OracleAS Infrastructure, Oracle Enterprise Manager 10g, Oracle Process Manager and Notification Server (OPMN), and Distributed Configuration Management (DCM).

Oracle Enterprise Manager 10g Grid Control is a Web-based system for centrally managing Oracle products, host systems, and applications. It provides a central console for monitoring distributed application servers and is integrated with Application Server Control for performing administrative operations. Oracle Enterprise Manager 10g Application Server Control is a Web-based console for administration and real-time performance monitoring of the entire application server platform, including J2EE, OracleAS Portal, and OracleAS Wireless.

Oracle Application Server provides a comprehensive, integrated set of security services for deploying applications and data on the Web. Identity Management is an integrated infrastructure that Oracle products use for distributed security.

OracleAS Web Cache accelerates static and dynamic content delivery. It uses caching, invalidation, compression, and assembly technologies to speed up the delivery of static and dynamic Web pages.

Oracle Application Server has a set of features that provide communications and integration capabilities for e-business applications.

Oracle Application Server Integration Solutions

- **Business Process Execution Language (BPEL)**
- **Oracle Application Server Integration InterConnect**
- **Oracle Application Server Integration B2B**
- **Oracle Application Server Adapter**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Components Integration Solutions

Oracle Application Server contains a set of features that enable A2A and B2B integration, and management tasks. The principles of Service-Oriented Architecture (SOA) and native support for standards such as XML enhance the usability and portability of the integration. The following are the different types of integration solutions available in Oracle Application Server:

- **BPEL:** This is a markup language for composing multiple services into an end-to-end business process. The BPEL process flow language represents machine-executable workflow. BPEL is typically generated by vendor designer tools, such as Oracle BPEL Designer. It provides support for synchronous and asynchronous interactions, parallel processing, conditional branching, and exception management. BPEL is an XML specification that defines orchestration and design of process flows for integrating heterogeneous services, such as Web services, Java services, database stored procedures, enterprise resource planning (ERP) applications, and user workflow tasks.

Oracle Application Server Components Integration Solutions (continued)

- **Oracle Application Server Integration InterConnect:** This solution works on an asynchronous communication infrastructure, which provides a robust architecture for integrated solutions.
- **Oracle Application Server Integration B2B:** This solution provides a standards-based platform to define, configure, manage, and monitor the exchange of information between two or more enterprises. This type of solution also provides a wizard interface for extensive protocol support.
- **Oracle Application Server Adapter:** This solution provides a complete solution for connecting applications and systems by using the J2EE Connector Architecture (J2CA) standards-based architecture.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Oracle Application Server Products

Oracle Application Server

Known as OracleAS Middle Tier; it includes Oracle HTTP Server, OC4J, OracleAS Web Cache, OracleAS Portal, OracleAS Wireless, and others

OracleAS Infrastructure

Identity Management services, and OracleAS Metadata Repository

OracleAS Developer Kits

Includes APIs and simple developer kits. This does not include Oracle Developer Suite products.

Application Server Control is installed with each Oracle Application Server installation.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Products

Oracle Application Server comprises three product sets:

- **Oracle Application Server:** Oracle Application Server is an integrated platform that enables you to deliver Web content, host Web applications, connect to back-office applications, and access your data on wireless devices.
- **OracleAS Infrastructure:** OracleAS Infrastructure consists of an Oracle database, OracleAS Single Sign-On, and a directory server. The database contains a collection of schemas and metadata that are used by the Oracle Application Server components. OracleAS Infrastructure is required for most OracleAS Middle Tier applications. It must be installed and configured before you install Oracle Application Server. You should install Infrastructure on a dedicated machine for optimal performance.
- **OracleAS Developer Kits:** OracleAS Developer Kits enables the user to create XML applications, develop portlets, enable wireless applications, integrate Web sites with wireless devices, and develop application provider Web services. OracleAS Developer Kits installs Oracle XML Developer Kit, OracleAS Portal Developer Kit, OracleAS Wireless Developer Kit, and Oracle LDAP Developer Kit.

Oracle Application Server Installation Types

Each Oracle Application Server product has installation types that enable you to select the Oracle Application Server components for your installation.

J2EE and Web Cache

Installs and configures Oracle HTTP Server, OC4J with J2EE 1.3, Web Services, and OracleAS Web Cache

Portal and Wireless

Installs and configures Portal and Wireless components, along with the J2EE and OracleAS Web Cache components

Business Intelligence and Forms

Installs and configures J2EE and Web Cache and Portal and Wireless components, along with Forms, Reports, and Discoverer

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Installation Types

Each Oracle Application Server product provides installation types that enable you to select the Oracle Application Server components for your installation.

Oracle Application Server offers the following installation types:

- **J2EE and Web Cache:** Provides a basic Web server that implements J2EE applications and accelerates Web caching
- **Portal and Wireless:** Enables the deployment of enterprise portals and wireless applications. This installation type includes the components available in the J2EE and Web Cache edition.
- **Business Intelligence and Forms:** Provides the deployment of Business Intelligence components. This installation type includes the components available in the J2EE and Web Cache edition, and Portal and Wireless edition.

Before installing an instance of Portal and Wireless or Business Intelligence and Forms, you must install and configure OracleAS Infrastructure in your network, optimally on a separate machine.

Oracle Application Server Management

Oracle Application Server provides the following management tools:

- **Grid Control:**
 - To manage multiple Oracle Application Server instances
- **Application Server Control:**
 - The preferred browser-based interface, which can be used from a remote location

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Management

Use Oracle Enterprise Manager 10g Grid Control to manage an enterprise environment that includes Oracle products and applications, in addition to Oracle Application Server. From a central location, you can use Grid Control to manage databases, application servers, and Oracle applications across your entire network.

The primary tool for managing individual Oracle Application Server instances is Oracle Enterprise Manager 10g Application Server Control.

Oracle Enterprise Manager 10g Application Server Control is installed with every instance of Oracle Application Server. From the Application Server Control Console, you can monitor and administer a single Oracle Application Server instance, an OracleAS Farm of application server instances, or an Oracle Application Server cluster.

Oracle Application Server Management

- **Oracle Process Management and Notification Server (OPMN):**
 - It monitors Oracle Application Server processes, and restarts them when needed.
 - `opmnctl` is the command-line interface.
- **Distributed Configuration Management (DCM):**
 - It manages the configuration and maintains the configuration repository.
 - `dcmctl` is the command-line interface.

ORACLE

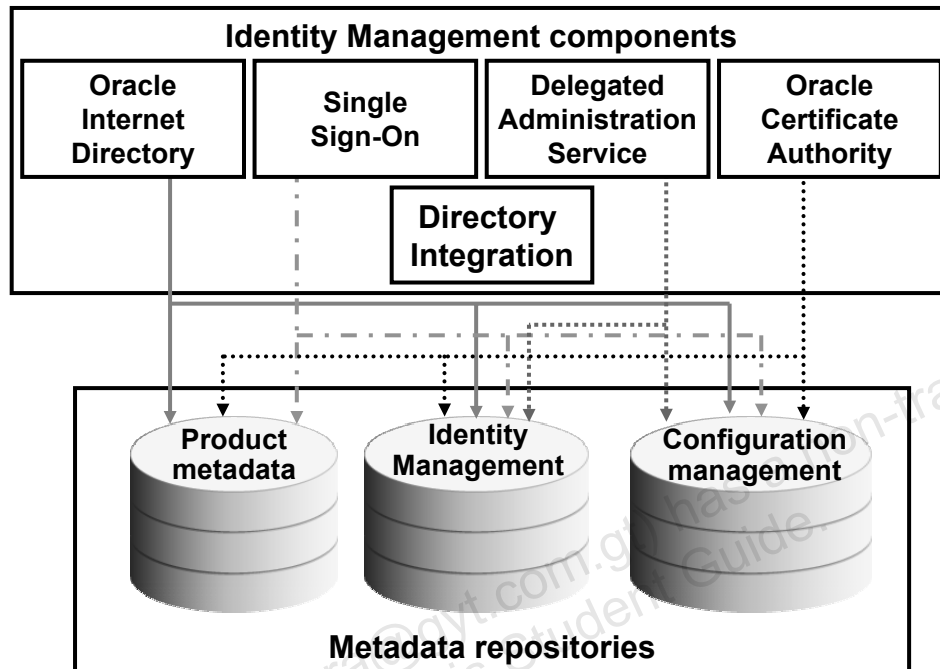
Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Management (continued)

In addition to Application Server Control, Oracle Application Server provides command-line interfaces to several key management technologies. The command-line tools can help you automate your management procedures with scripts and custom utilities. The two most important command-line tools are the following:

- `opmnctl`, which provides a command-line interface to Oracle Process Management Notification Server (OPMN)
- `dcmctl`, which provides a command-line interface to Distributed Configuration Management (DCM)

OracleAS Infrastructure



Copyright © 2005, Oracle. All rights reserved.

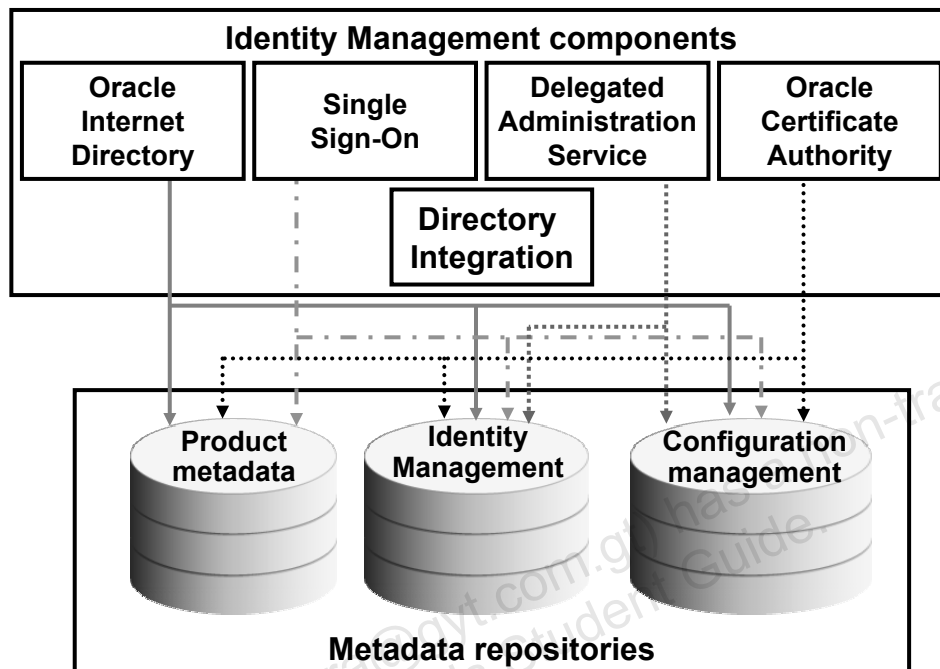
OracleAS Infrastructure

OracleAS Infrastructure provides centralized services that are related to the product metadata, Identity Management, and configuration management.

OracleAS Infrastructure provides centralized Identity Management services, configuration information, and data repositories for middle-tier installations. The key features that middle-tier instances typically use are the following:

- **Product Metadata Service:** Product Metadata Service provides all of the metadata that the middle-tier instances require. It is bundled as part of the OracleAS Infrastructure. Product Metadata is looked up by middle-tier Oracle Application Server instances for the successful execution of applications. Product metadata is not accessed directly by customer applications.
- **Security Service:** Security Service provides a consistent security model for all Oracle Application Server applications. It also provides a single source of identity metadata that contains all administration and user privileges.

OracleAS Infrastructure Components



Copyright © 2005, Oracle. All rights reserved.

OracleAS Infrastructure Components

Infrastructure components can be grouped into Identity Management components and OracleAS Metadata Repository components. When you install Infrastructure, you can specify whether you want to install Identity Management components, OracleAS Metadata Repository, or both. The Oracle HTTP Server, OracleAS Containers for J2EE (OC4J), and Application Server Control components are always installed, regardless of the installation type you selected.

- Identity Management components provide directory, security, and user-management functionality:
 - Oracle Internet Directory
 - OracleAS Single Sign-On
 - Oracle Delegated Administration Services
 - Oracle Directory Integration and Provisioning
 - Oracle Application Server Certificate Authority

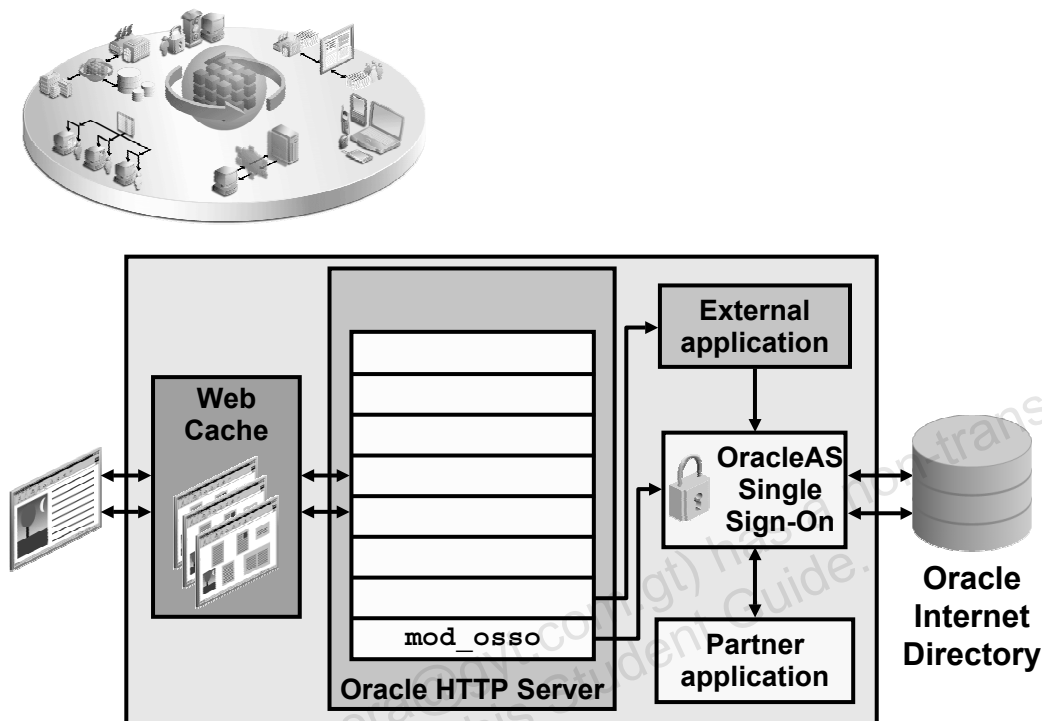
Some of these components (such as OracleAS Single Sign-On) have schemas in OracleAS Metadata Repository.

OracleAS Infrastructure Components (continued)

- OracleAS Metadata Repository is a collection of schemas that are used by other Oracle Application Server components. The schemas can be grouped into the following categories:
 - Product metadata
 - Identity Management metadata
 - Configuration Management metadata

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Oracle Internet Directory and Security



Copyright © 2005, Oracle. All rights reserved.

Oracle Internet Directory and Security

Oracle Internet Directory is an LDAP server that can be used to store all the credentials required for the enterprise. Oracle Internet Directory offers comprehensive and flexible support for directory access control. This includes entry-level, attribute-level, and prescriptive access control, to provide varying levels of security to fit enterprise and service provider needs.

Oracle Internet Directory implements three levels of user authentication:

- Anonymous
- Password-based
- Certificate-based, using secure sockets layer (SSL) for authenticated access and data privacy

The Web-based Oracle Delegated Administration Services enables application administrators to delegate user-management tasks, such as granting or restricting access to a specific directory attribute, entry, group, or naming context to application users.

After Oracle Internet Directory is deployed, organizations can use OracleAS Single Sign-On to provide a single point of validation for user credentials. After users sign on successfully, their credentials are automatically retrieved from Oracle Internet Directory when they launch any Oracle partner application.

Securing the Web Infrastructure

- **Oracle Application Server provides a comprehensive suite of security services, including OracleAS Single Sign-On.**
- **Secure sockets layer (SSL) encryption can be used to protect the Web site.**
- **OracleAS Single Sign-On validates user credentials against Oracle Internet Directory, which is an LDAP directory service.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Securing the Web Infrastructure

For network encryption and authentication, Oracle Application Server provides a comprehensive suite of security services, including OracleAS Single Sign-On. OracleAS Single Sign-On validates user credentials against Oracle Internet Directory, which is an LDAP directory service.

Also, SSL encryption can be used to protect these transactions against malicious intrusion.

OracleAS Infrastructure Installation Types

- **OracleAS Infrastructure components are grouped into two categories:**
 - Identity Management components
 - OracleAS Metadata Repository components
- **During an OracleAS Infrastructure installation, you can choose to install:**
 - Identity Management
 - Metadata Repository
 - Both Identity Management and Metadata Repository
- **This provides you with the flexibility to install different components on different systems or databases.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Infrastructure Installation Types

OracleAS Infrastructure components are grouped into Identity Management components and OracleAS Metadata Repository components. This grouping provides you with the flexibility to install the OracleAS Infrastructure components on multiple computers or databases.

For example, you can install OracleAS Metadata Repository on one computer and the Identity Management components on another computer. You can install the Identity Management components on multiple computers as well.

Selecting either the “OracleAS Metadata Repository” or the “OracleAS Metadata Repository and Identity Management” option causes the Oracle Universal Installer (Installer) to create a new database and populate it with OracleAS Metadata Repository.

When you install only OracleAS Metadata Repository, Application Server Control is not installed; instead, you can use Oracle Enterprise Manager10g Database Control to manage OracleAS Metadata Repository.

Selecting the Identity Management option requires that you have an existing OracleAS Metadata Repository.

You can use an existing database for creating OracleAS Metadata Repository by using the repository-creation utility.

Installation Types That Require Infrastructure

- **The Portal and Wireless installation type needs OracleAS Infrastructure as a prerequisite.**
- **In a J2EE and Web Cache installation type, you require:**
 - **OracleAS Metadata Repository to use database-managed Application Server clustering**
 - **Identity Management to use OracleAS Single Sign-On**
- **In a Business Intelligence and Forms installation type, you need OracleAS Infrastructure as a prerequisite.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Installation Types That Require Infrastructure

The J2EE and Web Cache installation type does not require OracleAS Infrastructure.

It is necessary to have Identity Management already installed if you intend to use OracleAS Single Sign-On. Similarly, you would need OracleAS Metadata Repository installed to use the database-managed application server clustering. An application server cluster is a collection of application server instances with identical configuration and application deployment. Clusters enforce homogeneity among member instances so that a cluster of application server instances can appear and function as a single instance.

Before installing the Portal and Wireless installation type, you must install and configure OracleAS Infrastructure somewhere in your network, optimally on a separate machine.

The Business Intelligence and Forms installation requires OracleAS Infrastructure by default. This installation of OracleAS Infrastructure might be somewhere in your network or on the same host that has the Business Intelligence and Forms installation.

Services and Component Matrix for OracleAS Infrastructure

Service	Description	Components
Product Metadata service	Schemas for components such as Portal and Wireless	OracleAS Metadata Repository
Identity Management service	A consistent security model for all applications Single source of security metadata containing all administration and user privileges	<ul style="list-style-type: none"> • Oracle Internet Directory • OracleAS Single Sign-On • Oracle Delegated Administration Services • Oracle Directory Integration and Provisioning • Oracle Application Server Certificate Authority
Configuration Management service	Schemas containing Oracle Application Server instance configuration	OracleAS Metadata Repository

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Services and Component Matrix for OracleAS Infrastructure

You must install OracleAS Infrastructure before you can install Oracle Application Server (middle tier), because the information about Oracle Internet Directory and Metadata Repository is required during certain middle-tier installations.

OracleAS Middle Tier Components

	Installation Type		
	<i>J2EE and Web Cache</i>	<i>Portal and Wireless</i>	<i>Business Intelligence and Forms</i>
Component	OracleAS Web Cache	X	X
	Oracle HTTP Server	X	X
	OracleAS Containers for J2EE	X	X
	Oracle Enterprise Manager 10g Application Server Control	X	X
	OracleAS Portal	X	X
	OracleAS Wireless	X	X
	OracleAS Forms		X
	OracleAS Reports		X
	Oracle Business Intelligence Discoverer		X

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Middle Tier Components

The table in the slide lists the installation options for Oracle Application Server, and the components that are installed with each option. As explained earlier, each installation installs all the components that are applicable to that installation type in all situations. The user can control what is configured, but not what is installed.

J2EE and Web Cache Installation

You can use the J2EE and Web Cache installation type to develop and deploy Java and J2EE applications, to improve the speed of your Web site with Web Cache, and to use J2EE and Web services. This topology does not support single sign-on or clustering functionality. In order to use single sign-on or clustering functionality, or Oracle Internet Directory, you must install OracleAS Infrastructure.

For the J2EE and Web Cache installation type, it is not a prerequisite to have OracleAS Infrastructure installed.

OracleAS Middle Tier Components (continued)

Portal and Wireless Installation

You can use OracleAS Portal to assemble and publish Web pages that consist of portlets. When you install OracleAS Portal, you can use Oracle Ultra Search to index and search database tables, files, Web sites, and mailing lists. You can also use Universal Discovery, Description and Integration (UDDI) to publish your Web services in a UDDI Registry. You can use OracleAS Wireless to develop applications for the mobile environment.

Business Intelligence and Forms Installation

You can use the Business Intelligence and Forms installation type to develop and deploy forms and reports over an enterprise. These forms and reports can retrieve information from multidimensional OLAP or relational data sources, including analytic workspaces, data warehouses, data marts, OLTP systems, and Oracle E-Business Suite. You can also use Oracle Business Intelligence Discoverer to access information from these data sources.

For the Business Intelligence and Forms installation type, it is a prerequisite to have OracleAS Infrastructure installed.

Regardless of the middle-tier installation type, the following components are installed:

- **Oracle HTTP Server**

Based on the industry-leading Apache Web Server, Oracle HTTP Server is the Web server component of Oracle Application Server. Oracle HTTP Server incorporates extended Apache functionality to provide SSL and HTTPS support. Oracle HTTP Server dispatches requests to invoke program logic written in Java, PL/SQL, PERL, PHP, or as CGI executables through a standard Apache module architecture.

- **OracleAS Web Cache**

OracleAS Web Cache operates as a caching reverse proxy server that is situated in front of Oracle HTTP Server. It improves performance of Web server instances by storing frequently accessed pages in memory, thereby eliminating the need to repeatedly process requests for pages from the Web server, the applications, or the Oracle database.

- **OracleAS Containers for J2EE (OC4J)**

Oracle Application Server provides a fast, lightweight, highly scalable, easy-to-use, and complete Java 2 Platform, Enterprise Edition (J2EE) container written entirely in Java. This container executes on the standard Java Development Kit (JDK) or Java Virtual Machine (JVM) available on the operating systems and hardware platforms on which Oracle Application Server is certified.

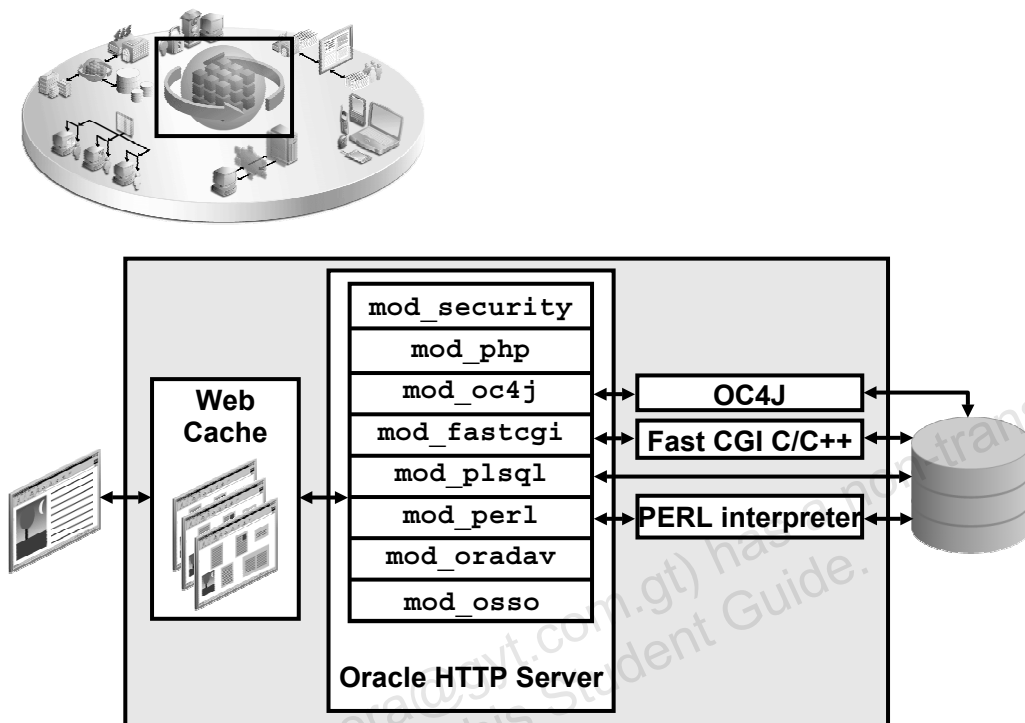
- **OracleAS Web Services**

Web services are a set of emerging standards that enable Internet applications to be developed and deployed in a Service-Oriented Architecture and to communicate with each other in standard ways. Some examples of Web services are currency converter, stock quotes, travel planner, and procurement workflow system.

- **Application Server Control**

Application Server Control is the Web-based administration interface for centrally managing your Oracle Application Server platform. Application Server Control provides a fully integrated monitoring, management, and diagnostics environment specifically for Oracle Application Server.

Oracle HTTP Server



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle HTTP Server

Oracle HTTP Server is the underlying deployment platform, and it provides a Web listener for OracleAS Containers for J2EE (OC4J) and the framework for hosting static and dynamic pages and applications on the Web. Oracle HTTP Server is based on Apache, and has been enhanced with the following modules:

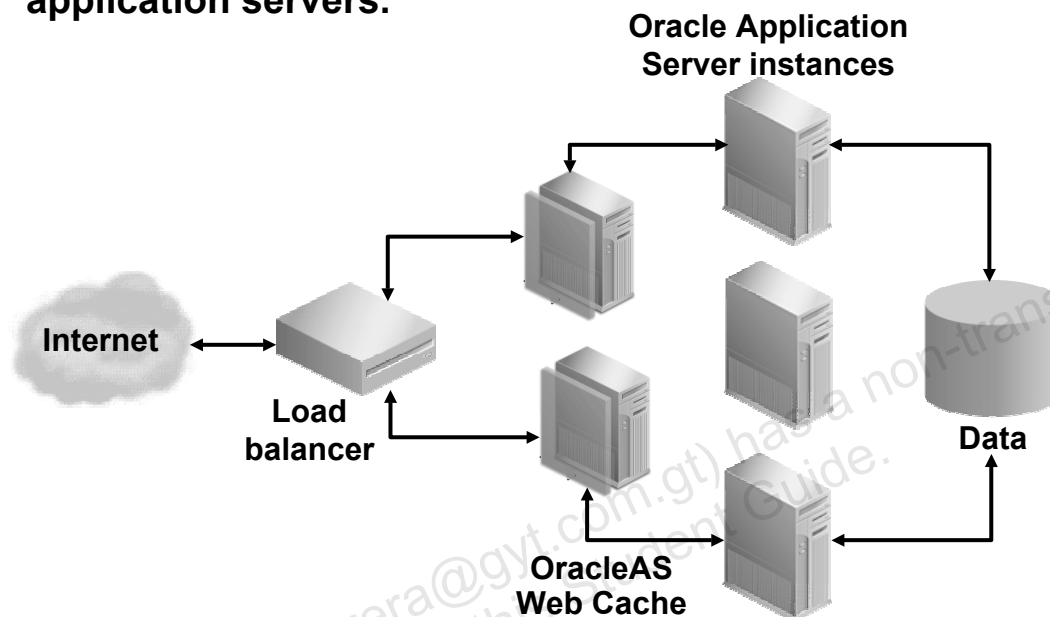
- **mod_security:** Is an open source intrusion detection and prevention engine for Web applications, which provides protection against known and unknown attacks
- **mod_php:** Enables PHP scripts to be executed in Oracle HTTP Server
- **mod_oc4j:** Routes communication between Oracle HTTP Server and OC4J
- **mod_fastcgi:** Supports persistent CGI processes
- **mod_plsql:** Routes requests for stored procedures to the database server
- **mod_perl:** Routes PERL requests to the PERL interpreter
- **mod_oradav:** Supports file- as well as database-distributed authoring and versioning
- **mod_osso:** Routes requests to OracleAS Single Sign-On

This is not a complete list, and some of the modules are discussed in detail later.

With Oracle Application Server, developers can choose familiar languages and technology to build Web sites and applications, including Java, XML, PL/SQL, PERL, C, C++, PHP, and Distributed Authoring and Versioning (DAV).

OracleAS Web Cache

OracleAS Web Cache functions as a front end for application servers.



ORACLE®

Copyright © 2005, Oracle. All rights reserved.

OracleAS Web Cache

OracleAS Web Cache functions as a front end for application servers. The first time that OracleAS Web Cache receives an HTTP or HTTPS request, it forwards the request to an HTTP server for processing. Web Cache stores the response from the HTTP server in memory so that it can respond directly to future requests.

OracleAS Web Cache understands HTTP headers, including cookies, and makes caching decisions based on administrator- or application-defined rules.

To invalidate the cache, administrators can specify expiration policies or applications can send an HTTP invalidation message.

Deploying OracleAS Web Cache before a farm of application or HTTP servers enables clustering, surge protection, and Web-server load balancing, so that cache misses are directed to the most available, highest-performing origin Web server. These cache misses are HTTP or HTTPS requests that cannot be served from the cache and are further forwarded to an origin server.

OracleAS Web Cache is a content-aware reverse proxy and content accelerator that can be clustered to provide scalability and availability.

OracleAS Web Cache (continued)

To Web browsers, OracleAS Web Cache acts as the virtual server for application Web servers. You configure a load balancer with the same IP address that is registered for a site's domain name and the application Web servers' host names. This load balancer receives requests for OracleAS Web Cache.

This configuration enables Web browsers to communicate with OracleAS Web Cache rather than application Web servers when accessing a Web site.

OracleAS Web Cache attempts to handle a request for cacheable content from its memory cache. If that is not successful, it passes the request to an application server instance.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Enhancing Performance with Caching

- **OracleAS Web Cache enables you to:**
 - Accelerate the delivery of static and dynamic content
 - Reduce your hardware and administration costs
- **You can cluster multiple Web Cache instances to:**
 - Provide ease of configuration and management
 - Avoid a single point of failure

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Accelerating Performance with Caching

OracleAS Web Cache can render the service from a Web site faster by reducing unnecessary hits on the other middle-tier and back-end components.

Furthermore, deploying Web Cache helps to reduce your hardware and administration costs. In a distributed environment, you can deploy Web Cache on machines at remote sites instead of deploying multiple HTTP servers. As a result, many requests can be handled locally by Web Cache, avoiding middle-tier and back-end processing, as well as slower throughput on WANs.

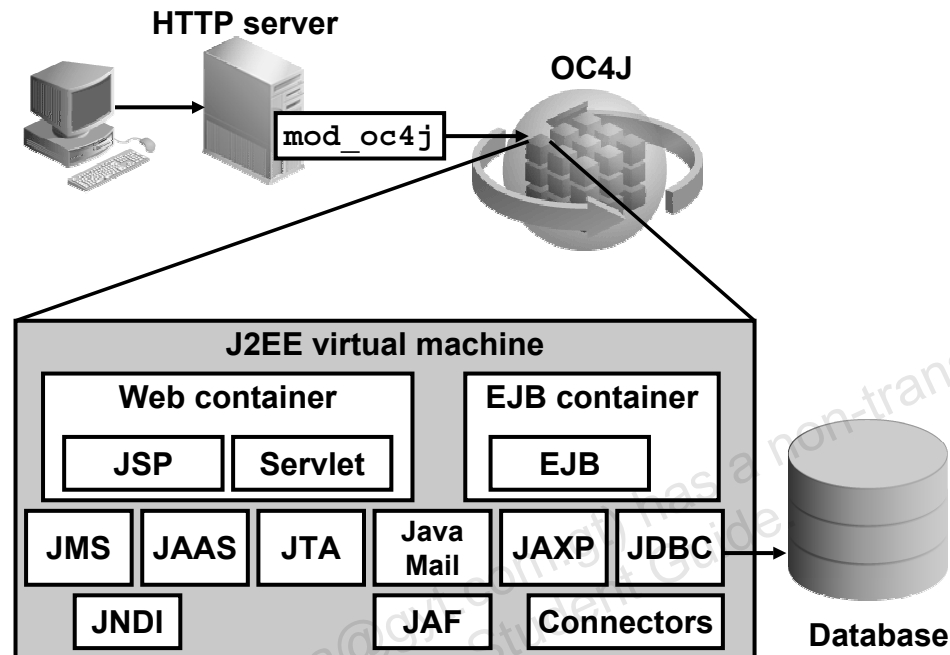
Web Cache accelerates delivery of both static and dynamic contents. Caching prevents overload at the origin servers, reduces bandwidth through compression, and increases Web site availability by throttling sudden traffic surges.

Web Cache also provides load balancing, by distributing cache miss requests according to the relative capacity of each HTTP server.

Multiple instances of Web Cache, called cluster members, can operate as one logical cache. They communicate with one another to request cacheable content, which is cached by another cache cluster member, and to detect when a cache cluster member fails.

To enable cache clusters to function as a single unit, you need to set up a load balancer.

OracleAS Containers for J2EE (OC4J)



Copyright © 2005, Oracle. All rights reserved.

OracleAS Containers for J2EE (OC4J)

The J2EE platform that is provided in Oracle Application Server uses a multitiered distributed application model that divides application logic into components according to function.

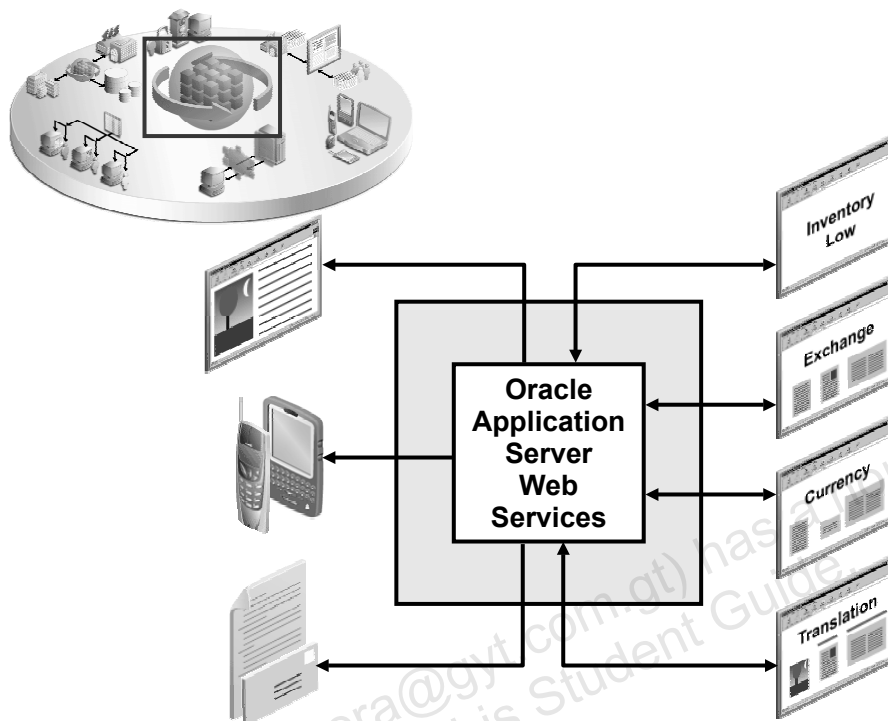
A container provides run-time support for J2EE application components. Containers provide a federated view of the underlying J2EE APIs to the application components.

OracleAS Containers for J2EE (OC4J) is a J2EE server implementation that runs on a standard Java Virtual Machine (JVM). OC4J has the following J2EE containers:

- A Web container that has:
 - A servlet container
 - A JSP container
- An EJB container that has:
 - Session Beans
 - Entity Beans
 - Message-Driven Beans

The J2EE concepts are explained in detail later in the course.

OracleAS Web Services



ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Web Services

You can use Web Services to expose your applications in a manner you choose, so that they can receive formatted instructions over the Web (for example, a location information service that provides information about a location from postal code, area code, city, and state).

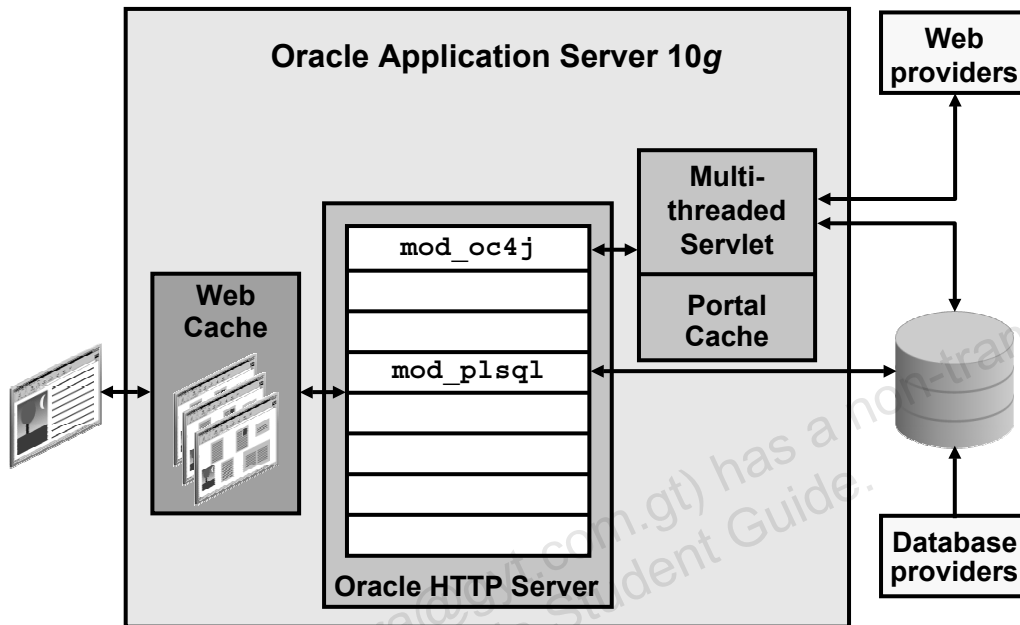
A Web service is a discrete business process that:

- Exposes and describes its functionality and attributes in Web Services Description Language (WSDL)
- Uses the Universal Description, Discovery and Integration (UDDI) registries to allow other services to locate a service on the Web, such as the translation or currency converter service
- Allows remote services to invoke a service by using standard Internet protocols
- Returns a response to the requesting application over the same protocol

OracleAS Web Services:

- Provides support for developing and deploying Web services
- Runs as servlets in the OC4J servlet container
- Supports both Remote Procedure Call (RPC)-style exchange, and message-oriented or Document style exchange

OracleAS Enterprise Portal



ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Enterprise Portal

Portals enable clients to access information through any Web browser. This information usually comes from different data sources that the portal combines into a single entry point. Portals also support personalized views, so that each user or user group can customize both the content and the appearance of the portal to suit individual preferences and requirements.

OracleAS Portal is a Web-based tool for building and deploying e-business portals. It provides a secure, manageable environment for accessing and interacting with enterprise software services and information resources. It enables you to efficiently manage, access, and interact with information by enabling you to create portal pages.

OracleAS Portal has an extensible framework that integrates information components called portlets. Portlets are Web-based resources (such as Web pages, applications, business intelligence reports, and syndicated content feeds) within standardized, reusable information components.

The OracleAS Portal interface provides an organized, consistent view of the business information, Web content, and applications that each user needs.

The self-service publishing features of OracleAS Portal enable authorized users to post and share any kind of document or Web content with other users anywhere in the world.

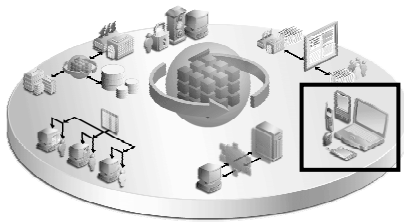
OracleAS Enterprise Portal (continued)

You can use OracleAS Portal to build and customize Enterprise Information Portals (EIP). The request enters the server farm through OracleAS Web Cache and is evaluated by Oracle HTTP Server. The packages that define the objects and pages reside as packages in the database.

The parallel page engine is a multithreaded servlet running in OC4J. Also, you can enable OracleAS Web Cache to perform the task of assembling pages. After the page is assembled, it is returned to the client.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Wireless-Enabled Applications



Using OracleAS Wireless, you can:

- **Develop or extend applications to be location based, personalized, or voice enabled, and deploy them to all devices**
- **Provide personalization from PCs or wireless devices**
- **Use advanced messaging techniques, such as voice messaging, short message service (SMS), or e-mail**

ORACLE

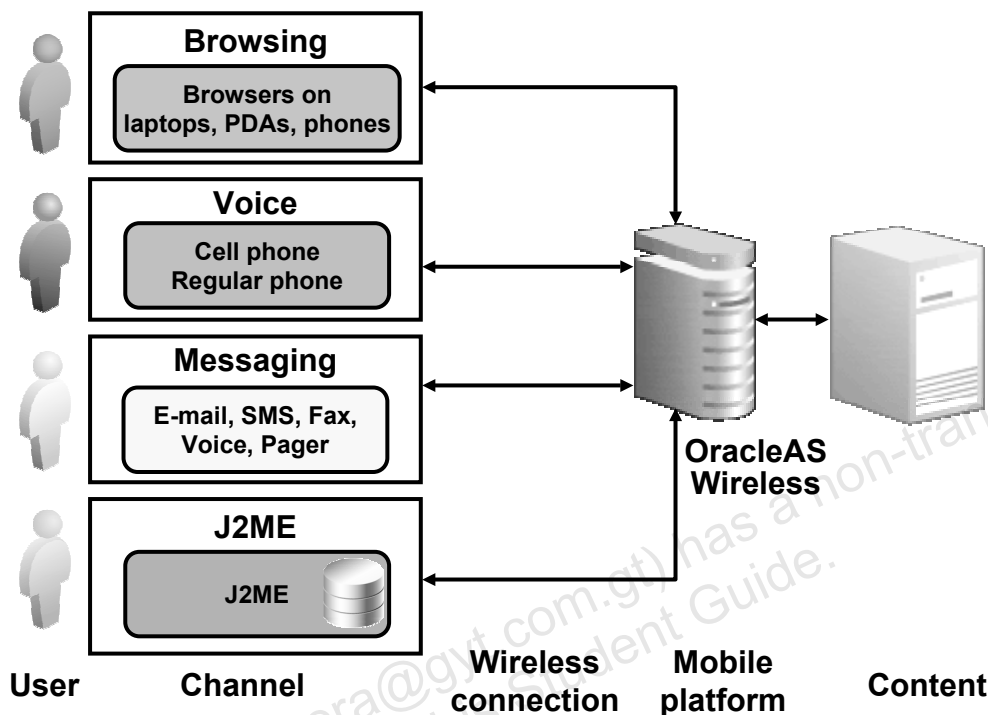
Copyright © 2005, Oracle. All rights reserved.

Wireless-Enabled Applications

Mobile users increasingly rely on wireless devices for communication while away from the office. OracleAS Wireless enables enterprises and service providers to efficiently build, manage, and maintain wireless and voice applications. OracleAS Wireless also provides:

- Geographic modeling that turns existing applications into location-based applications
- E-mail and directory modules to access corporate e-mail and directory applications:
 - mWallet supports mobile commerce transactions and tracking.
 - Mobile E-mail supports accessing IMAP and POP e-mail.
 - Mobile Directory supports access to LDAP directories.
 - Mobile Calendar enables schedule and appointment management.
 - Instant Messaging supports exchanging instant messages from mobile devices.
- Open platform standards for simple development and easy integration with existing applications

Wireless-Enabled Applications



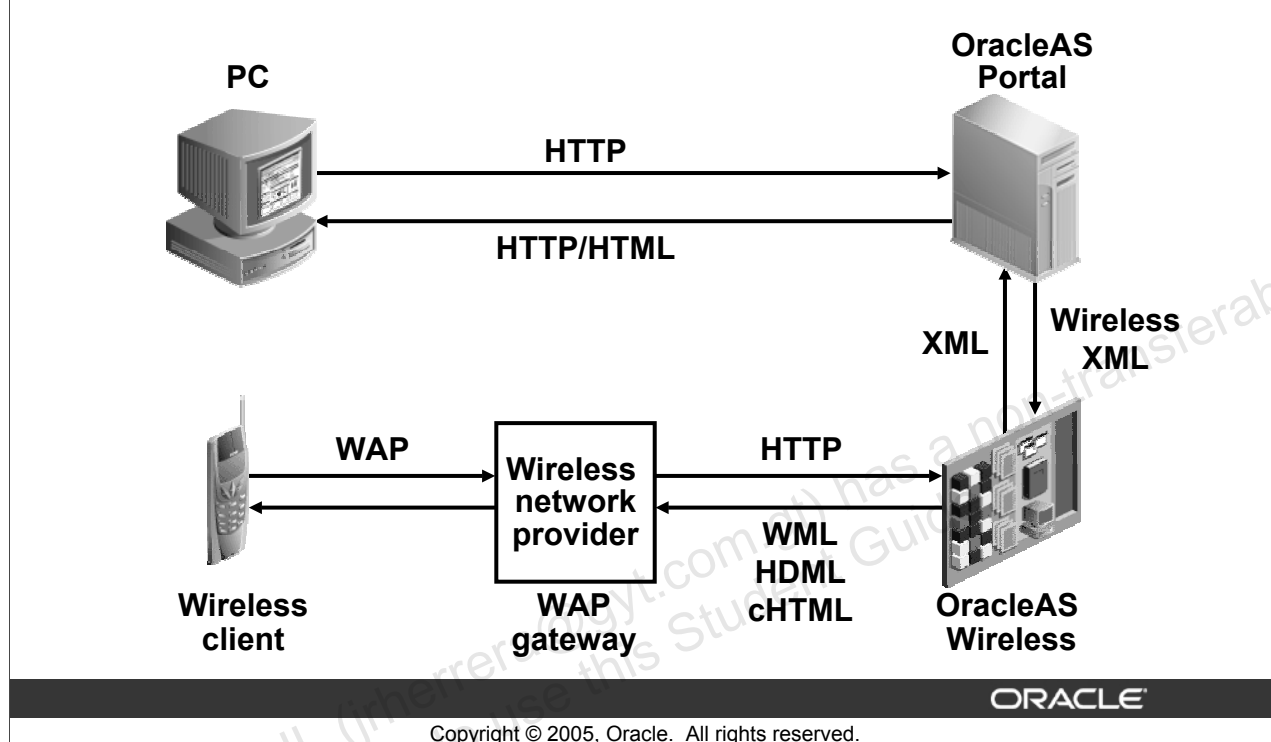
Copyright © 2005, Oracle. All rights reserved.

Wireless-Enabled Applications (continued)

- Tools to turn applications into voice applications accessible from non-Web phones:
 - Service Designer helps developers manage applications.
 - Content Development Tool helps the end user to increase his mobile experience.
 - Help Desk provides support to end users.
 - System Monitor helps manage the OracleAS Wireless environment.

OracleAS Wireless includes a set of services that allow content access by wireless devices. The base station and IP router convert the signals between wire and radio waves. The Wireless Application Protocol (WAP) gateway is responsible for making the XML data translation from the required format of the specific device to a standard XML message and back. A component of this WAP gateway transforms the XML data from the wireless application to specific clients. This allows Oracle Application Server to service the requests as standard requests, so that a single application can be accessed from any device.

Mobile Portal Architecture



Mobile Portal Architecture

OracleAS Portal can be accessed not only from Web browsers through HTTP but also from mobile devices.

The requests for portal pages coming from a mobile device over WAP need to go through a WAP gateway, which is the wireless network provider (for example, PacBell or Sprint PCS), that authenticates the wireless device and the subscriber. The wireless network provider sends the user information (for example, the phone number), device identification (for example, the model, browser/mobile language type), and location (for example, the spatial information, such as subscriber location within the cell site) to the OracleAS Wireless that acts as an intermediary between the mobile device and portal. The requests are passed to the portal, which responds with a device-independent markup language, mobileXML. OracleAS Wireless then transforms mobileXML to the actual language of the mobile device, such as WML, HDML, or cHTML, and returns the information to the WAP gateway to be rendered on the mobile client.

OracleAS Developer Kits

OracleAS Developer Kits enables the user to:

- **Develop portlets**
- **Enable wireless applications**
- **Integrate Web sites with wireless devices**
- **Develop application provider Web services**
- **Create XML applications**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Developer Kits

OracleAS Developer Kits includes the Portal, Wireless, XML, and Lightweight Directory Access Protocol (LDAP) developer kits. In addition, Oracle Application Server provides other toolkits for developing applications. For more information about installing OracleAS Developer Kits, see *Oracle® Application Server Installation Guide 10g Release 2 (10.1.2.0.2) for hp HP-UX PA-RISC (64-bit), and Linux x86*.

Summary

In this lesson, you should have learned how to:

- **Describe the solution areas addressed by Oracle Application Server**
- **Describe the key components of Oracle Application Server and their features**
- **Describe how the main components build the Oracle Application Server architecture**
- **Explain the different installation options for Oracle Application Server**
- **Explain the installation dependencies of Oracle Application Server components**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

3

Installing OracleAS Infrastructure

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

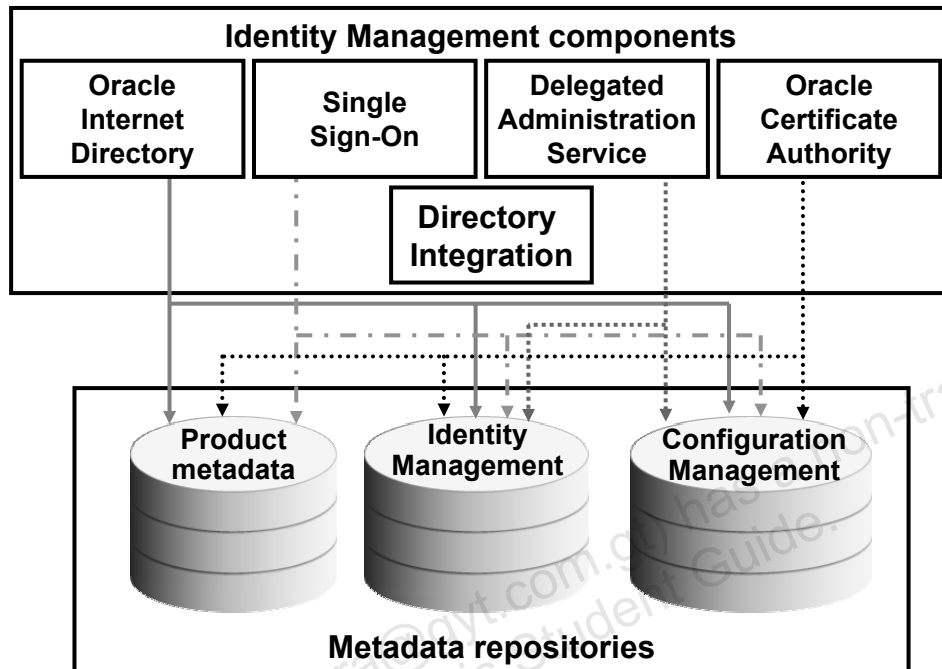
After completing this lesson, you should be able to do the following:

- **Define the installation requirements for OracleAS Infrastructure**
- **Describe OracleAS Infrastructure installation types**
- **Install OracleAS Infrastructure**
- **Start and stop OracleAS Infrastructure**

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

OracleAS Infrastructure Components



ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Infrastructure Components

Infrastructure components can be grouped into Identity Management components and OracleAS Metadata Repository components. When you install Infrastructure, you can specify whether you want to install the Identity Management components or OracleAS Metadata Repository, or both. The Oracle HTTP Server, OracleAS Containers for J2EE (OC4J), and the Application Server Control components are always installed, regardless of the installation type you selected.

- **Identity Management components:** These components provide directory, security, and user-management functionality:
 - Oracle Internet Directory
 - OracleAS Single Sign-On
 - Oracle Delegated Administration Services
 - Oracle Directory Integration and Provisioning
 - Oracle Application Server Certificate Authority

Some of these components (such as OracleAS Single Sign-On) have schemas in OracleAS Metadata Repository.

OracleAS Infrastructure Components (continued)

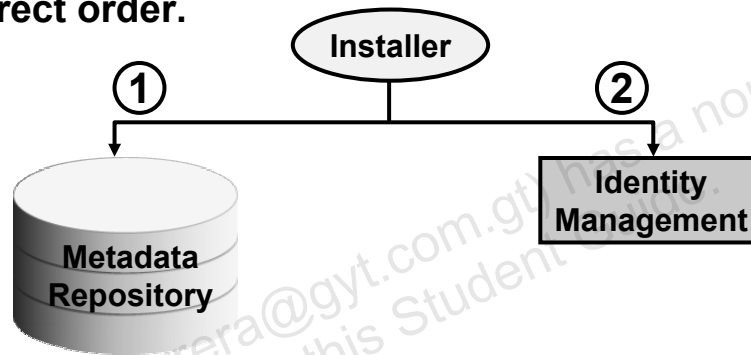
- OracleAS Metadata Repository is a collection of schemas that are used by other Oracle Application Server components. The schemas can be grouped into the following categories:
 - Product metadata
 - Identity Management metadata
 - Configuration Management metadata

You must install OracleAS Infrastructure before you can install OracleAS Middle Tier, because the information about Oracle Internet Directory and metadata repository is required during the middle-tier installation.

The only scenario where you do not need to install OracleAS Infrastructure first is when you are installing a J2EE and Web Cache instance without the database-managed OracleAS clusters or the Identity Management features. In this case, you simply install the J2EE and Web Cache; you do not need to install any OracleAS Infrastructure services at all. If you later decide that you want to associate your J2EE and Web Cache instance with OracleAS Infrastructure, then you can install OracleAS Infrastructure and do the association.

Order of Installing OracleAS Infrastructure Components

- When you choose to install components on different systems, you should do the following:
 1. Install OracleAS Metadata Repository
 2. Install the Identity Management components
- When you install both, the Installer uses the correct order.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Order of Installing OracleAS Infrastructure Components

If you plan to install OracleAS Infrastructure on separate computers, you must install them in the following order:

1. Install OracleAS Metadata Repository:
 - a. You can have the Installer create a new database and populate it with OracleAS Metadata Repository, or you can install OracleAS Metadata Repository in an existing database.
 - b. You cannot register OracleAS Metadata Repository with Oracle Internet Directory at this point, because you do not have Oracle Internet Directory yet. The registration is performed in the next step.
2. Install the Identity Management components:
 - a. The Installer prompts you to enter the connect information for the OracleAS Metadata Repository database.
 - b. The Installer registers OracleAS Metadata Repository with the newly created Oracle Internet Directory.

The Installer installs the components in the proper order when you choose to install both OracleAS Metadata Repository and the Identity Management components on the same computer.

OracleAS Infrastructure Installation: Overview

The installation of OracleAS Infrastructure involves the following steps:

- **Preinstallation tasks:**
 - Check Metalink, installation guide, and release notes.
 - Check the requirements.
 - Create operating system users and groups as required.
- **Installation:**
 - Select the installation type and components to configure.
 - Perform the postinstallation tasks and checks.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Infrastructure Installation: Overview

The installation of OracleAS Infrastructure involves the following steps:

- Preinstallation tasks:
 - For details about the installation, refer to the *Oracle Application Server Installation Guide* and the *Release Notes*.
 - Check whether the system has the required resources in terms of CPU, memory, and disk space.
 - Verify the version of the operating system and verify that the necessary patches and packages are present. Designate a disk location for installing OracleAS Infrastructure.
 - Ensure that the user who is performing the installation has sufficient access rights. On platforms such as UNIX and Linux, some scripts are required to be run as superuser during the installation. You can either grant the installing user the superuser (root) privilege or ensure that the user with the superuser (root) privilege is available to run such scripts. On Windows platforms, you should install OracleAS Infrastructure as a user with Administrator privileges.

Minimum Requirements for OracleAS Infrastructure

CPU	Pentium (32 bit): 450 MHz	Checked by the Installer
Disk	OracleAS Infrastructure: 3.7 GB OracleAS J2EE and Web Cache: 0.9 GB OracleAS Portal and Wireless: 1.2 GB OracleAS Business Intelligence and Forms: 2 GB	Not checked by the Installer
Memory	OracleAS Infrastructure (complete): 1024 MB Oracle Identity Management only: 512 MB OracleAS Metadata Repository only: 750 MB OracleAS J2EE and Web Cache: 512 MB OracleAS Portal and Wireless: 1024 MB OracleAS Business Intelligence and Forms: 1024 MB	Checked by the Installer
Temporary	(Defined by TMP or TMPDIR variable) 400 MB	Checked by the Installer
Swap/Page	Swap 1.5 GB	Checked by the Installer
Monitor	256 color	Checked by the Installer
Operating system	RHAS 2.1, RHEL 3.0, SuSE Server 8, SuSE Server 9, RHAS 4.0	Checked by the Installer

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Infrastructure Requirements

The requirements in the slide relate to the installation of OracleAS Infrastructure. For more information, refer to the *Oracle Application Server Installation Guide* for the operating system you are using.

- In a Linux system, you can check the processor details from the `/proc/cpuinfo` file and the operating system from the `/etc/issue` file:


```
# cat /proc/cpuinfo | grep -i name
model name      : Intel(R) Pentium(R) 4 CPU 1.70GHz
# cat /etc/issue
Red Hat Enterprise Linux AS release 3 (Taroon Update 3)
Kernel \r or an \m
```
- You can get the kernel version by using the `rpm` command:


```
# rpm -qa | grep kernel
kernel-2.4.21-20.EL
kernel-pcmcia-cs-3.1.31-13
kernel-utils-2.4-8.37.6
```

OracleAS Infrastructure Requirements (continued)

- You can get the memory and the swap configuration by using the free command:

```
# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/hda7                  14G       7.6G   5.3G  59% /
/dev/hda5                   23M       6.0M    16M  28% /boot
none                      502M         0   502M   0% /dev/shm
/dev/hda1                 1004M     567M   438M  57% /mnt/cdrive
144.25.69.70:/vol/vol1/relbuilder/LNX/linux/ias
                        286G     260G    27G  91%
/modules/stage/AS1012
```

You can get the disk space usage or the disk space availability by using the df command.

```
# df -m (to get values in MBs)
```

```
Filesystem      1M-blocks      Used Available Use% Mounted
on
/dev/hda6       14384        5271      8383  39% /
none           502          0        501   0% /dev/shm
/dev/hda1       1004         696       308  70%
```

Setting Up the Environment

You must:

- **Configure kernel parameters at the operating system level**
- **Set up the following environment variables:**
 - `TMP`
 - `DISPLAY`
 - `ORACLE_HOME`, `ORACLE_SID` (unset these)
- **Set up the `/etc/hosts` file**
- **Verify that the default port for the metadata repository listener is 1521**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Setting Up the Environment

You may also have to set some parameters at the operating system level. The following is a list of the essential parameters for the Linux Red Hat Advanced Server operating system:

- The current settings of semaphores are stored in the `/proc/sys/kernel/sem` file. The order and values of the semaphore variables are:
 - `semmsl 100, semmns 32000, semopm 100, and semmni 100`
 - You can use the `cat` command to verify the values:

```
# cat /proc/sys/kernel/sem
256      32000   100      142
```
 - To alter the values as required, you can use the `echo` command:

```
# echo 100 32000 100 100 > /proc/sys/kernel/sem
# cat /proc/sys/kernel/sem
100      32000   100      100
```
- The current settings for the shared memory parameters `shmmax`, `shmmni`, and `shmall` are stored in their respective files under the `/proc/sys/kernel` directory. You can use the `cat` command to view the value and the `echo` command to reset the values.
- Set the maximum file handles for a process to 131072:

```
# echo 131072 > /proc/sys/fs/file-max
# ulimit -n 131072
```

Oracle Application Server 10g R2: Administration I 3-9

Setting Up the Environment (continued)

- Open the port range between 1024 and 65000 for access, by setting the port range as follows:

```
# echo 1024 65000 > /proc/sys/net/ipv4/ip_local_port_range
```
- Set the maximum number of processes spawned by a user to 16384 by using the `ulimit` command:

```
# ulimit -u 16384
```
- The kernel parameters set in this step are transient, and do not survive a reboot of the system. To make these values permanent, you should incorporate these parameters in the `/etc/sysctl.conf` file.

Setting Up the Environment Variables

- **TMP:** The temporary space is used during the installation process for expanding and configuring the installable modules. Set this variable to point to a directory that has at least 1 GB of free space, and also ensure that the user who is installing has the write privilege.
- **DISPLAY:** This variable is used by the Installer on UNIX and Linux platforms to direct the user interface prompts and responses. If you are running the Installer remotely from another workstation, then set `DISPLAY` to the system name or IP address of your local workstation where you launch the Installer.
 - For example, if you are working from a workstation with IP address 123.45.67.89, then set the `DISPLAY` as follows:

```
DISPLAY=123.45.67.89:0.0; export DISPLAY
```
- **ORACLE_HOME and ORACLE_SID:** `ORACLE_HOME` is an environment variable that indicates the directory in which the Oracle Application Server software is installed. The `ORACLE_SID` variable points to the database to which you normally log in. If you have other Oracle products installed on the computer where you plan to install Oracle Application Server, then you may have set `ORACLE_HOME` and `ORACLE_SID`.
To avoid confusion and problems in the installation of OracleAS Infrastructure, you should remove the settings for `ORACLE_HOME` and `ORACLE_SID` variables.

Setting Up the Hosts File

- In UNIX and Linux systems, the `/etc/hosts` file (in Windows, the `%SYSTEMROOT%\system32\drivers\etc\hosts` file) describes the host name and IP address of the system. Ensure that the format that is used in this file is consistent with the following:

```
<IP_ADDRESS> <DOMAIN_QUALIFIED_HOSTNAME> <ALIASES>
```

 - For example:

```
123.123.123.123 myappsrv.mycompany.com myappsrv
```

Database Listener Port

- The database installed with OracleAS Infrastructure uses port 1521 by default. If you have another application on the host using port 1521, you can use the `staticports.ini` file to suitably assign the ports for the Infrastructure that you are installing. For more information, refer to the *Oracle Application Server Installation Guide*.

OracleAS Infrastructure: Installation Steps

1. Welcome
2. Inventory Location
3. File Location
4. Product to Install
5. Installation Type
6. Configuration Option
7. Identity Management Realm
8. Certificate Authority
9. Database Identification
10. Infrastructure Instance

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Infrastructure: Installation Steps

1. In the Welcome window, review the information about the Installer. Click Next to proceed with the installation.
2. Verify the location of the inventory directory for installation files.
3. In the File Locations window, verify the destination name and destination path.
OracleAS Infrastructure must be installed in a separate Oracle home, preferably on a separate host from any Oracle Application Server installations.
4. In the Select a Product to Install window, select OracleAS Infrastructure. OracleAS Infrastructure installs the OracleAS Metadata Repository and Identity Management components.
5. Select the product type that you want to install: Metadata Repository or Identity Management components, or both.
6. In the Select Configuration Options window, you can select the components that you want to configure. You can choose to configure Oracle Internet Directory, OracleAS Single Sign-On, Oracle Delegated Administration Services, Oracle Directory Integration and Provisioning, and Oracle Application Server Certificate Authority.

OracleAS Infrastructure: Installation Steps (continued)

7. Select the Identity Management Realm.
8. Provide the Oracle Application Server Certificate Authority details.
9. Provide the Database Identification details.
10. Provide the OracleAS Infrastructure instance details.
11. Verify the summary.

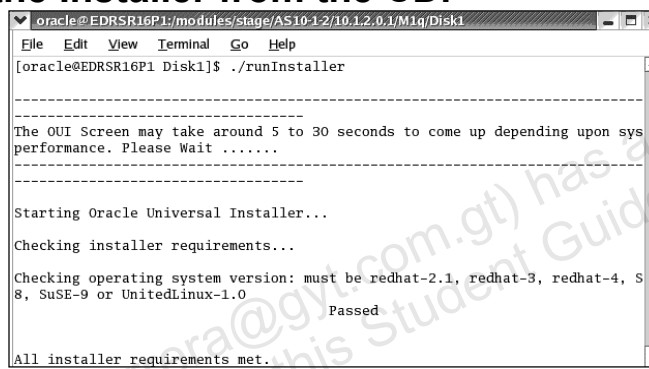
The installation proceeds in two phases. Initially, the necessary product files are copied and extracted. Then, the configuration assistants are run. Before invoking the configuration assistants, you are required to run the `root.sh` script as the superuser to create necessary information for the Database Configuration Assistant.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Starting the Installation

To start your installation in UNIX or Linux systems, perform the following steps:

1. Mount the installation CD-ROM drive.
2. Insert your Oracle Application Server 10g Release 2 CD into the drive.
3. Run the Installer from the CD.



```
oracle@EDRSR16P1/modules/stage/AS10-1-2/10.1.2.0.1/M1g/Disk1
File Edit View Terminal Go Help
[oracle@EDRSR16P1 Disk1]$ ./runInstaller

-----
The OUI Screen may take around 5 to 30 seconds to come up depending upon sys
performance. Please Wait .....
-----

Starting Oracle Universal Installer...

Checking installer requirements...

Checking operating system version: must be redhat-2.1, redhat-3, redhat-4, S
8, SuSE-9 or UnitedLinux-1.0
                                     Passed

All installer requirements met.
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting the Installation

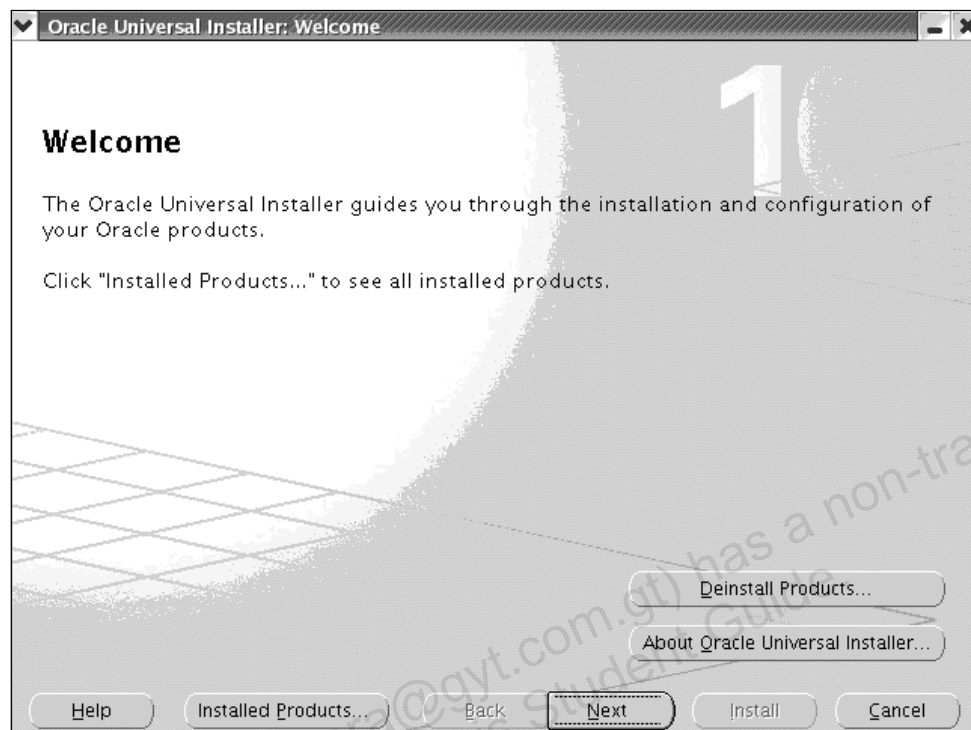
To launch the Installer and install Oracle Application Server, perform the following steps:

1. In a Linux or UNIX environment, mount the drive that you would use to install Oracle Application Server, if it is not already mounted.
2. Insert your Oracle Application Server 10g Release 2 CD into the drive. In Windows systems, the Installer is invoked automatically if Autorun is enabled.
3. In Linux or UNIX systems, invoke the Installer by using the runInstaller script.

When you invoke the Installer, it runs a prerequisites check, and notifies you whether the verification has passed or failed.

Then, Oracle Universal Installer is invoked.

Oracle Universal Installer



Copyright © 2005, Oracle. All rights reserved.

Oracle Universal Installer

Oracle Application Server uses Oracle Universal Installer to install and configure components. The Installer guides you through each step of the installation process, so that you can choose configuration options for a customized product.

The Installer includes features that perform the following tasks:

- Exploring and providing installation options for products
- Detecting preset environment variables and configuration settings
- Setting environment variables and configuration during installation
- Uninstalling products

The Installer creates the `oraInventory` directory the first time it is run on your machine. The `oraInventory` directory keeps an inventory of products that the Installer installs on your machine, as well as other installation information. If you have previously installed Oracle products, then you may already have an `oraInventory` directory.

The latest log file can be obtained from the `oraInventory_location/logs` directory. Log file names take the form `installActions<datetime>.log`. Do not delete or manually alter the `oraInventory` directory or its contents. Doing so can prevent the Installer from locating the products that you have installed on your system.

First Installation of an Oracle Product

Specify Inventory directory and credentials

You are starting your first installation on this host. As part of this install, you need to specify a directory for installer files. This is called the "inventory directory". Within the inventory directory, the installer automatically sets up subdirectories for each product to contain inventory data and will consume typically 150 Kilobytes per product.

Enter the full path of the inventory directory:

You can specify an Operating System group that has write permission to the above inventory directory. You can leave the field blank if you want to perform the above operations as a Superuser.

Specify Operating System group name:

Copyright © 2005, Oracle. All rights reserved.

First Installation of an Oracle Product

If OracleAS Infrastructure is the first Oracle product that is to be installed on your computer, then the Installer displays a window in which you specify an inventory location. You are not prompted for the inventory location after the first installation. The inventory location directory has the following attributes:

- It contains the permanent and per-product component files in subdirectories.
- Any user installing or updating Oracle products on the computer must be able to write to it.

If the different Oracle product installations need to be managed separately, keep the inventory location in a common place so that the other users in the operating system group have access when they install or update Oracle products.

If you have installed an Oracle product previously on the computer, the Installer uses the existing inventory location. Ensure that you have write permissions on that directory.

Group Window on UNIX and Linux

When creating the inventory location on UNIX or Linux systems, the Installer invokes the group window in which you specify the operating system group that performs the installation or update. The Installer then prompts you to run a script as the `root` user to ensure that the permissions to the inventory location is granted to the group.

Specify File Locations Window

Specify File Locations

Source

Enter the full path of the file representing the product(s) you want to install:

Path:

Destination

Enter or select a name for the installation and the full path where you want to install the product.

Name:

Path:

Copyright © 2005, Oracle. All rights reserved.

Specify File Locations Window

This window enables you to provide the Oracle home details for the product that you are installing. You provide the following:

- Name, using which this installation will be identified in the Inventory. The name cannot contain spaces, and has a maximum length of 16 characters; for example, `infra`.
- Path, where you enter the full path for the OracleAS Infrastructure executables and configuration files. If the directory does not exist, the Installer creates it. If you want to create the directory in advance, create it as the user installing the software; do not create it as the root user. The Oracle home path cannot contain environment variables or spaces.

Select a Product to Install



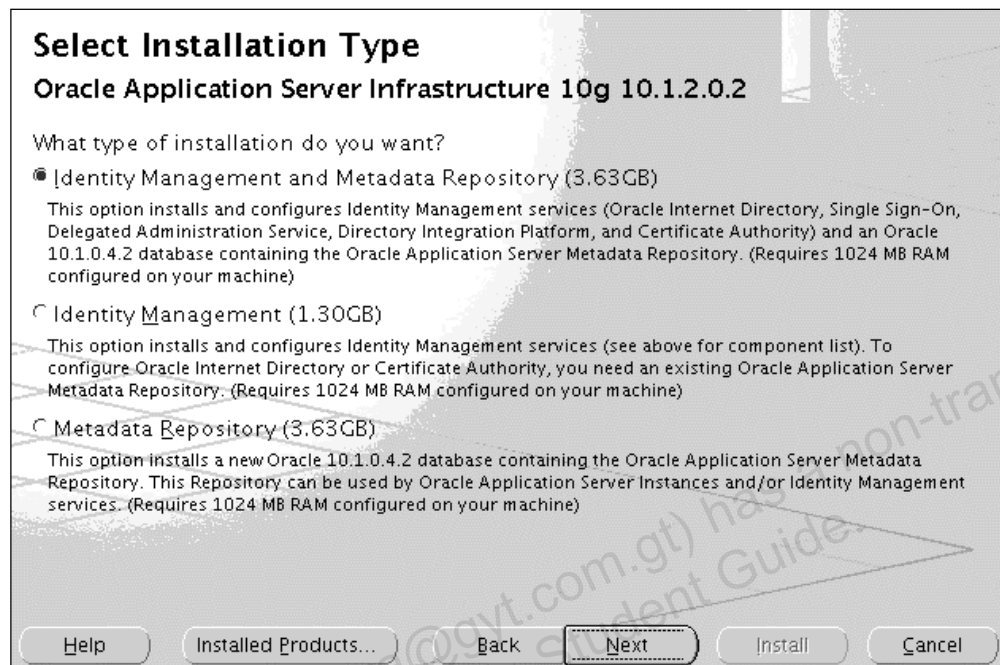
Copyright © 2005, Oracle. All rights reserved.

Select a Product to Install

Select which part of Oracle Application Server you want to install. Select "Oracle Application Server Infrastructure 10g 10.1.2.0.2" to install OracleAS Infrastructure.

By default, the Installer installs Oracle Application Server with text in English and in the operating system language. Click Product Languages to select additional languages to install. The Installer installs the text in the selected languages and also installs fonts required to display the languages. It is important that you select all the languages that you need during installation, because you cannot add or remove languages after installation.

Select Installation Type



Copyright © 2005, Oracle. All rights reserved.

Select Installation Type

You should select the installation type that you want based on the following:

- **Identity Management and OracleAS Metadata Repository**
 - Select this option to install Identity Management services and a new Oracle database that contains OracleAS Metadata Repository on this host.
 - Do not select this option if you want to use an existing Oracle database that contains OracleAS Metadata Repository, or if you want to install the database and the Identity Management services on separate hosts.
- **Identity Management**
 - If you select this option to install OracleAS Single Sign-On or Oracle Application Server Certificate Authority, then you need an existing OracleAS Metadata Repository.
- **OracleAS Metadata Repository**
 - Select this option to install a new Oracle database that contains OracleAS Metadata Repository. The repository can then be used by Oracle Application Server instances and Identity Management services.
 - When you install only OracleAS Metadata Repository, Application Server Control is not installed. You can use Database Control or Oracle9i Database Studio to manage OracleAS Metadata Repository Database.

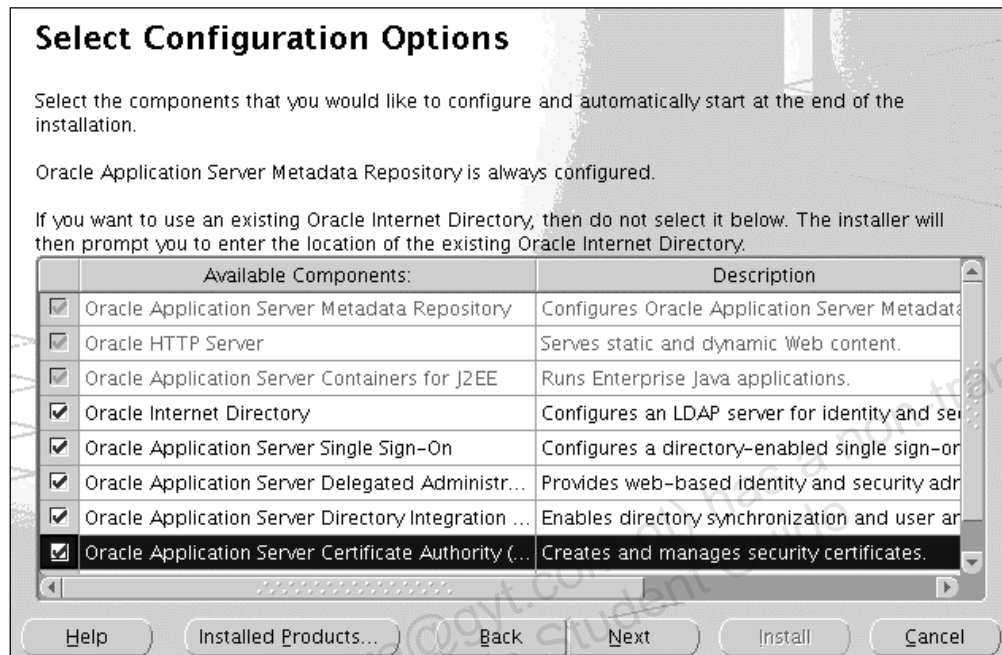
Oracle Application Server 10g R2: Administration I 3-18

Select Installation Type (continued)

- To install the repository in an existing database, run the OracleAS Metadata Repository Creation Assistant that is available on the “OracleAS Metadata Repository Creation Assistant” CD.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Select Configuration Options



Copyright © 2005, Oracle. All rights reserved.

Select Configuration Options

You can select the OracleAS Infrastructure components that you want the Installer to configure and start after the installation. OracleAS Metadata Repository, Oracle HTTP Server, and Oracle Application Server Containers for J2EE components will be configured always. When you perform an installation of both the OracleAS Metadata Repository and Identity Management components, the following components are selected by default:

- Oracle Internet Directory
- OracleAS Single Sign-On
- OracleAS Delegated Administration Service
- OracleAS Directory Integration and Provisioning

You can select Oracle Application Server Certificate Authority if you intend to use the component. As already mentioned, irrespective of your selection in this window, all the components will be installed. However, the configuration assistants relating to the components not selected in this window will not be run after installation.

If you decide to use that component at a later time, you must manually launch the configuration assistant to configure that component. You can configure components after installation by using the Configure Component page in the Application Server Control Console.

Select Configuration Options (continued)

Oracle Application Server is designed to provide a wide variety of high-availability solutions, ranging from load balancing and basic clustering to providing maximum system availability during catastrophic hardware and software failures. Select High Availability and Replication to view new high availability–related windows during installation.

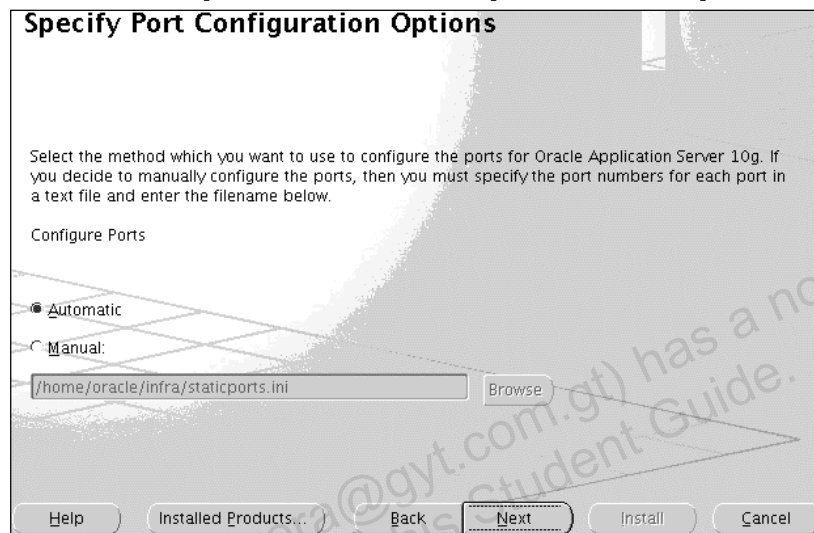
Oracle Application Server failure protection can be implemented by:

- Deploying all Oracle Application Server components in a redundant fashion to make their services more available
- Load balancing, intelligent routing, and crash prevention
- Death-detection and out-restarts
- Backing up and restoring data
- Disaster recovery solution
- Planned down-time protection

For additional information about high availability, refer to the *Oracle Application Server High Availability Guide*.

Specifying Port Configuration Options

This is to enable the Oracle Application Server 10g Release 2 components to use predefined ports.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specifying Port Configuration Options

Select whether you want to use default port numbers for components or custom port numbers as specified in a port configuration file (the `staticports.ini` file).

Automatic

- Select this option to assign default ports to components. For a list of default ports, see the *Oracle Application Server Installation Guide*.

Manual

- Select this option if you have already created a port configuration file that specifies the port numbers that you want to use for each component. Enter the full path for this file in the field provided.
- This port configuration file is typically referred to as the `staticports.ini` file, but the name does not matter. The Installer reads this file and assigns the specified ports to the components.

Specify Namespace in Internet Directory

Specify Namespace in Internet Directory

Specify a location, or namespace, in Oracle Internet Directory to contain users, groups, and Identity Management policies. This namespace will be the default Identity Management Realm.

☒ Suggested Namespace:

☐ Custom Namespace:

Example: dc=acme,dc=com

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

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specify Namespace in Internet Directory

An Identity Management Realm in Oracle Internet Directory contains management policies for all users and groups in Oracle Internet Directory. You specify the root-level location of the default realm. This realm will be created when Oracle Internet Directory is installed.

Suggested Namespace

- The default location is derived from the DNS domain name of the host where the Installer installs Oracle Internet Directory. For example, if the host name is `myhost.acme.com`, then the root location of the default Identity Management Realm would be `dc=acme,dc=com`.
- The Installer creates the directory tree corresponding to the default location. It also creates two subcontainers (`cn=users` and `cn=groups`) under the root location. You can add users and groups to these subcontainers.
- The Installer also creates default naming, authentication, and authorization policies in the realm. You can customize these policies after installation.

Custom Namespace

- If the default location does not meet your deployment needs, use this field to specify an alternative root location (using a distinguished name, or DN) for your default Identity Management Realm.

Oracle Application Server Certificate Authority

Specify OCA Distinguished Name

To issue digital certificates Oracle Application Server Certificate Authority(OCA) must have a unique identifier in the form of a Distinguished Name(DN) based on the X.500 standard.

Enter the DN you want OCA to use. You can enter a typical set of DN components, or a custom DN.

☒ Typical DN:

Common Name (CN):

Organizational Unit (OU):

Organization (O)*:

Country (C):

☐ Custom DN:

Example DN: CN=Acme Certificate Authority,OU=IT Division,O=Acme Corporation,C=US

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Certificate Authority

You configure Oracle Application Server Certificate Authority to provide digital certificates. The Certificate Issuing Authority must have a distinguished name. You can use the typical distinguished name (DN), in which case the DN from the Identity Management Realm you entered is used.

You have to enter value in the Organization (O) field for the certificate authority.

In the next window, you enter the key length for encryption. The longer the key, the higher is the security; but it will take longer for the certificate issuance. Generally, a key length of 2048 would be adequate.

Specify Database Configuration Options

Specify Database Configuration Options

Database Naming
A Global Database Name, typically of the form "name.domain", uniquely identifies an Oracle database. In addition, each database is referenced by at least one Oracle System Identifier (SID). Specify the Global Database Name and SID for this database.

Global Database Name: SID:

Database Character Set
The number of language groups to be stored determine which database character set to use. See "Help" for the definition of language groups. For the Unicode database character set, select "Unicode Standard UTF-8 AL32UTF8"

Select Database Character set:

Database File Location
Use the file system for database storage. For best database organization and performance, Oracle recommends installing database files and Oracle software on separate disks.

Specify Database File Location:

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specify Database Configuration Options

Database Naming

The Specify Database Configuration Options window is used in creating the Oracle database to store the metadata repository. The identifiers that are specified here are used for managing and connecting to the repository. The global database name should be unique across the network.

Database Character Set

Decide which language groups you want to support in the database, and choose an appropriate character set. Choose:

- WE8ISO8859P15, WE8MSWIN1252, or Unicode (AL32UTF8 or UTF8) to support the Euro symbol
- The AL32UTF8 Unicode character set, if you need to use characters in the NE8ISO8859P10 or CEL8ISO8859P14 character sets
- The default character set

Database File Location

Enter the full path to the parent directory for the data files directory. The Installer will create a subdirectory with the same name as the SID, and place your data files in this subdirectory.

Specify Database Schema Passwords

Specify Database Schema Passwords

The Starter Database contains pre-loaded schemas, most of which have passwords that will expire and be locked at the end of installation. After the installation is complete, you must unlock and set new passwords for those accounts you wish to use. Schemas used for the database management and post-install functions are left unlocked, and passwords for these accounts will not expire. Specify the passwords for these accounts.

☐ Use different passwords for these accounts

User Name	Enter Password	Confirm Password
SYS		
SYSTEM		
SYSMAN		
DBSNMP		

☒ Use the same password for all the accounts

Enter Password: Confirm Password:

Help Installed Products... Back Next Install Cancel

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specify Database Schema Passwords

You must specify passwords in the Specify Database Schema Passwords window for the following database administrative accounts:

- SYS
- SYSTEM
- SYSMAN
- DBSNMP

You can use the same password for all the accounts, or specify different passwords for each account.

Specify Instance Details

Specify Instance Name and ias_admin Password

All Oracle Application Server Infrastructure instances installed on a host must have unique names. The hostname and domain name of the host are appended to the instance name.

Each Oracle Application Server Infrastructure instance has its own password, regardless of which user performed the installation. Passwords are not shared across instances, even if the instances were installed by the same user.

The password must have a minimum of 5 alphanumeric characters, maximum 30 characters, and at least one of the characters must be a number.

Administrator Username: ias_admin

Instance Name:

ias_admin Password:

Confirm Password:

ORACLE

Copyright © 2005, Oracle. All rights reserved.

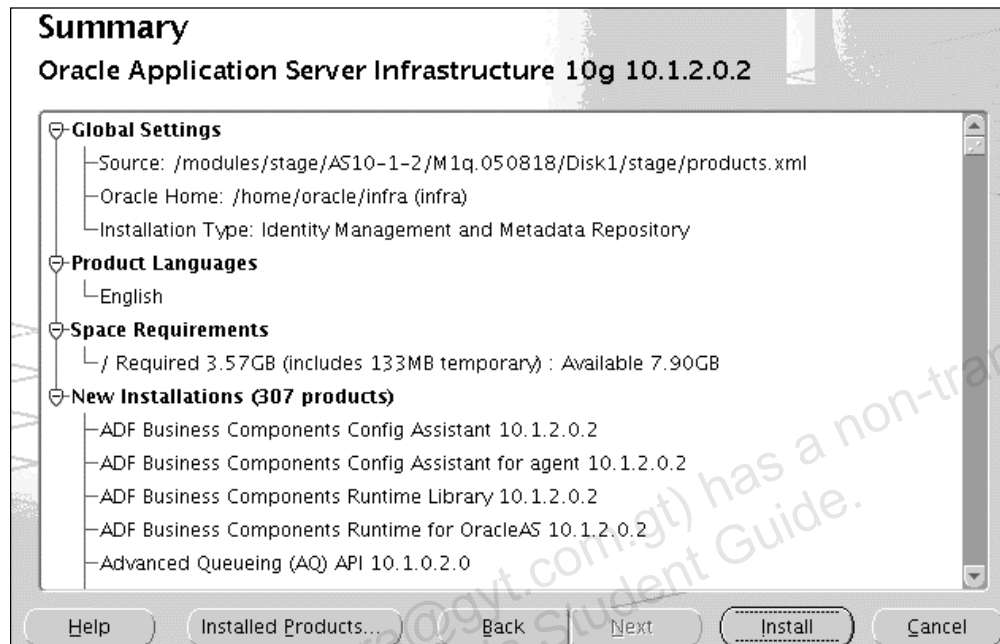
Specify Instance Details

Each Oracle Application Server instance on a machine has a unique name. This name is essential for routing requests and administrative purposes as well.

Also ias_admin, which is the administrative user for each instance, has its own password regardless of who performed the installation.

This password is used by other administrative users, such as the Portal administrator (portal_admin), and Oracle Internet Directory administrator (orcladmin). This password is also used by the ODS schema account.

Summary of Installation



Summary of Installation

The Installer provides the summary of options that you have chosen. Verify the details in this window, and click Install to begin the installation. The Installer performs three installation actions:

- Copies files
- Links the executables
- Sets up the configuration

Then, it invokes the configuration assistants.

Before invoking the configuration assistants, you should run the `root.sh` script as superuser from another window or terminal to set up database entries.

The entries are used for two main purposes:

- Automating the startup of databases
- Creating entries to enable Oracle Net Configuration Assistant and Database Configuration Assistant to run

End of Installation Window



End of Installation Window

The End of Installation window appears at the end of the installation process. It notifies you whether the installation is successful or unsuccessful.

In this window, note the following information that you will need in order to manage OracleAS Infrastructure:

- The URL to access Oracle HTTP Server and the Welcome page
- The URL to access Application Server Control

Postinstallation Tasks

- **Set the ORACLE_HOME and ORACLE_SID variables.**
- **Include \$ORACLE_HOME/bin in your \$PATH.**
- **Verify the status of the following:**
 - **Infrastructure database and its listener**
 - **OracleAS Infrastructure instance and components**
 - **Application Server Control**
- **Note the port assignments for your installation.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Postinstallation Tasks

Setting the Environment Variables

- After you have installed OracleAS Infrastructure, you should set the ORACLE_HOME and ORACLE_SID environment variables for ease of managing OracleAS Infrastructure. The following examples show the setting using Korn or bash shell (SID is infra, and ORACLE_HOME is /oracle/oraias/infra):

```
export ORACLE_SID=infra
export ORACLE_HOME=/oracle/oraias/infra
export PATH=$PATH:$ORACLE_HOME/bin
```
- You can include the commands above in the .login file of the operating system user installing and managing OracleAS Infrastructure.

Verifying the Status of the Database Listener

- To verify that the database listener is operational, you can use the lsnrctl command. In the following example, it is presumed that you are using the default database listener:

```
$ORACLE_HOME/bin/lsnrctl status | grep status
...
Instance "infra", status READY, has 3 handler(s) for this
service...
```

Accessing the OracleAS Infrastructure Instance

Welcome

to Oracle Application Server 10g Release 2 (10.1.2)

Overview



Oracle Application Server 10g Release 2 (10.1.2) is an integrated, standards-based application platform suite that allows organizations of all sizes to respond better to changing business requirements.

The Oracle Application Server application platform suite can improve your organization's ability to predict and respond to market dynamics, enhance productivity, and simplify your information technology environment, all while allowing you to use your existing investments to their full potential. Oracle Application Server 10g Release 2 (10.1.2) achieves these goals through:

- **Service-Oriented Computing:** Oracle Application Server uses a service-oriented computing architecture to facilitate the development of enterprise applications as business services, which enables you to build a flexible enterprise application infrastructure.
- **Grid Computing:** The Oracle Application Server architecture coordinates the use of large numbers of low cost, modular servers and storage to act as one large computer to run your enterprise applications. This allows you to start small, minimize unused resources, and add processing or storage capacity as you need it.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Accessing the OracleAS Infrastructure Instance

Your installation of OracleAS Infrastructure is successful if you are able to do the following:

- You can log in by using the URL `http://<your-server>:<HTTP-port>` to access the Welcome page of your Oracle HTTP Server. You can get the HTTP port from the `portlist.ini` file in your `ORACLE_HOME/install/` directory.
- Using the Enterprise Manager link on your HTTP Server page, you can access Oracle Enterprise Manager 10g Application Server Control Console. You will be prompted for the `ias_admin` username and password.
- From Application Server Control, you can drill down to Oracle Application Server instances and view the status of the instance and its components.

Application Server Control

Enter username and password for "enterprise-manager" at edrsr16p1:1156

User Name:

Password:

☐ Use Password Manager to remember these values.

OK Cancel

Farm: infra.us.oracle.com

Instances can be grouped and managed together by configuring standalone instances in a Oracle Application Server Farm.

Repository Type **Database**

Clusters

Select Name

There are no clusters in the farm.

Standalone Instances

These instances belong to the farm but are not part of any cluster.

[Join Cluster](#)

Select Name

 [infra.edrsr16p1](#)

Host

edrsr16p1

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Application Server Control

You can access the Application Server Control of your OracleAS Infrastructure by using the URL `http://<host>:<em-port>`. For example, in the slide, it is accessed by using the following URL:

`http://edcdr5p1.us.oracle.com:1810`

When you install OracleAS Infrastructure, a farm is also initiated and the farm page becomes the entry point for Application Server Control.


You can drill down to the OracleAS Infrastructure instance and monitor and administer the components of that instance.

Verifying Oracle Internet Directory Server

Farm > Application Server: infra.edrsr16p1 >

Oracle Internet Directory

Page Refreshed

General

Status **Up**
Version **10.1.2.1.0**
Repository **EDRSR16P1:1521:infra**
CPU Usage (%) **0.021**
Memory Usage (MB) **10.09**

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

Status

[Security Alerts](#)
[LDAP Metrics](#)
[Directory Integration](#)
[Directory Replication](#)

Directory Server Instances

[Restart](#) [Stop](#) [View Load](#) [View Operations](#) [Start New Instance](#)

Select	Instance Number	Host Name	Port Number	Start Time	Downtime Count	Configuration Set Number	Status
	1	edrsr16p1	389	2005-08-24 02:44:50	0		 

ORACLE

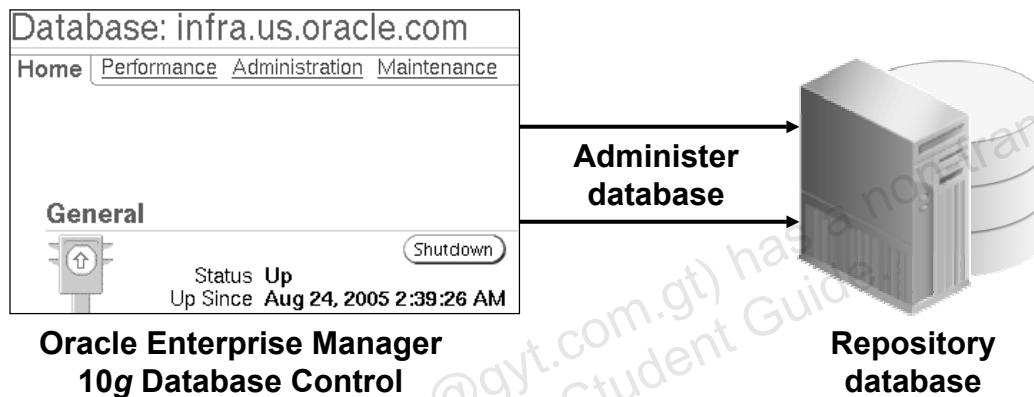
Copyright © 2005, Oracle. All rights reserved.

Verifying the Oracle Internet Directory Server

From the OracleAS Infrastructure Home page, you can drill down to the Oracle Internet Directory Server Home page to verify the details of the Oracle Internet Directory server. The Directory Server Instances section describes the details of the Oracle Internet Directory server including which port the Oracle Internet Directory server is running on. This information will be required when you install the middle-tier components that use OracleAS Infrastructure.

Oracle Enterprise Manager 10g Database Control

You can administer the OracleAS Metadata Repository database by using Oracle Enterprise Manager 10g Database Control.



Copyright © 2005, Oracle. All rights reserved.

Oracle Enterprise Manager 10g Database Control

You can manage your OracleAS Metadata Repository database by using Oracle Enterprise Manager 10g Database Control. Oracle Enterprise Manager 10g Database Control is installed and configured when you install OracleAS Metadata Repository.

Note: If you install OracleAS Metadata Repository in an existing database, then Database Control is also available. But this depends on whether:

- The existing database is Oracle Database 10g
- Database Control is configured when the database is created

The Database Control Console provides a Web-based user interface similar to that provided by the Application Server Control Console, but is designed to help you manage your Oracle database. You can perform database management tasks, such as monitoring the performance of the database, scheduling backups, and managing the database tablespaces.

Managing OracleAS Metadata Repository Database

Login to Database:infra.us.oracle.com

* User Name

sys

* Password

Connect As

SYSDBA

Login

Availability (%)

100

(Last 24 hours)

Instance Name

infra

Version

10.1.0.4.2

Read Only

No

Oracle Home

/home/oracle/infra

Listener

LISTENER_EDRSR16P1

Host

EDRSR16P1

CPU

Other

infra

Run Queue

0.93

Paging (pages per second)

0.0

High Availability

Instance Recovery Time (seconds)

16

Last Backup

n/a

Archiving

Disabled

Archive Area Used (%)

n/a

Flashback Logging

Disabled

Space Usage

Database Size (GB)

2

Problem Tablespaces

✓ 0

Segment Findings

Not Configured

Policy Violations

✓ 0

Dump Area Used (%)

✓ 48

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Managing OracleAS Metadata Repository Database

To display Database Control to manage the OracleAS Metadata Repository database, perform the following tasks:

1. Access the Database Control URL:
<http://<host name.domain>:<port>/em>
<http://myhost.com:5500/em>
 where:
 “host name” is the name of the computer on which you installed the Oracle database
 “port” is the port number reserved for Database Control during installation
 You can verify the correct port number in `portlist.ini` stored in the `install` directory of the OracleAS Metadata Repository Database Oracle home. The installation reserves the first available port in the range 5500–5519 for Database Control.
2. Log in to the database from the Database Control login page. Use SYS as the username and connect as SYSDBA. Use the password that you specified for the SYS account during the installation.
3. Database Control displays the Database Home page. You can review the current state of your infrastructure database, and monitor and administer your infrastructure database from the Database Home page.

Note: For information about database management with the Database Control Console, refer to “Oracle 2 Day DBA” in the Oracle Database 10g documentation library.

Oracle Application Server 10g R2: Administration I 3-35

Unauthorized reproduction or distribution prohibited. Copyright© 2009, Oracle and/or its affiliates.

Accessing the SSO Server



Accessing the SSO Server

You can access and administer the SSO server as follows:

1. Invoke the Application Server Control and navigate to the Infrastructure instance page.
2. Click the Single Sign-On:orasso link in the Systems Components Table. Verify that the SSO component is active.
3. Click Administer via Single Sign-On Web Application link under Related Links section of the Single Sign-On:orasso page.
4. Click the Login link on the SSO Server Home page.
5. Enter orcladmin as the username and the password for administrative users (welcome1) that you entered during the OracleAS Infrastructure installation.

The SSO Server Administration link appears on your SSO Server Home page.

Starting and Stopping OracleAS Infrastructure

- To start OracleAS Infrastructure, start the components in the following order:
 1. Start the database listener.
 2. Start the metadata repository database.
 3. Start OracleAS Infrastructure instance processes.
 4. Start Application Server Control.
- To stop OracleAS Infrastructure, stop the components in the following order:
 1. Stop Application Server Control.
 2. Stop OracleAS Infrastructure instance processes.
 3. Stop the metadata repository database.
 4. Stop the database listener.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting and Stopping OracleAS Infrastructure

To start OracleAS Infrastructure, start the components in the following order:

1. Start the database listener:

```
$ORACLE_HOME/bin/lsnrctl start
```
2. Start the repository database:

```
$ORACLE_SID=infra; EXPORT ORACLE_SID
$ORACLE_HOME/bin/sqlplus /nolog
sql> connect sys/password_for_sys as sysdba
sql> startup
sql> exit
```
3. Start the processes of the OracleAS Infrastructure instance:

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```
4. Start OracleAS Console:

```
$ORACLE_HOME/bin/emctl start iasconsole
```

To stop OracleAS Infrastructure, stop the components in the reverse order.

Note: You can also start an entire application server instance, which includes the infrastructure and the middle tier, by using the `runstartupconsole` command:

```
$ORACLE_HOME/bin/runstartupconsole.sh start all
```

Summary

In this lesson, you should have learned how to:

- **Define the installation requirements for OracleAS Infrastructure**
- **Describe OracleAS Infrastructure installation types**
- **Install OracleAS Infrastructure**
- **Start and stop OracleAS Infrastructure**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

4

Installing OracleAS Middle Tier

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Describe the Oracle Application Server 10g Middle Tier installation types and their requirements**
- **Perform preinstallation tasks**
- **Install the middle tier with the Portal and Wireless installation type**
- **Access the installed OracleAS Middle Tier components**
- **Upgrade 10.1.2.0.2 Portal to 10.1.4 Portal**
- **Verify the completion of the installation**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Middle Tier Installation

Phases: Overview

1. Preinstallation

- a. Check the requirements.
- b. Create the required operating system users and groups.
- c. Set up the environment.

2. Installation

- a. Select an installation type and components to configure.
- b. Provide the required information to connect to OracleAS Infrastructure.

3. Postinstallation

- a. Access the component Web pages.
- b. Verify the installation.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Middle Tier Installation Phases: Overview

Preinstallation

During the first phase of installation, you perform the following tasks:

- Create Linux or UNIX accounts and groups.
- Perform component-specific preinstallation tasks on the middle tier and the origin database where you stored your application data. Do not perform these tasks on OracleAS Infrastructure.

Installation

During the second phase of installation, the Installer guides you through the installation steps that include selecting an installation type, defining connect information to the Metadata Repository, and selecting the components that you want to configure automatically and start at the end of the installation.

Postinstallation

During the final phase of the installation process, verify the installation by accessing the middle-tier component Web pages and checking the component status by using Application Server Control.

Preinstallation: OracleAS Middle Tier Requirements

CPU	Pentium (32 bit): 450 MHz	Checked by the Installer
Disk	OracleAS Infrastructure: 2.6 GB OracleAS J2EE and Web Cache: 0.5 GB OracleAS Portal and Wireless: 1.1 GB	Not checked by the Installer
Memory	OracleAS Infrastructure (complete): 1024 MB Oracle Identity Management only: 512 MB OracleAS Metadata Repository only: 750 MB OracleAS J2EE and Web Cache: 512 MB OracleAS Portal and Wireless: 1024 MB	Checked by the Installer
Temporary	(Defined by TMP or TMPDIR variable) 400 MB	Checked by the Installer
Swap/Page	Swap 1.5 GB	Checked by the Installer
Monitor	256 color	Checked by the Installer
Operating system	RHAS 2.1, RHEL 3.0, SuSE Server 8, SuSE Server 9	Checked by the Installer

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Preinstallation: Checking OracleAS Middle Tier Requirements

The requirements that are listed in the table in the slide relate to the OracleAS Portal and Wireless installation type. For more information, refer to the *Oracle Application Server Installation Guide*.

Preinstallation: Setting Up the Environment

The following must be verified before starting the Installer:

- The **DISPLAY** environment variable is set:
 - This variable enables you to run the Installer remotely.
- The operating system user installing should have permission to write to the inventory directory.
- The host name file is configured correctly:
 - *<host IP> <host name.domain> <host name> <alias>*

Example:

- 123.456.789.012 myhost.mydomain myhost

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Preinstallation: Setting Up the Environment

Setting **DISPLAY** (Linux or UNIX only)

On Linux and UNIX systems, you can install OracleAS Middle Tier by using a remote workstation. Setting the **DISPLAY** environment variable also enables you to run Installer remotely from a local workstation. On the system where you run the Installer, set **DISPLAY** to the system name or IP address of your local workstation.

Installation: Starting the Installer

To start your installation, perform the following steps:

- Insert your Oracle Application Server CD into the drive.
- On Linux or UNIX:
 - Mount the installation media
 - Run Oracle Universal Installer from the media
- On Windows 2000:
 - In the Autorun window that appears, choose **Install/Deinstall Products** or run `autorun.exe` directly from the `autorun` directory on your media

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Installation: Starting the Installer

To launch Oracle Universal Installer and install Oracle Application Server, perform the following steps:

- Insert your Oracle Application Server 10g Release 2 CD into the drive.
- On Linux or UNIX systems, mount the installation media. If you are using the Solaris Volume Management software (installed by default with Solaris Operating Environment), then the drive is mounted automatically when you insert it in the disk drive. On Windows 2000 with autorun capabilities, in the Autorun window that appears, choose **Install/Deinstall Products**. If your machine is not set up with the Autorun capability, run `autorun.exe` directly from the `autorun` directory on the media.

Specifying File Locations

Specify File Locations

Source

Enter the full path of the file representing the product(s) you want to install:

Path:

Destination

Enter or select a name for the installation and the full path where you want to install the product.

Name:

Path:

Copyright © 2005, Oracle. All rights reserved.

Specifying File Locations

In the Specify File Locations window, you can enter the Oracle home name and path. Each installation must have its own Oracle home. Initially, the Installer displays the list of names of the currently installed products. Enter a new name for your installation.

The Oracle home path must be a real, absolute path. It cannot contain environment variables or spaces.

Selecting a Product



Installer: Selecting a Product

In the Select a Product to Install window, select Oracle Application Server as a product to install the middle tier.

Selecting an Installation Type



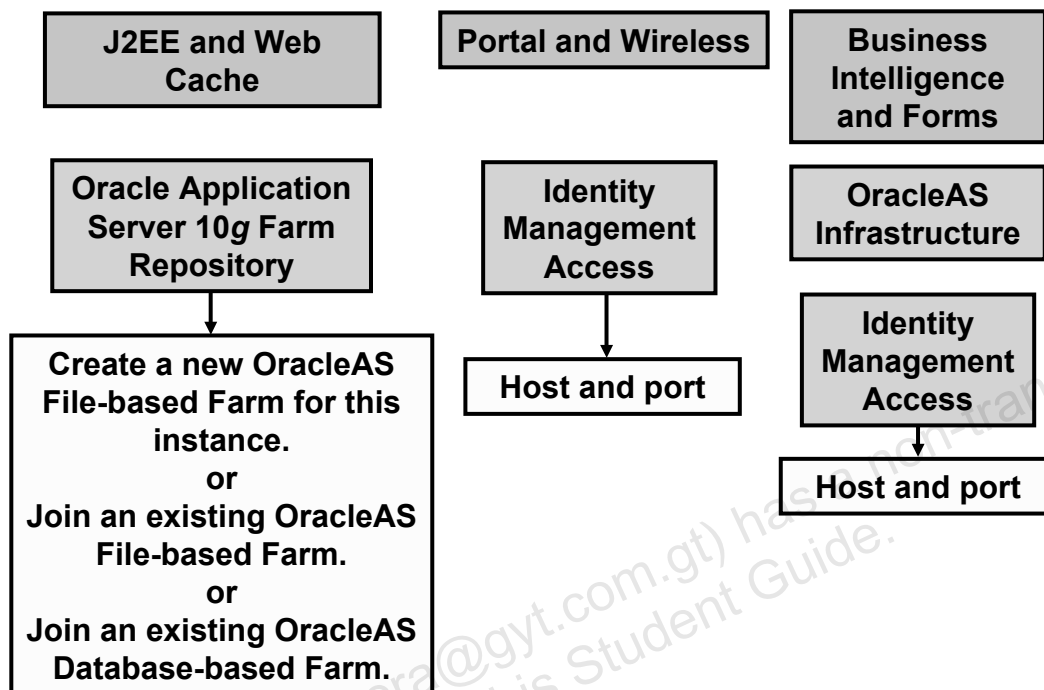
Selecting an Installation Type

The installation type defines the Oracle Application Server Middle Tier components that are installed on your machine. Select one of the following Oracle Application Server installation options in the Select Installation Type window:

- **J2EE and Web Cache:** Installs OracleAS Web Cache, Oracle HTTP Server, and OracleAS Container for J2EE
- **Portal and Wireless:** Installs OracleAS Web Cache, Oracle HTTP Server, OracleAS Container for J2EE, OracleAS Portal, and OracleAS Wireless
- **Business Intelligence and Forms:** Installs OracleAS Web Cache, Oracle HTTP Server, OracleAS Container for J2EE, OracleAS Portal, OracleAS Wireless, Oracle Discoverer, OracleAS Forms Services, and Reports Services

The installation type also defines what information is required from you in the later steps of the installation. For example, you need to provide information about how to connect to Oracle Internet Directory and SSO server if you are installing Portal and Wireless.

Middle-Tier Installation Options



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Middle-Tier Installation Options

When you install Oracle Application Server Middle Tier, select one of the following installation types:

- J2EE and Web Cache
- Portal and Wireless
- Business Intelligence and Forms

The J2EE and Web Cache installation type does not require infrastructure components. However, if you want to make this J2EE and Web Cache instance part of a farm, the Installer displays the Select Repository Type screen. In this screen, you select whether you want to create a new OracleAS File-based Farm, join an existing OracleAS File-based Farm, or join an OracleAS Database-based Farm.

If you select “Create a new Oracle Application Server File-based Farm for this instance,” the instance that you are installing will be a repository host. The instance holds the repository information, and you can point other instances to use the repository information stored with this instance.

If you select “Join an existing Oracle Application Server File-based Farm,” the Installer prompts you for the location of the repository host and the port.

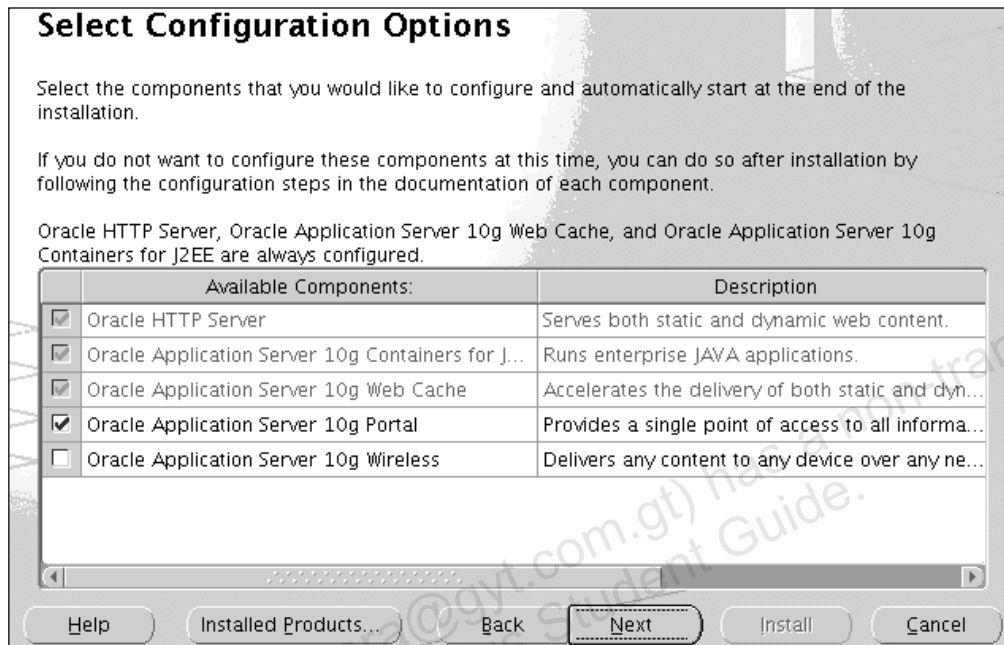
Middle-Tier Installation Options (continued)

If you select “Join an existing Oracle Application Server Database-based Farm,” the Installer prompts you for the location of OracleAS Metadata Repository. OracleAS Metadata Repository does not need to be registered with Oracle Internet Directory. This option does not require any Identity Management components.

When you install Portal and Wireless, the following components must be previously installed: Oracle Internet Directory, OracleAS Single Sign-On, and OracleAS Metadata Repository. You can install these components by installing OracleAS Infrastructure. Additionally, OracleAS Metadata Repository must be registered with Oracle Internet Directory.

The Business Intelligence and Forms installation is by default associated with OracleAS Infrastructure. OracleAS Infrastructure can be on the same host that contains the Business Intelligence and Forms installation or on a different host. When you install Business Intelligence and Forms, the following components are installed by default: Oracle HTTP Server, Oracle Application Server 10g Web Cache, and Oracle Application Server 10g Containers for J2EE. Additionally, OracleAS Metadata Repository must be registered with Oracle Internet Directory.

Installer: Selecting Component Configuration



Installer: Selecting Component Configuration

In the Component Configuration window, specify which middle-tier components should be configured during the installation and started upon its completion. The Installer automatically configures and starts some of the mandatory components (for example, Oracle HTTP Server, OC4J, and OracleAS Web Cache). You do not have to select all the components that are to be configured during the installation. The Installer copies all the required component files to the middle tier so that you can log in to Application Server Control and configure the corresponding components later.

Selecting Configuration Options

In the Configuration Options window, you can:

- **Select different options based on the installation type that you have selected**
- **Separate Identity Management–related configuration from Metadata Repository–related configuration at the time of the installation itself**

Available Components:		Description
<input checked="" type="checkbox"/>	Oracle HTTP Server	Serves both static and dynamic web content.
<input checked="" type="checkbox"/>	Oracle Application Server 10g Containers for J...	Runs enterprise JAVA applications.
<input type="checkbox"/>	Oracle Application Server 10g Web Cache	Accelerates the delivery of both static and dyn...
<input checked="" type="checkbox"/>	Oracle Application Server 10g Farm Repository	Oracle Application Server 10g Farm Repository..
<input type="checkbox"/>	Identity Management Access	Enables Single Sign-On for J2EE and web appli...

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Selecting Configuration Options

The Select Configuration Options window provides you with the configuration options relating to the J2EE and Web Cache, and the Portal and Wireless installation types. The change is mainly in the J2EE and Web Cache type of installation.

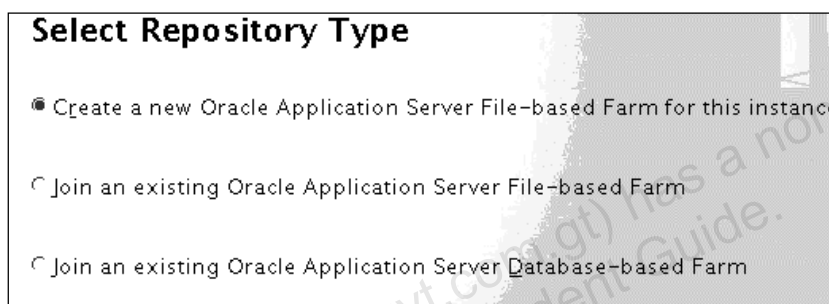
At the time of installation, you can separate the configuration of Identity Management and Metadata Repository for the J2EE and Web Cache type of installation. You can select:

- **OracleAS 10g Farm Repository:** This option enables the current instance to join an existing farm or create and join a new farm. Subsequently, you can select whether you want the instance to use an OracleAS File-based Farm or an OracleAS Database-based Farm.
- **Identity Management Access:** This feature enables your J2EE applications to make use of the OracleAS Single Sign-On and Oracle Internet Directory components provided by OracleAS Infrastructure. This feature applies only to the J2EE and Web Cache middle-tier type, because the other middle-tier types can use the OracleAS Infrastructure services automatically.

Selecting Repository Type

This window appears only when you:

- **Install J2EE and Web Cache**
- **Select OracleAS Farm Repository**



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Selecting Repository Type

The Select Repository Type window appears if you have selected to use OracleAS Farm Repository in the Select Configuration Options window. In this window, you find the following main options:

Create a new Oracle Application Server File-based Farm for this instance

- Select this option if:
 - This is the first instance that you are installing
 - You want a farm that stores its configuration metadata in files (as opposed to a database)

Join an existing Oracle Application Server File-based Farm

- Select this option if you have an existing J2EE and Web Cache instance that has an OracleAS File-based Farm and you want the instance that you are installing to join the existing farm.

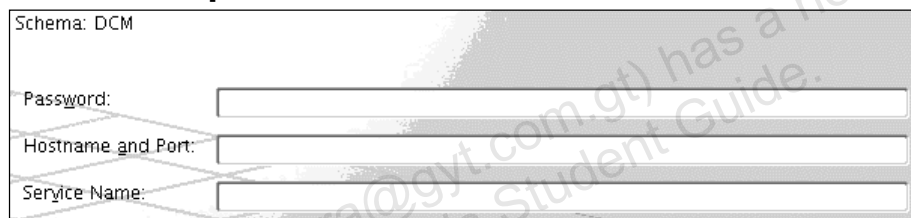
Join an existing Oracle Application Server Database-based Farm

- Select this option if you have OracleAS Metadata Repository, and if you want the instance that you are installing to join the existing repository.

Specifying Metadata Repository for Database-Managed Farm

The Specify Metadata Repository for DB-Managed Clusters window appears if:

- You are installing the J2EE and Web Cache installation type
- You have selected to join an existing OracleAS Database-based Farm
- You have not selected the Identity Management Access option



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specifying Metadata Repository for Database-Managed Farm

The Specify Metadata Repository for DB-Managed Clusters window contains three fields:

Password

- Specify the password for the Distributed Configuration Management (DCM) schema in OracleAS Metadata Repository. This schema is used by the OracleAS Database-Based Cluster feature.
- The initial password for the DCM schema is randomized. If your OracleAS Metadata Repository is registered with Oracle Internet Directory, then you can get the randomized password from Oracle Internet Directory.
- If your OracleAS Metadata Repository is not registered with Oracle Internet Directory, then you can reset the password of the DCM schema by using a SQL statement.

Host Name and Port

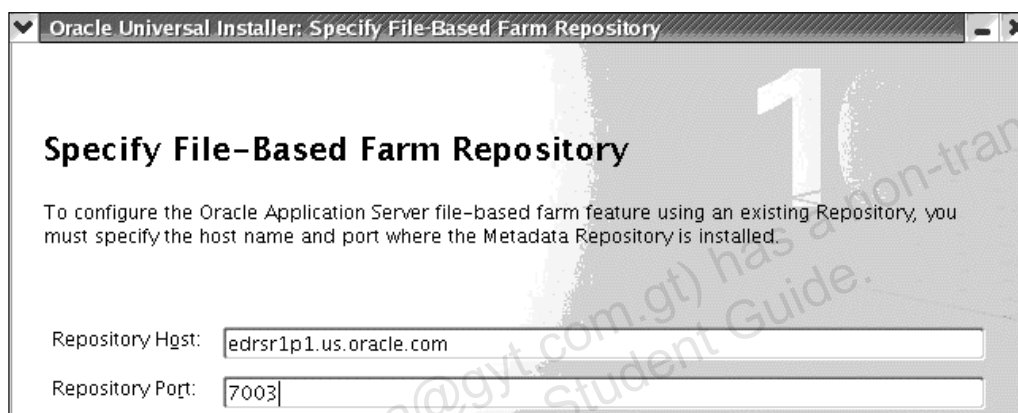
- Specify the host name and port for the database listener of OracleAS Metadata Repository.

Service Name

- Specify the service name for the database containing OracleAS Metadata Repository. Typically, the service name is the same as the global database name.

Selecting File-Based Farm Repository

This window appears when you have selected to join the J2EE and Web Cache installation type to an existing OracleAS File-based Farm repository.



Oracle Universal Installer: Specify File-Based Farm Repository

Specify File-Based Farm Repository

To configure the Oracle Application Server file-based farm feature using an existing Repository, you must specify the host name and port where the Metadata Repository is installed.

Repository Host:

Repository Port:

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Selecting File-Based Farm Repository

Specify the machine name and the port where the existing OracleAS File-based Farm is.

To determine the host and port values, you can use the following `dcmctl` command or use Application Server Control on the application server instance that hosts the farm repository:

```
$ $ORACLE_HOME/dcm/bin/dcmctl getRepositoryID
```

where, `ORACLE_HOME` specifies the location of the Oracle home directory of the instance that hosts the farm.

The command returns the host name and port in a host name:port format. Use these values in the respective fields in this window.

Specifying Port Configuration Options

Specify Port Configuration Options

Select the method which you want to use to configure the ports for Oracle Application Server 10g. If you decide to manually configure the ports, then you must specify the port numbers for each port in a text file and enter the filename below.

Configure Ports

☒ Automatic

☐ Manual:

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specifying Port Configuration Options

Select whether you want to use default port numbers for components or custom port numbers as specified in a port configuration file (the `staticports.ini` file).

Automatic

- Select this option to assign default ports to components. For a list of default ports, refer to the *Oracle Application Server Installation Guide*.

Manual

- Select this option if you have already created a port configuration file that specifies the port numbers that you want to use for each component. Enter the full path for this file in the provided field.
- This port configuration file is typically referred to as the `staticports.ini` file, but the name does not matter. The Installer reads this file and assigns the specified ports to the components.

Registering with Oracle Internet Directory

Register with Oracle Internet Directory

To register this instance of Oracle Application Server 10g with an existing Oracle Internet Directory, enter the hostname and port where Oracle Internet Directory is located.

Host:

Port:

☐ Use only SSL connections with this Oracle Internet Directory

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Installer: Registering with Oracle Internet Directory

As already discussed, the OracleAS Middle Tier components, such as OracleAS Portal, require Oracle Internet Directory provided by Identity Management of OracleAS Infrastructure. While installing the middle tier, you should specify the location (the host name and port) of Oracle Internet Directory that you want your middle tier to use.

Using Metadata Repository

Select Oracle Application Server 10g Metadata Repository

Select the Oracle Application Server 10g Metadata Repository that you want to use for this installation.
The format of the database entries below is as follows:
hostname : port : global database name : service name

Database Connect String

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

Using Metadata Repository

The middle-tier installations require the metadata repository to store its own metadata information. You specify the location of OracleAS Metadata Repository in this window.

Instance Name and the `ias_admin` Password

Specify Instance Name and `ias_admin` Password

All Oracle Application Server 10g instances installed on a host must have unique names. The hostname and domain name of the host are appended to the instance name.

Each Oracle Application Server 10g instance has its own password, regardless of which user performed the installation. Passwords are not shared across instances, even if the instances were installed by the same user.

The password must have a minimum of 5 alphanumeric characters, maximum 30 characters, and at least one of the characters must be a number.

Administrator Username: `ias_admin`

Instance Name:

`ias_admin` Password:

Confirm Password:

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Instance Name and the `ias_admin` Password

Specify Instance Name

Each middle-tier installation should have a unique name, which is used to identify the installation and the corresponding middle-tier instance on the system. Each installation creates one instance of Oracle Application Server. It is possible to scale up an installation from the J2EE and Web Cache installation type to the Portal and Wireless installation type. It is not possible to scale down an installation.

Specifying the `ias_admin` Password

A default Oracle Application Server installation administrative user `ias_admin` is created during the installation. Application Server Control uses the `ias_admin` user to manage the instance. The password that you select for `ias_admin` allows you to manage all instances of Oracle Application Server across the installation, run management tools, and facilitate future installations.

Installer: Summary



Installer: Summary

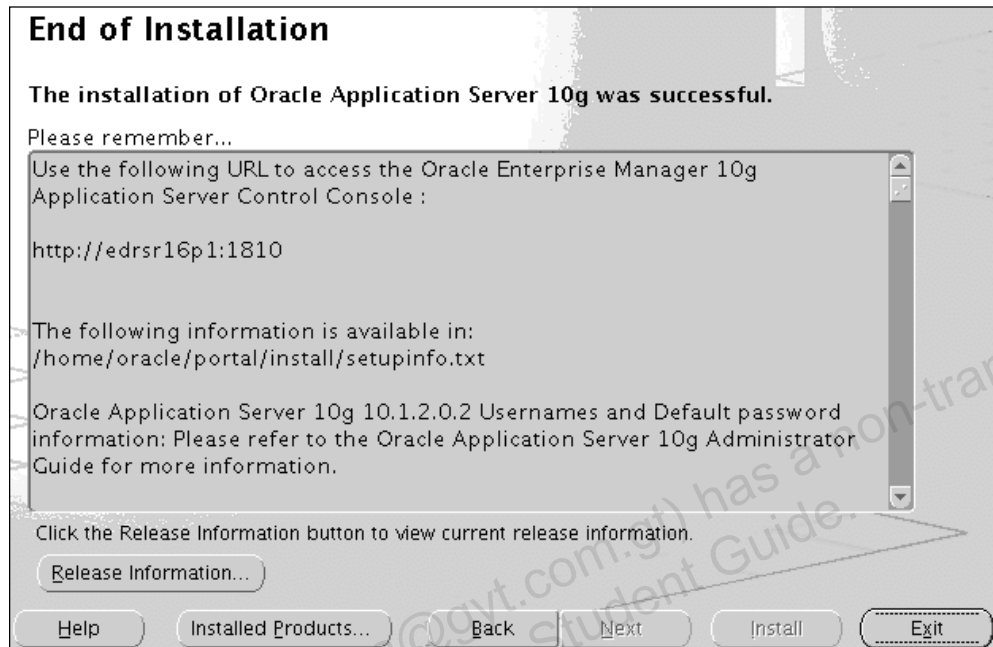
You can review all the settings in the Summary window before the actual installation process. These settings include source, destination, installation type, product language, space requirements, and a list of components.

To change any of these settings, click Back to return to the respective windows. When you click Install, the installation process begins.

Note: Insufficient disk space is indicated in red under Space Requirements.

The Installer then performs the installation action in three phases: copying, linking, and setting up files. Then, it invokes the component configuration assistants.

Installer: End of Installation



Installer: End of Installation

The End of Installation window appears at the end of the component configuration process. It notifies you whether the installation is successful or unsuccessful.

In this window, note the URLs that specify the access ports for Application Server Control and Web Cache Listener.

Accessing Application Server Control

General Stop All Restart All

Status: **Up**
 Host: edrsr16p1
 Version: 10.1.2.0.2
 Installation Type: Portal and Wireless
 Oracle Home: /home/oracle/portal
 Farm: infra.us.oracle.com

CPU Usage

Application Server (2%)
 Idle (58%)
 Other (40%)

Memory Usage

Application Server (57% 570MB)
 Free (1% 11MB)
 Other (42% 423MB)

System Components Enable/Disable Components Configure Component

Start Stop Restart Delete OC4J Instance

Select All | Select None

Select Name	Status	Start Time	CPU Usage (%)
<input type="checkbox"/> home	↑	Aug 24, 2005 11:13:49 PM	0.00
<input type="checkbox"/> HTTP_Server	↑	Aug 24, 2005 11:13:46 PM	0.53
<input type="checkbox"/> OC4J_Portal	↑	Aug 24, 2005 11:13:49 PM	0.00
<input type="checkbox"/> Portal:portal	↑	N/A	N/A
<input type="checkbox"/> Web Cache	↑	Aug 24, 2005 11:13:45 PM	0.00
<input type="checkbox"/> Management	↑	Aug 24, 2005 11:15:46 PM	1.50

Farm: infra.us.oracle.com

Instances can be grouped into clusters.

Repository Type: Data

Clusters

Select Name

There are no clusters in the farm.

Standalone Instances

These instances belong to the farm but are not part of any cluster.

Join Cluster

Select Name	Host	Oracle Home
<input type="radio"/> infra.edrsr16p1	edrsr16p1	/home/oracle/infra
<input type="radio"/> portal.edrsr16p1	edrsr16p1	/home/oracle/portal

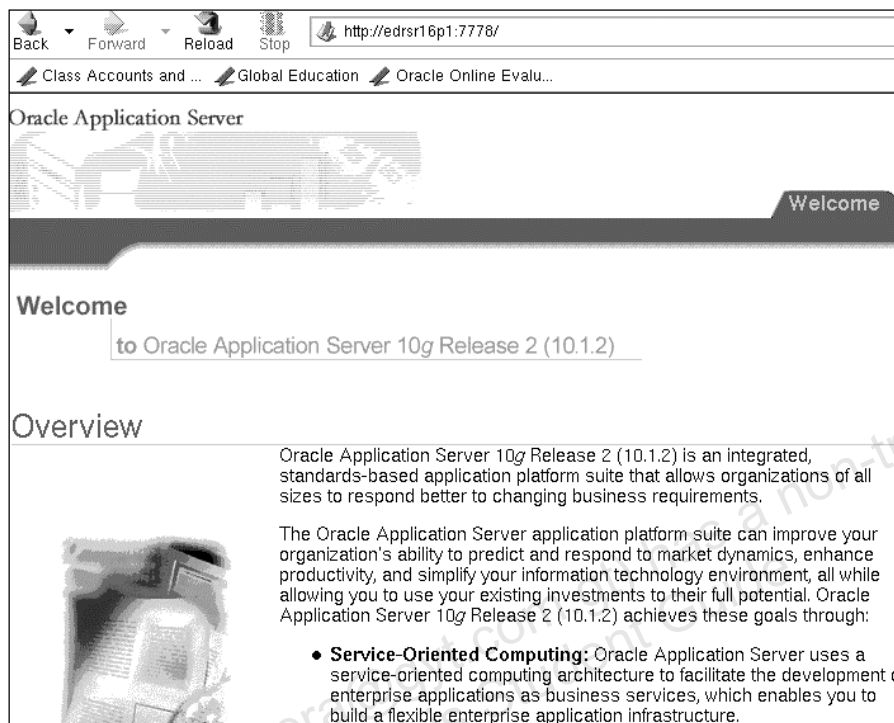
Accessing Application Server Control

You can access Application Server Control by using the URL `http://<host name>:<emport>`. The details of the ports can be obtained from the `portlist.ini` file in the `install` directory of your Oracle home.

The Portal and Wireless installation type is registered with OracleAS Infrastructure. Therefore, the farm page becomes the entry point for Application Server Control.

You can ensure that all the components that you had opted to configure during the installation are available in the System Components table.

Accessing the Welcome Page



Oracle Application Server Welcome Page

This page is a good starting point to validate your installation. If you do not remember the port on which Oracle HTTP Server has been installed, then check the `portlist.ini` file in the `$ORACLE_HOME/install` directory. This file lists all the ports that the Installer has assigned during installation.

Note: This file is populated during installation and is not dynamically updated. Any modifications to port values that are performed after installation are not reflected in this file.

The Oracle Application Server Welcome page contains information about how to access the documentation and the Quick Tour from Oracle Technology Network. The right pane is divided into three regions:

1. From the Release Notes region, you can obtain the latest information according to the Oracle Application Server installation that you have performed.
2. The Oracle Enterprise Manager region links you to Application Server Control to manage and configure your application server.
3. The New Features region contains links to obtain information about different key features, such as J2EE and Internet applications, Portal, and Wireless.

OracleAS Portal 10.1.4: New Features

OracleAS Portal is a Web-based application for building and deploying portals.

You can use the following new features of OracleAS Portal 10.1.4 for managing portals:

- **Improved caching architecture**
- **Support for SSL**
- **New URL format**
- **WSRP support**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal 10.1.4: New Features

OracleAS Portal is a Web-based application for building and deploying portals. It provides a secure, manageable environment for accessing and interacting with enterprise software services and information resources. Oracle Portal provides users centralized access to all data, applications, and business processes. By using Oracle Portal 10.1.4, you can seamlessly integrate enterprise applications, business intelligence applications, business process systems, and Web services into a single, highly productive workplace.

You can use the following new features of OracleAS Portal 10.1.4 for managing portals:

- **Improved caching architecture:** By implementing Edge Side Includes (ESI) processing, OracleAS Web Cache acts as the first point of contact for all page request processing. This type of processing simplifies the page metadata and enables different types of metadata to be cached.
- **Support for SSL:** By using the UTL_HTTP package, OracleAS Portal accesses information through secure database calls. If you are using HTTPS, then after configuring OracleAS Single Sign-On to use SSL, you need to update the OracleAS Single Sign-On query path URL for supporting the database calls.

OracleAS Portal 10.1.4: New Features (continued)

- **New URL format:** The URL format for accessing OracleAS Portal 10.1.4 has changed to `http://<host>:<port>/portal/pls/<dad>`. This new URL supports all portal services that are required to run in the OC4J_Portal instance.
- **WSRP support:** Web Services for Remote Portlets (WSRP) is a Web services standard that allows integration of visual, client-end Web services with portals or other intermediary Web applications. By using WSRP, you can enable interoperability between standards-enabled containers and any WSRP portal.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Upgrading OracleAS Portal Middle Tier from 10.1.2.0.2 to 10.1.4

To upgrade the portal schema from 10.1.2.0.2 to 10.1.4, you must perform the following tasks:

- 1. Stop all middle-tier instances.**
- 2. Ensure that Oracle Internet Directory and database processes are running.**
- 3. Perform the upgrade.**
- 4. Start all middle-tier instances.**
- 5. Access the upgraded OracleAS Portal.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Upgrading OracleAS Portal Middle Tier from 10.1.2.0.2 to 10.1.4

A new installation of OracleAS Portal 10.1.4 consists of three phases:

- Installing Oracle Application Server Infrastructure Release 10.1.2.0.2
- Installing Oracle Application Server Middle Tier Release 10.1.2.0.2
- After you have installed Oracle Application Server Infrastructure and Oracle Application Server Middle Tier, you need to upgrade Oracle Application Server Portal from 10.1.2.0.2 to 10.1.4. The steps to perform the upgrade are listed in the slide.

Preinstallation: Setting Up the Environment

To stop the middle tier, perform the following steps:

1. Use OPMN to stop all processes:

- `$ORACLE_HOME/opmn/bin/opmnctl stopall`

2. Use `emctl` to stop the Application Server Console:

- `$ORACLE_HOME/bin/emctl stop iasconsole`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Preinstallation: Setting Up the Environment

Before you perform the upgrade, you need to stop all the processes associated with each middle tier that uses the existing portal schema:

1. You can use the following Oracle Process Manager and Notification Server (OPMN) command within the Oracle home of each instance:

```
$> $ORACLE_HOME/opmn/bin/opmnctl stopall
```

2. You can stop the Application Server Console by using the following command:

```
$> $ORACLE_HOME/bin/emctl stop iasconsole
```

Note: For more information, refer to the *Oracle Application Server Portal Installation and Upgrade Guide 10g Release 2 (10.1.4)*.

You learn more about the `opmnctl` and `emctl` commands in the lesson titled “Using Oracle Application Server Management Tools.”

Preinstallation: Setting Up the Environment

Before you perform the upgrade, you need to ensure that the following processes are running:

- **The OracleAS Metadata Repository database that hosts the portal schema.**
- **The listener for the OracleAS Metadata Repository database.**
- **The Oracle Internet Directory instance where the portal schema is registered.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Preinstallation: Setting Up the Environment (continued)

Before you perform the upgrade, you need to ensure that the following processes are running:

- The OracleAS Metadata Repository database that hosts the portal schema
- The listener for the OracleAS Metadata Repository database
- The Oracle Internet Directory instance where the portal schema is registered

You can log in to the Application Server Control Console of the Oracle Identity Management instance to verify that the necessary processes are running and that the required components are configured properly.

Performing the Upgrade

- To perform the complete upgrade, you need to run the `mrua.sh` script.
- The following information is required for running the `mrua.sh` script:
 - Password for the `sys` user
 - Password for the `cn=orcladmin` administrator

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Performing the Upgrade

After performing the installation, you can use the Oracle Application Server Portal Upgrade feature to upgrade the portal schema in OracleAS Metadata Repository. To perform this upgrade, you need to do the following:

1. Mount the Oracle Application Server Portal Upgrade CD-ROM.
2. Run the `mrua.sh` script:

```
$ mrua.sh -oracle_home <Oracle Home for Metadata repository>
          -oid_host   <Host on which OID resides> -oid_ssl_port <SSL
port          for OID>
```

The description for the arguments used in running the script is:

- `-oracle_home`: The OracleAS Metadata Repository home directory
- `-oid_host`: The name of the Oracle Internet Directory host that has the registered OracleAS Metadata Repository
- `-oid_ssl_port`: The secure port for Oracle Internet Directory

You need to provide the following details while running the `mrua.sh` script:

- Password for the `sys` user
- Password for the `cn=orcladmin` administrator

Postinstallation: Setting Up the Environment

To start the middle tier, perform the following steps:

1. Use OPMN to start all processes:

- `$ORACLE_HOME/opmn/bin/opmnctl startall`

2. Use `emctl` to start the Application Server Console:

- `$ORACLE_HOME/bin/emctl start iasconsole`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Postinstallation: Setting Up the Environment

After you perform the upgrade, you need to start all processes associated with each middle tier that uses the existing portal schema.

1. You can use the following Oracle Process Manager and Notification Server (OPMN) command within the Oracle home of each instance:

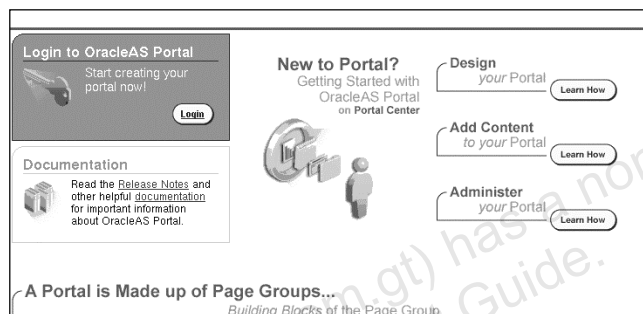
```
$> $ORACLE_HOME/opmn/bin/opmnctl startall
```

2. You can start the Application Server Console by using the following command:

```
$> $ORACLE_HOME/bin/emctl start iasconsole
```

Accessing the OracleAS Portal Welcome Page

- Enter the following URL:
<http://<host>:<port>/portal/pls/<dad>>
- Log in as the `portal` user with the password used for the `ias_admin` user.



Copyright © 2005, Oracle. All rights reserved.

Accessing the OracleAS Portal Welcome Page

You can verify that OracleAS Portal has been successfully installed and configured. To access the OracleAS Portal Welcome page, enter the following URL in your browser:

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

where “host” is the machine on which you have installed OracleAS Portal 10.1.4, and “port” is the port number of the OracleAS Middle Tier instance.

For example: <http://portal.mycompany.com:7779/portal/pls/portal>

You can log in to the portal by clicking the Login link on the Welcome page. Use `portal` as the username and the password that you have specified for the `ias_admin` user during the installation.

Summary

In this lesson, you should have learned how to:

- Describe the Oracle Application Server installation types and their requirements
- Perform preinstallation tasks
- Install the middle tier with the Portal and Wireless installation type
- Access the installed OracleAS Middle Tier components
- Upgrade 10.1.2.0.2 Portal to 10.1.4 Portal
- Verify the completion of the installation

ORACLE

Copyright © 2005, Oracle. All rights reserved.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

5

Using Oracle Application Server Management Tools

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Compare Oracle Enterprise 10g Grid Control with Oracle Enterprise Manager 10g Application Server Control**
- **Start and stop an Oracle Application Server instance or a component by using:**
 - **Application Server Control**
 - **Oracle Process Manager and Notification Server (OPMN)**
- **Use the `dcmctl` utility to obtain configuration information**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

- **View, monitor, and control your application server processes by using the Topology Viewer**
- **View and explain all performance metrics being monitored**
- **Change the Oracle Enterprise Manager 10g port values**
- **Query from the diagnostic message database repository**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Management Controls

- **Grid Control:**
 - Centrally manages Oracle products, host systems, and applications
 - Monitors distributed servers
- **Application Server Control:**
 - Manages one application server or application server farm/cluster
 - Monitors host server
- **Oracle Enterprise Manager 10g Database Control**
 - Manages one database and associated listeners
 - Monitors host server

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Management Controls

Each Oracle 10g product comes with its own management console that can be used to monitor and administer the product as an individual entity.

Each component of the console's associated product is referred to as a *managed target*.

Grid Control integrates all monitoring and management functions in a single console. It replaces Database Control and has links to Application Server Control. Application Server Control is needed to administer each application server target.

Application Server Control manages farm/cluster, application server, OracleAS Web Cache, Oracle HTTP Server, Oracle Application Server Containers for J2EE, OracleAS Single Sign-On, Portal, Wireless, Business Intelligence, and host targets.

Database Control manages database, listener, Automatic Storage Management (ASM), and host targets.

Comparing System Management Tools

Grid Control:

- **Manages the entire grid**
- **Supports collaborative management**
- **Provides real-time and historical performance monitoring**
- **Supports configuration management**
- **Provides group management functionality**

Application Server Control:

- **Provides real-time performance monitoring**
- **Offers ready management capabilities**
- **Supports configuration of application server instance**
- **Is available as a ready-to-use console**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

System Management Tools

As an administrator, you can leverage the management tools that Oracle Application Server provides. Using these management tools, you can manage, monitor, tune, and troubleshoot across Oracle Application Server instances.

In the context of Oracle Application Server 10g, Oracle Enterprise Manager has two main management tools:

- Grid Control
- Application Server Control

By using the Grid Control framework, you can obtain a unified view of your entire Oracle environment, including Oracle databases, Oracle Application Server, and Oracle Collaboration Suite. By deploying the Grid Control framework, you gain additional management capabilities, such as the following:

- Managing entire grid of application servers, databases, Oracle Collaboration Suites, and applications
- Providing multiple administrator accounts for Grid Control, each having separate roles or privileges for accessing different areas of functionality
- Automatic monitoring for all targets on the host system and collection of real-time and historical data for trend analysis

System Management Tools (continued)

- Enabling configuration management for tracking hardware and software configurations and implementing changes throughout the enterprise, patching, cloning, and policy standardization
- Using Application Service Level Management (ASLM) for monitoring business transactions and understanding your Web application's end-user experience
- Automating repetitive administration tasks through job system

Application Server Control is a Web-based console for administration and real-time performance monitoring of the entire Oracle Application Server platform including J2EE, OracleAS Portal, and OracleAS Wireless. Using Application Server Control, you can:

- Manage and configure application server components
- Monitor server performance
- Access a graphical view of the server environment
- Deploy and monitor J2EE applications
- Manage port values across application server components
- View server and application diagnostic logs

Grid Control: Overview

Grid Control provides the framework for monitoring and administering the grid, including the following:

- **A comprehensive overview of the grid's status**
- **A single point of monitoring for all associated targets**
- **Alerts with drill-down capability to identify trouble spots**
- **ASLM to enable holistic management of application performance**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Grid Control: Overview

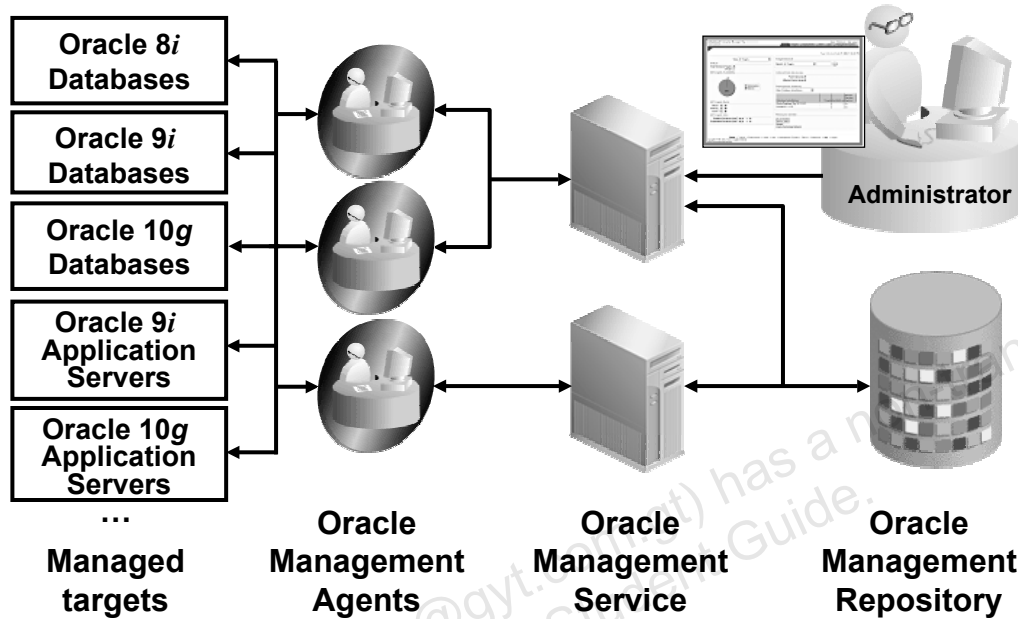
Grid Control gathers information from across the grid, monitoring key metrics and warning the administrator as those metrics approach threshold values. By using Grid Control, administrators can concentrate on areas that are operating inefficiently, without being distracted by areas of the grid that do not require administrator intervention.

Grid Control provides:

- **Application Service Level Management (ASLM):** Reduces the complexity of managing Web applications and ensures the highest quality of service
- **Policy-based standardization:** Lowers the cost of managing a large number of systems and helps to eliminate errors and inefficiencies. Automated monitoring of the policy ensures consistency and improved performance throughout the grid.
- **Automated provisioning and administration:** Reduces maintenance and administration costs for every system. With automation, the grid can scale out to more systems with minimal incremental cost increases for system administration.

For additional information about Grid Control, refer to the *Oracle Enterprise Manager 10g Grid Control* course.

Grid Control Architecture



Copyright © 2005, Oracle. All rights reserved.

Grid Control Architecture

The Grid Control framework includes the underlying components that provide the ability to centrally manage your computing environment. These components include:

- Oracle Management Repository (OMR)
- One or more than one Oracle Management Service (OMS)
- One or more than one Oracle Management Agent (OMA)

This framework can be concentrated on a single server or distributed across multiple servers, depending on the needs of the business and the size of the enterprise being managed.

Grid Control manages all currently supported versions of the Oracle database, Oracle9i Application Server Release 2, and Oracle Application Server 10g. In addition, Grid Control can be extended to monitor custom targets, such as routers, storage area networks, and many other systems.

Managed targets are monitored by OMA. There is one OMA for each host server. OMAs collect information about target availability, configuration, and performance and pass that information to an OMS. Each OMA talks to only one OMS, but an OMS may service hundreds of OMAs.

Oracle Management Service passes information received from OMAs to the OMR.

Grid Control Architecture (continued)

An OMS talks to only one OMR, although an OMR may service many OMSs. The OMS is also the point of connection for the administrator, producing the Grid Control console Web pages from information contained within the OMR and passing commands from the administrator to the appropriate OMA.

An OMR is a collection of schema objects within an Oracle database. It provides a single point of service for information about the grid's availability, performance, and configuration.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Application Server Control: Overview

- **Application Server Control provides monitoring and administration capabilities for each instance of Oracle Application Server.**
- **Using Application Server Control, you can manage:**
 - **Services such as hosts, databases, application servers, and Web applications**
 - **Hardware and software configurations across your enterprise**
- **Application Server Control enables the management of Oracle Application Server installations.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Application Server Control: Overview

Application Server Control provides monitoring and administration capabilities for each instance of Oracle Application Server. Application Server Control consists of a series of home pages, which enable you to manage an Oracle Application Server instance from your Web browser.

Using Application Server Control, you can:

- Manage a single Oracle Application Server instance, including all of its components and applications
- Manage a group of Oracle Application Server instances that share and take advantage of a common Oracle Application Server configuration repository (also known as a farm)
- Manage Oracle Application Server clusters, which enable you to streamline the process of configuring and deploying Web applications across multiple Oracle Application Server instances
- View a graphical topology of your application server environment
- Manage application server port values from a single, central location
- Locate and review application server log files to quickly diagnose problems
- Start and stop services
- Modify server configuration parameters

Application Server Control: Overview (continued)

- Configure J2EE resources
- Deploy applications

Application Server Control is installed as part of any Oracle Application Server installation on the application server host computer.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Application Server Control Architecture

The Application Server Control architecture includes:

- **Application Server Control Console**
- **Distributed Configuration Management (DCM)**
- **Oracle Process Manager and Notification Server (OPMN)**
- **Dynamic Monitoring Service (DMS)**
- **Oracle Management Agent**
- **Oracle Management Watchdog process**
- **Log Loader**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Application Server Control Architecture

Application Server Control Console

Each Oracle Application Server installation includes an Application Server Control Console for managing that installation. The Application Server Control Console provides the user interface to monitor and administer your application server components. In turn, Application Server Control is based on several underlying pieces that comprise the application server management stack, including Distributed Configuration Management (DCM), Oracle Process Manager and Notification Server (OPMN), and Dynamic Monitoring Service (DMS).

Distributed Configuration Management (DCM)

Application Server Control uses DCM to make configuration changes and to propagate configuration changes to the deployed applications across the cluster. DCM manages configurations among application server instances that are associated with a common Metadata Repository.

Oracle Process Manager and Notification Server (OPMN)

Application Server Control uses OPMN for tasks such as starting and stopping the components of your application server instance. OPMN provides process control and monitoring for application server instances and their components.

Application Server Control Architecture (continued)

Oracle Process Manager and Notification Server (OPMN) (continued)

It gathers component status information and distributes the status information to components that are interested in it.

Dynamic Monitoring Service (DMS)

Application Server Control uses Dynamic Monitoring Service to gather performance data about Oracle Application Server components.

Oracle Management Agent

Application Server Control uses a local version of Oracle Management Agent to monitor your application server components.

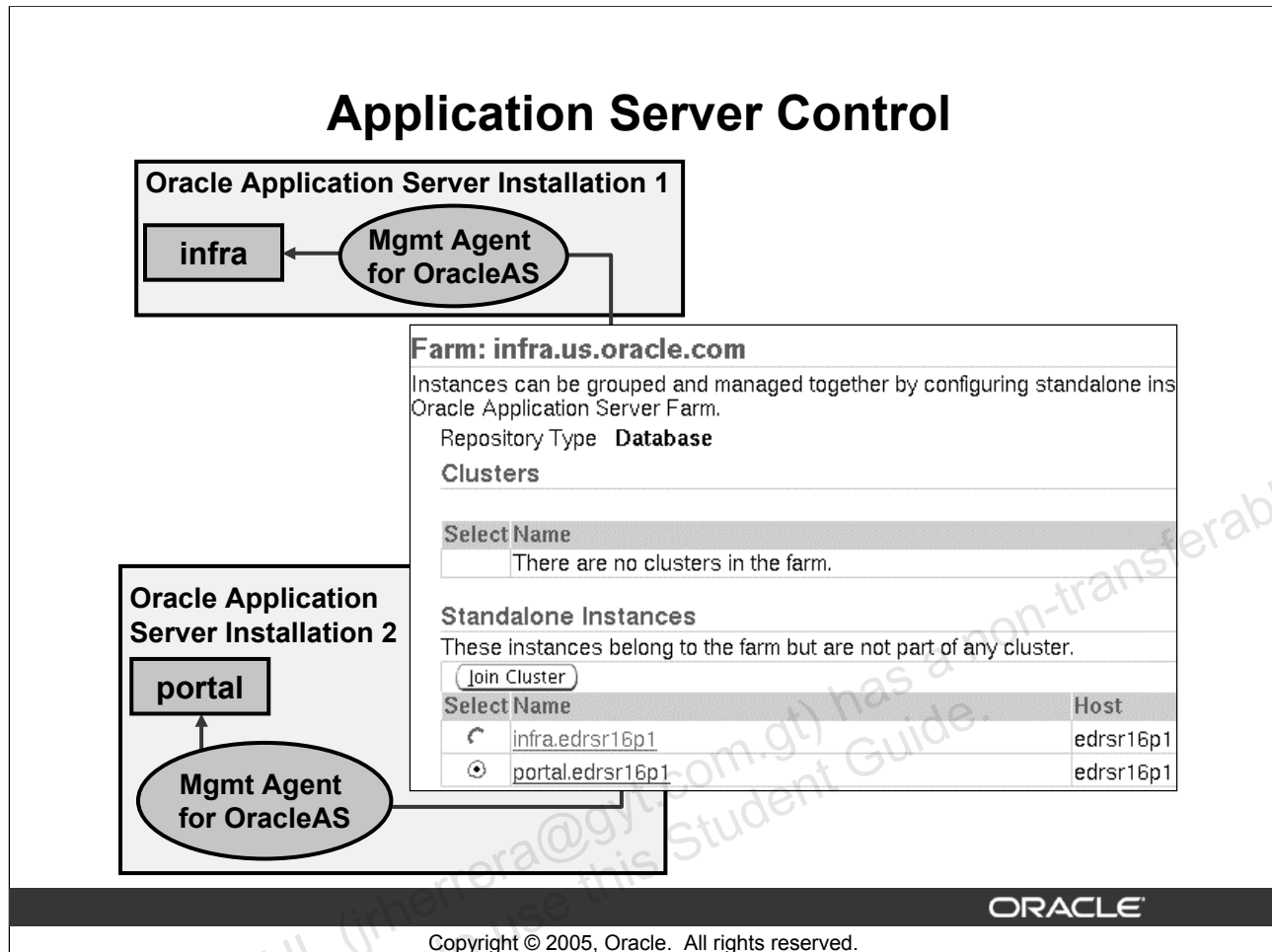
Oracle Management Watchdog Process

The Management Watchdog process monitors the Management Agent and the Application Server Control Console to ensure that both the processes are running and available at all times.

Log Loader

The Oracle Application Server Log Loader component is a process that periodically updates a Log Repository. A Log Repository stores diagnostic messages read from multiple log files across Oracle Application Server components in a single Oracle home. After the Log Loader starts, at regular intervals it reads the contents of log files incrementally and writes the contents to the Log Repository.

Application Server Control



Application Server Control

Application Server Control comprises the Management Agent for Oracle Application Server and the Application Server Control interface. Application Server Control is installed along with each Oracle Application Server installation.

Oracle Application Server provides a home page for each component of Oracle Application Server. Each home page provides the information that you need to monitor the performance and availability of Oracle Application Server from a particular level of management detail. Selected home pages also provide tools for configuring your Oracle Application Server components.

When using Application Server Control, you can:

- Use the Farm page to view a set of related Oracle Application Server instances on your network and to create clusters
- Use the Oracle Application Server Instance home page to manage all aspects of an individual Oracle Application Server instance
- Drill down to a component home page to monitor or to configure an individual component of Oracle Application Server. For example, use the Oracle HTTP Server home page to monitor the performance of your Web server, or use the applications property page accessible from the OC4J home page to deploy a custom Web-based application.

Using Application Server Control

- Each Oracle Application Server installation has its own Application Server Control.
- You must start the Application Server Control process with the `emctl` utility before using Application Server Control:
`emctl start iasconsole`
- You can get the Application Server Control port from the `setupinfo.txt` file in the `$ORACLE_HOME/install` directory.
- Invoke the Web browser and access Application Server Control using the following URL:
`http://<host name>.<domain>:<em_port>`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using Application Server Control

Each Oracle Application Server installation has its own Application Server Control. You must start the console process for Application Server Control before you can start managing your Oracle Application Server instance by using Application Server Control. To start the process, start Application Server Control with the following command:

```
emctl start iasconsole
```

The management agent for Oracle Application Server is also started and stopped with Application Server Control.

To access Application Server Control, invoke a Web browser and use the following URL:
`http://<host name>.<domain>:<em_port>`

For example, on a system with the fully qualified name `edcdr6p1.us.oracle.com` and port 1156 that is used by the process, you can access Application Server Control by using the following URL:

```
http://edrsr16p1.us.oracle.com:1156
```

Application Server Control Pages

Application Server Control provides different pages:

- **OracleAS Farm page:**
 - One or more Oracle Application Server instances that are associated with a common configuration repository
- **Oracle Application Server Instance home page:**
 - A single Oracle Application Server instance, which does not share the configuration repository with another instance; also available as a drilldown from the OracleAS Farm page
- **Oracle Application Server Component home page:**
 - Available as a drilldown from the Oracle Application Server Instance home page

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Application Server Control Pages

The slide describes the Enterprise Manager home page that forms the starting point when you first navigate to Oracle Application Server:

- The OracleAS Farm page enables you to view a list of all Oracle Application Server instances that are associated with a particular configuration repository.
- The Oracle Application Server Instance home page enables you to monitor and configure a single Oracle Application Server instance.
- The Component home page, such as Oracle HTTP Server home page, enables you to view, monitor, or configure an individual component of the application server. For example, use the Oracle HTTP Server home page to monitor the performance of your Web server, or use the OC4J home page to deploy a custom Web-based application. Drill down to the Component home page from an Instance home page.

OracleAS Farm Page

Farm: infra.us.oracle.com	
Instances can be grouped and managed together by configuring standalone instances Oracle Application Server Farm.	
Repository Type Database	
Clusters	
Select Name	
There are no clusters in the farm.	
Standalone Instances	
These instances belong to the farm but are not part of any cluster.	
Join Cluster	
Select Name	Host
infra.edrsr16p1	edrsr16p1
portal.edrsr16p1	edrsr16p1

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Farm Page

If you have installed one or more components that require OracleAS Infrastructure, or if you have OracleAS File-based Farm or OracleAS Database-based Farm, then your start page for Application Server Control is the OracleAS Farm page.

The OracleAS Farm page displays a list of the stand-alone Oracle Application Server instances and Oracle Application Server clusters that are associated with the configuration repository. The configuration repository can be a File-based Repository or Database-based Repository.

The following are the advantages of using the OracleAS Farm page:

- You can view, compare, and monitor multiple Oracle Application Server instances on multiple hosts.
- You can drill down to the Oracle Application Server Instance home page for each instance.
- You can create and manage Oracle Application Server clusters.

Topology Viewer: Overview

The Topology Viewer is:

- **A graphical tool to view and monitor application server processes managed by OPMN**
- **Available integrated and ready to use with Oracle Enterprise Manager 10g Application Server Control**
- **Available in two versions (HTML-only version and Java Applet version)**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Topology Viewer: Overview

The Topology Viewer is a graphical tool that provides you with a real-time view of application server processes managed by OPMN, and enables you to monitor them. It is now available integrated and ready to use with Application Server Control.

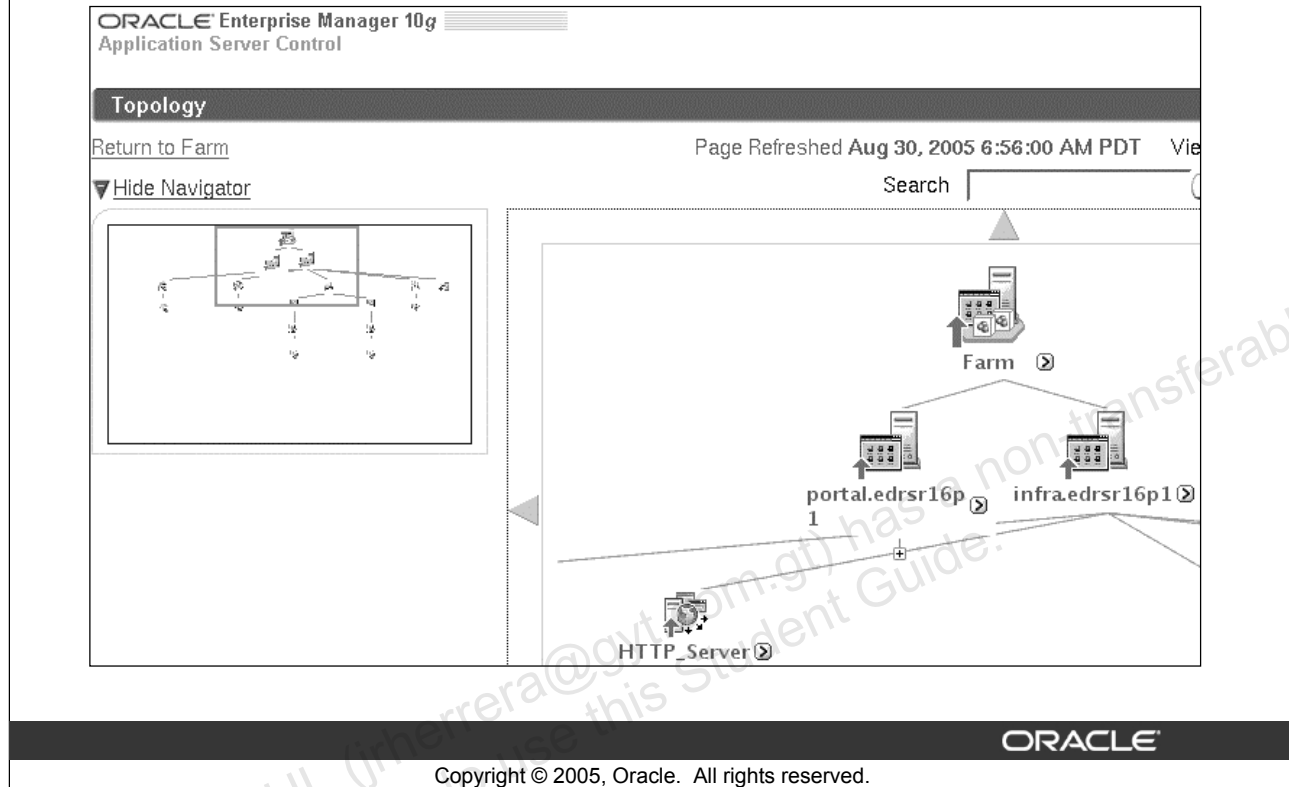
Each icon on the Topology Viewer represents an application server or a component of an application server. In addition to representing the type of object in the topology, this icon indicates the state of the component as up or down.

The Topology Viewer retrieves its data from OPMN and also provides an interface to manage the OPMN-controlled processes. Therefore, if OPMN is down, you do not see the components in the Topology Viewer. For example, if you are managing an Oracle Application Server Cluster, all the instances in the cluster appear in the Topology Viewer as long as OPMN is running in each of the cluster instances.

On installing Oracle Application Server, the Topology Viewer is immediately available without any additional configuration. There are two versions of the Topology Viewer that are available:

1. HTML-only version (default selection)
2. Java Applet version

Topology Viewer: Overview



Topology Viewer: Overview (continued)

Both versions offer the same functionality in a slightly different format. You can use the default HTML version to view and manage the topology. You do not require a Java Plug-in 1.4 for the HTML version. To use the applet version, however, you require Java Plug-in 1.4 or later. If Java Plug-in 1.4 is not already installed on your client machine, you are prompted to install it. Using the applet version of the Topology Viewer, in addition to viewing and managing the topology, you can also customize the colors used within the topology interface.

When you access either version of the Topology Viewer from Application Server Control, you see a clear visualization of your application server environment.

Accessing the Topology Viewer

The screenshot shows the Oracle Enterprise Manager 10g Application Server Control console. At the top, there's a header with 'ORACLE Enterprise Manager 10g' and 'Application Server Control'. Below this, a 'Farm: infra.us.oracle.com' is listed. A description states: 'Instances can be grouped and managed together by configuring standalone instances in a common repository. This collection of instances is an Oracle Application Server Farm.' The 'Repository Type' is set to 'Database'. On the right, there are tabs for 'Topology', 'Preferences', and 'Help'. The 'Topology' tab is active, showing the 'Topology Viewer' section. It prompts the user to 'Choose your preferred viewing format for the Topology Viewer. This preference will be stored in your browser as a cookie.' There are two radio buttons: 'HTML only' (selected) and 'Java Applet'. A 'TIP' note states: 'The applet version requires Java Plug-in 1.4 or higher. You will be prompted to download and install a new version of Java Plug-in if your browser does not meet this requirement.' At the bottom right of the dialog are 'Cancel' and 'Apply' buttons. A navigation bar at the bottom of the dialog shows 'Topology | Preferences | Help'.

Accessing the Topology Viewer

To access the Topology Viewer, perform the following steps:

1. Navigate to the OracleAS Farm page.
2. Click the Topology link at the top-right corner of the Application Server Control Console.

To change your preferred viewing from the HTML-only version to Java applet version, perform the following steps:

1. Click the Preferences link at the top-right corner of the Oracle Application Server Control Console.
2. Click the Topology Viewer link in the side navigation.
3. Select Java Applet.
4. Click Apply.

This preference is stored in your browser as a cookie.

Java Plug-in 1.4 or later is required to use the applet version. This Java Plug-in 1.4 is available in your Oracle Application Server installation in `$Oracle Home/jre/1.4.2`.

Benefits of Using the Topology Viewer

By using the Topology Viewer, you can perform the following administration tasks:

- **Visualize your application server environment.**
- **Start, stop, or restart application server processes.**
- **View the status of the application server farm, clusters, and member components.**
- **Monitor performance across the application server environment.**
- **Drill down to component home pages for additional details.**

ORACLE

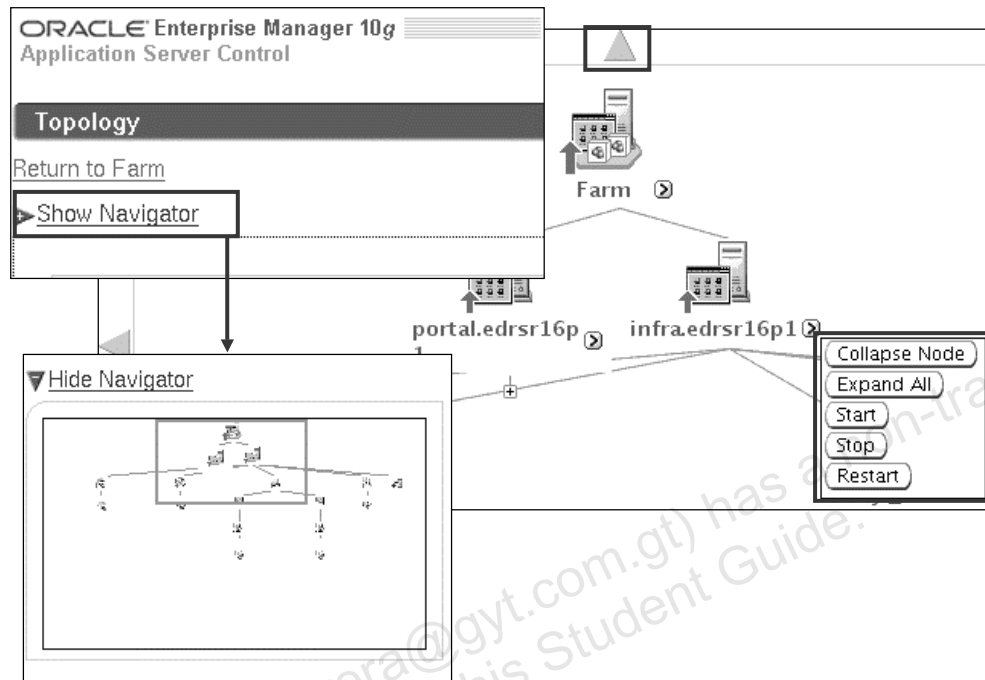
Copyright © 2005, Oracle. All rights reserved.

Benefits of Using the Topology Viewer

As an administrator, by using the Topology Viewer, you can:

- Visualize your application server environment
- Start, stop, or restart application server processes
- View the status of the application server farm, clusters, and member components
- Monitor performance across the application server environment
- Drill down to component home pages for additional details

Navigating Around the Topology



Copyright © 2005, Oracle. All rights reserved.

Navigating Around the Topology

On the Topology Viewer page, the topology controls palette includes a navigator. Use the navigator to visualize the whole topology and to quickly navigate around the topology. The navigator contains scaled-down icons representing each of the components in the topology.

To use the navigator, perform the following tasks:

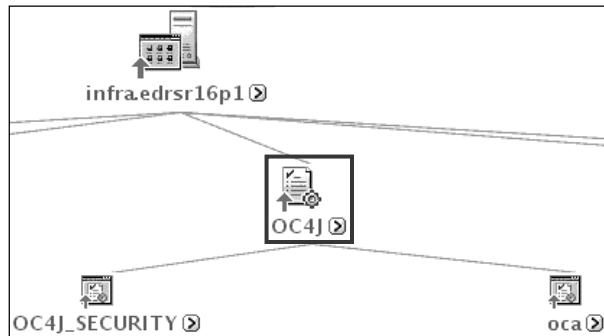
1. Click the Show Navigator link if the topology controls are not already showing.
2. Click in a region in the navigator to view a particular region of the topology.

Alternatively, you can click the arrow icons to move up, down, left, or right in the topology. In the Java Applet version, you can (in addition to regular navigation) navigate around the topology by clicking and dragging the canvas.

In addition to navigating between components, you can:

- **Control the processes:** Click the icon at the lower-right corner of a component. This displays a list of buttons corresponding to possible actions on that component, such as Start, Stop, and Restart. Clicking one of these buttons redirects the browser to a processing page for that action. When starting or stopping a process, the processing page goes to a confirmation page. Click the Return to Topology link to navigate back to the Topology Viewer window.

Navigating Around the Topology



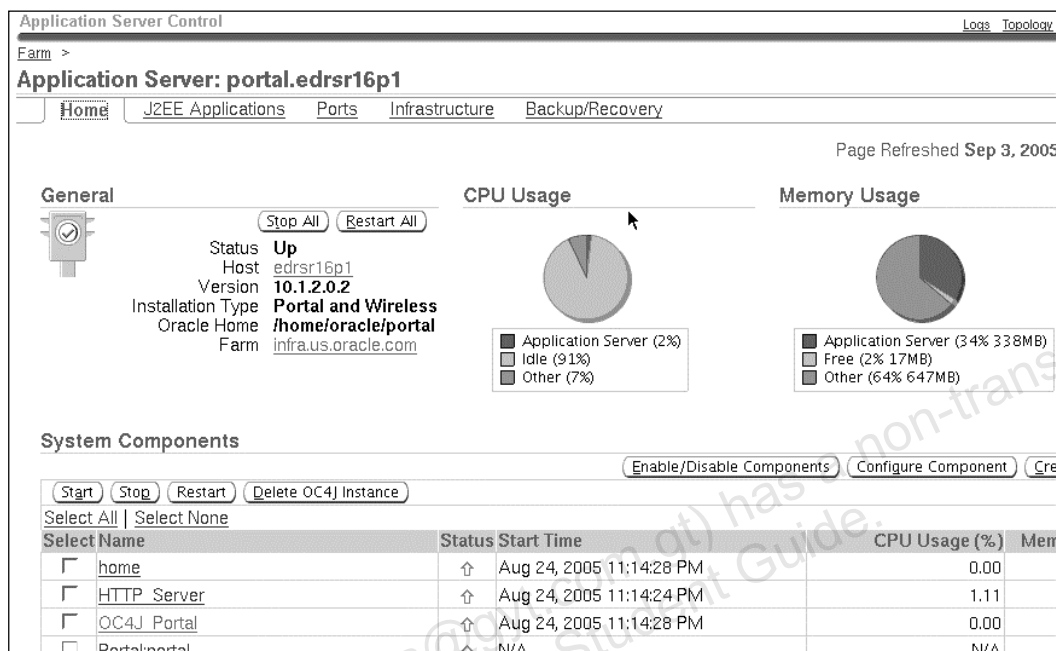
ORACLE

Copyright © 2005, Oracle. All rights reserved.

Navigating Around the Topology (continued)

- **Expand or collapse a node:** When you access the Topology Viewer for the first time, only the current Oracle Application Server instance is completely expanded. All the other instances in the farm are collapsed. A node can be collapsed or expanded. That is, the child nodes can be hidden or shown by clicking the icon at the lower-right corner of a node. This displays the Collapse Node and Expand All buttons. Clicking the Expand All button recursively expands a node and all of its child nodes.
- **Navigate to the target home page:** If the target in the topology has an Oracle Application Server Control home page, then you can access it by using the Topology Viewer. Apart from a node (such as dcm-daemon) that does not have a home page or a component-level node (such as OC4J), all other nodes redirect the browser to the home page when you click the node icons.
- **View the operating system process ID (PID):** You can view the operating system PID of a process and high-level performance statistics, such as memory and CPU usage, for key components, such as the Oracle HTTP Server instances.

Oracle Application Server Instance Home Page



Oracle Application Server Instance Home Property Page

The slide shows the Oracle Application Server Instance Home property page that provides general information about the Oracle Application Server instance. It has two charts, one chart shows the memory usage and the other, the CPU usage. The components are listed in the form of a table.

There are four property pages: Home, J2EE Applications, Ports, and Infrastructure. The Home property page is the first entry point. It contains the following sections:

- General:** This section contains status of the Oracle Application Server instance, the name of the machine that hosts this instance, the installation type, the Oracle home location, and the Farm that this instance is associated with. This section also displays charts showing the overall CPU and memory usage for all components.
- System Components:** The System Components section contains a table that provides the status of the components of the Oracle Application Server instance. This section shows only the components that have been configured and are enabled in the Oracle Application Server instance. The disabled components are not shown in this table. Using the link in the Name column of the table, you can access the component home page of Application Server Control.

Starting, Stopping, and Restarting Oracle Application Server Instances

Page Refreshed Sep 1

General

Stop All Restart All

Status: Up
Host: edrsr16p1
Version: 10.1.2.0.2
Installation Type: Portal and Wireless
Oracle Home: /home/oracle/portal
Farm: infra.us.oracle.com

CPU Usage

Application Server (1%)
Idle (91%)
Other (8%)

Memory Usage

Application Server (27% 21MB)
Free (1% 9MB)
Other (72% 721MB)

System Components

Start Stop Restart Delete OC4J Instance

Enable/Disable Components Configure Component

Select All | Select None

Select	Name	Status	Start Time	CPU Usage (%)
<input type="checkbox"/>	home	↑	Aug 24, 2005 11:14:20 PM	0.00
<input type="checkbox"/>	HTTP Server	↑	Aug 24, 2005 11:14:16 PM	0.71
<input type="checkbox"/>	OC4J Portal	↑	Aug 24, 2005 11:14:20 PM	0.00

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting, Stopping, and Restarting Oracle Application Server Instances

If you start an Oracle Application Server instance using the Oracle Application Server instance home page, then all the enabled components are activated. If you start an instance when some components are already started, then the remaining components are also started. If you stop an instance using the Instance home page, all running components are stopped. Restarting an instance using the Instance home page stops any components that are running, and starts all components.

To start, stop, or restart all the components of the Oracle Application Server instance, perform the following steps:

1. Navigate to the Oracle Application Server Instance home page.
2. In the General region, click the appropriate button.

You can use the System Components table to review the status of each component and confirm whether the component is running or not.

Oracle Application Server Component Home Pages

Each Oracle Application Server component has its own home page with the following elements:

- **General information section, which provides:**
 - The current state of the application
 - Buttons for starting and stopping the component
- **Status information, which shows CPU and memory usage**
- **Component-specific information**
- **Links to administrative functions**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

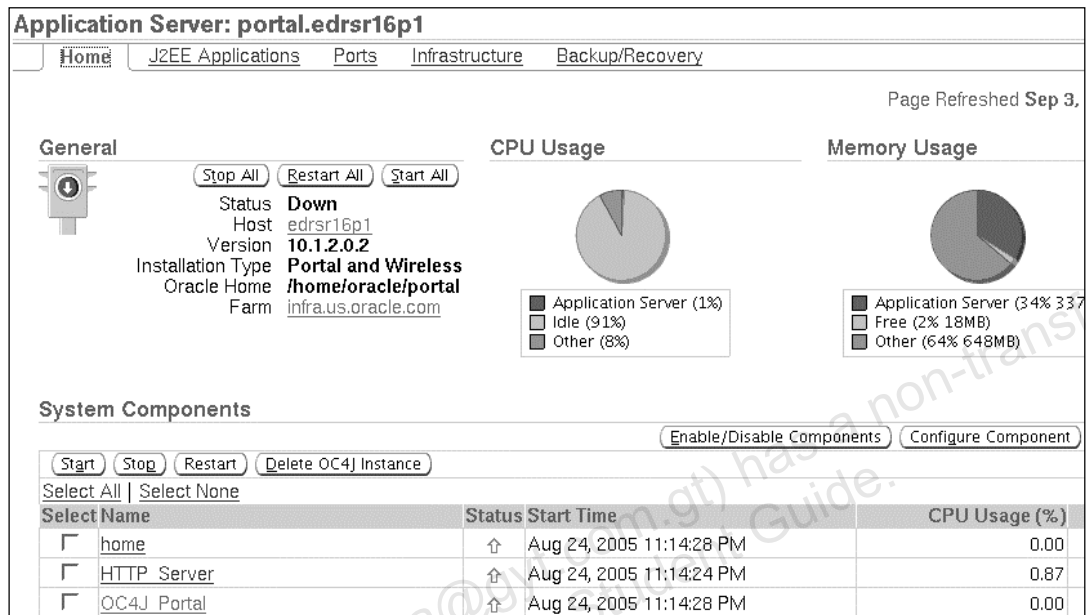
Oracle Application Server Component Home Pages

Each of the Oracle Application Server components has its own home page in Application Server Control, which has the following common elements:

- A general information section that includes an icon indicating the current state of the application, and provides buttons for starting and stopping the component
- Status information that includes CPU and memory usage charts, which you can use to get a snapshot of the performance of the component
- Component-specific information, such as a list of Virtual Hosts on the HTTP Server home page
- Links to administrative functions that you can use to modify the configuration of selected components. In most cases, this means that you can use a graphical user interface to modify complex configuration files.

The different component home pages are discussed in detail later when covering the components such as Oracle HTTP Server or OC4J.

Starting, Stopping, and Restarting the Components

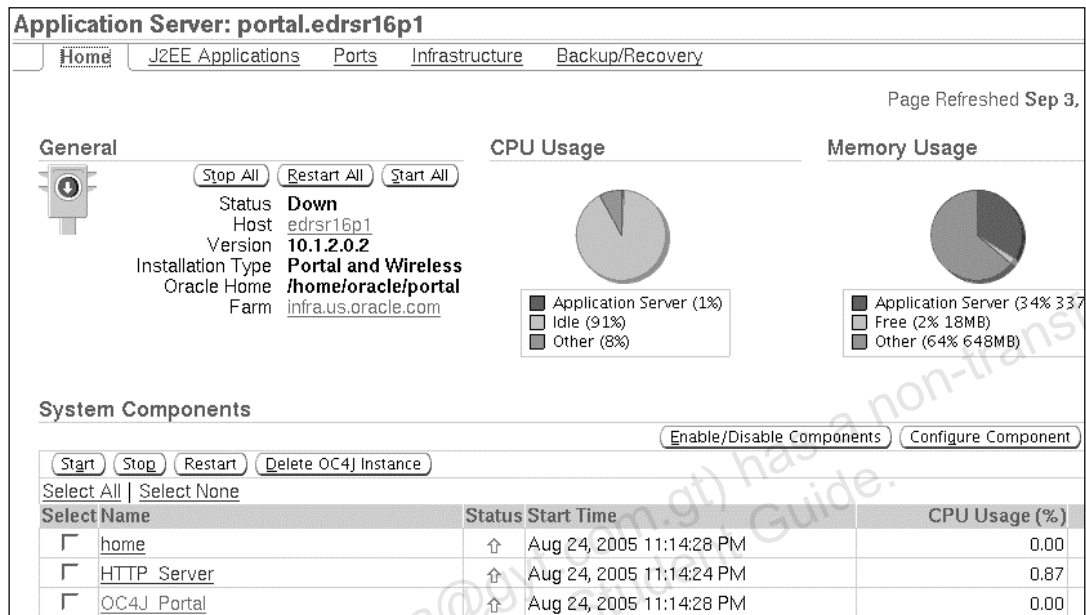


Starting, Stopping, and Restarting the Components

Each component of Oracle Application Server can be started, stopped, and restarted either from the System Components table on the Oracle Application Server Instance home page, or from the home page of the component. Additionally, all components can be started, stopped, and restarted using command-line instructions. The screenshot in the slide shows how to start a component from the Oracle Application Server Instance home page:

1. Open the Oracle Application Server Instance home page. Scroll down to the System Components table.
2. Select the component by selecting the check box to its left. Some components can be managed from their home pages. To access the home page, click the component instance name in the Name column.
3. Click the appropriate button at the top left of the System Components table to start or to stop the component. If you are on the component home page, you can click the appropriate start, stop, or restart button.

Obtaining Common Metrics About Oracle Application Server



Obtaining Common Metrics About Oracle Application Server

The Oracle Application Server Instance home page provides information about two important metrics in the Application Server metrics region:

- CPU Usage (%)**
 The pie chart in the slide shows the percentage of the central processing unit (CPU) currently in use by the selected Oracle Application Server instance or Oracle Application Server component. The CPU usage for an Oracle Application Server instance includes the total CPU usage of all the components of Oracle Application Server.
- Memory Usage (MB)**
 The pie chart shows the usage of memory by the Oracle Application Server instance or Oracle Application Server component. The memory usage of an Oracle Application Server instance includes the total memory used by all the components.

Obtaining Information About the Host Computer

Host: edrsr16p1

General

Status	Up
Boot Time	10 days
Operating System	Linux Red Hat Enterprise Linux AS release 3 (Taroon Update 3)
Hardware Platform	i686
IP Address	127.0.0.1
Total Physical Memory (MB)	1003
Total Disk Capacity (GB)	15
Total Swap Space (MB)	2001

Load

CPU Utilization (%) **9**

Run Queue Length (5 minute average) **0.46**

Processes **200**

Memory Utilization (%) **99**

Swap Utilization (%) **48**

Memory Scan Rate (pages per second) **9.2E-4**

Total I/Os per second **31.23**

Longest Service Time (ms) **50.75**

System Components

Start Stop Restart Delete OC4J Instance

Select Name	Status	Start Time
<input type="checkbox"/> home	↑	Aug 24, 2005 11:14:28 PM
<input type="checkbox"/> HTTP Server	↑	Aug 24, 2005 11:14:25 PM
<input type="checkbox"/> OC4J Portal	↑	Aug 24, 2005 11:14:28 PM

ORACLE

Copyright © 2005, Oracle. All rights reserved.

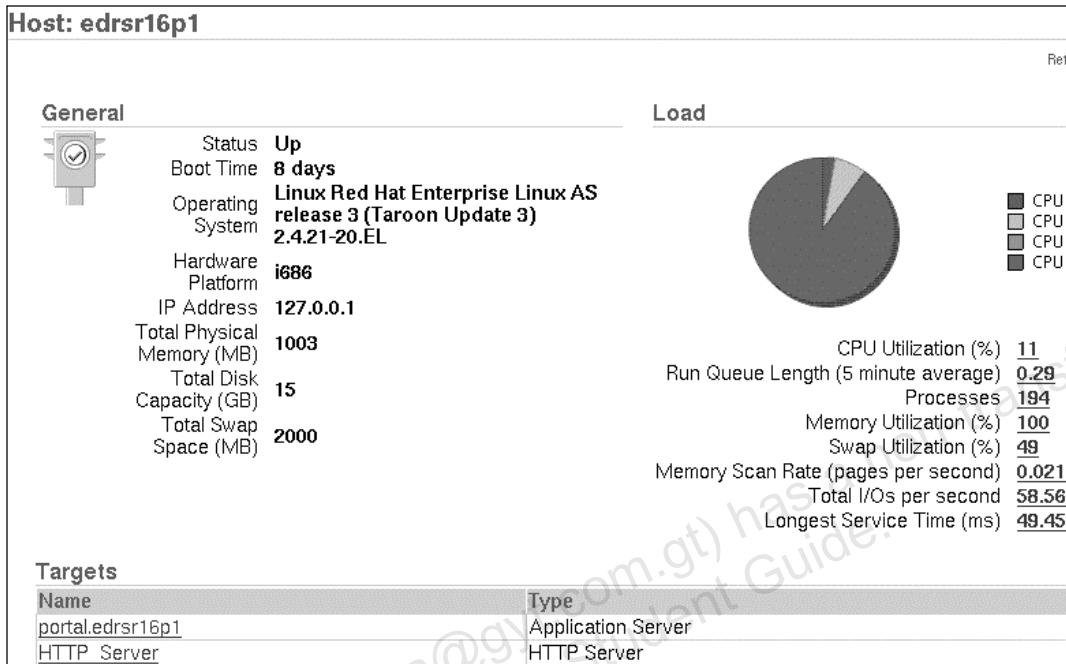
Obtaining Information About the Host Computer

Besides monitoring how your Oracle Application Server software is running, you can also review the status of the host computer where Oracle Application Server is installed. By reviewing the host statistics on a regular basis, you can troubleshoot existing performance problems or prevent performance problems from happening in the future.

To review the status of your Oracle Application Server host, perform the following steps:

1. Navigate to the Oracle Application Server Instance home page.
2. In the General region of the page, click the name of the host computer. Enterprise Manager displays the Host home page. For more information, click Help at the top of the page.

Oracle Application Server Host Home Page



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Host Home Page

The Oracle Application Server Host home page includes the following:

- **General:** Contains general information about the host computer (including whether or not the system is up and running), the operating system, the disk capacity, and the memory
- **Load:** Contains the CPU usage chart and a list of performance metrics that you can use to determine the overall health of the system. When you click the value of a metric, Application Server Control displays the Metric page that shows metric statistics and a time-based chart.
- **Targets:** Lists the Oracle Application Server components (or targets) that are currently installed on this host. Click the name of the target to display the Enterprise Manager page for that component.
- **Real-Time Metrics:** Contains a set of links to help you analyze the current available disk space, number of users, and other real-time statistics that can affect the overall performance of your application server. Each page that contains real-time data can be refreshed to display the most recent data.
- **Open Telnet Session:** Click this button to open a telnet session on the host computer. After you log in, you can use the operating system command line to learn more about the status and performance of the host computer.

Viewing Performance Metric Details

Using metrics data, you can get real-time update from a single page about the performance of the application server instance and individual components being monitored by Application Server Control.

You can:

- **Obtain details for each performance metric, including real-time data for each metric**
- **View data over a period of time**
- **Explain how the metric is defined**
- **Use Application Server Control to resolve potential performance problems**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Viewing Performance Metric Details

Metrics are units of measurement used to report the health of the system. In the Application Server Control Console, metrics provide you with real-time data about the overall performance of the application server instance and its components. From the All Metrics page, you can view a comprehensive list of metrics that are monitored by Application Server Control.

Use the Application Server Control All Metrics page to:

- Obtain details for each performance metric, including real-time data for each metric
- Use real-time refreshing of a metric chart to view data over a period of time
- Use online Help to understand how the metric is defined
- Resolve potential performance problems

After identifying potential performance problems based on the available metric details on the page, consult online Help to understand the metric better. Use metric information to improve performance and to resolve issues.

Viewing Performance Metric Details

System Components

Start Stop Restart

Select All | Select None

Select Name

<input type="checkbox"/>	home
<input type="checkbox"/>	HTTP Server
<input type="checkbox"/>	OC4J Portal
<input type="checkbox"/>	Portal:portal

JDBC Usage

Open JDBC Connections	7
Total JDBC Connections	1,051
Active Transactions	1
Transaction Commits	1
Transaction Rollbacks	0

Related Link [All Metrics](#)

OC4J Instance Metrics

- OC4J Instance Metrics
- OC4J Application Metrics
- OC4J Web Module Metrics
- OC4J Servlet Metrics
- OC4J JSP Metrics
- OC4J EJB Module Metrics
- OC4J EJB Metrics
- OC4J EJB Method Metrics
- OC4J JVM Metrics
- OC4J Datasource Metrics

OC4J Instance Metrics

OC4J Instance - Heap Usage (MB)
OC4J Instance - Start Time (ms since Epoch)
OC4J Instance - Active Sessions
OC4J Instance - Active Requests
OC4J Instance - Request Processing Time (seconds)
OC4J Instance - Requests Per Second
OC4J Instance - Active EJB Methods

Viewing Performance Metric Details (continued)

You can get a complete list of all the performance metrics for a particular component by clicking All Metrics on the target's home page. For example, to view the complete list of metrics and a particular metric for OC4J, perform the following steps:

1. Navigate to your Application Server Control instance home page.
2. Select a target component (OC4J_Portal).
3. Click All Metrics to display all the metrics defined for this component.
4. Click the plus (+) sign next to a specific metric (OC4J Instances Metrics).
5. Click OC4J Instance – Requests Per Second to view detailed metric information on the Metric Detail page. In this example, Requests Per Second is the rate at which servlets and JavaServer Pages pages (JSPs) are being invoked for OC4J instances in the application server.

If you leave the Metric Detail page in the window, you can set the page to automatically refresh itself every 30 seconds, 1 minute, or 5 minutes. You can see real-time changes in the metric data.

Application Server Ports Page

Application Server:portal.edrsr16p1				
Home	J2EE Applications	Ports	Infrastructure	Backup/Recovery
Page Refreshed Sep 1, 2005 11:46:30 PM				
The Port In Use column is empty if the port is not defined or if the component is not running. The Configure column contains an icon if you can configure the port using Enterprise Manager. Otherwise, you must refer to the component documentation. Regardless of how you modify the ports, you must consider any port dependencies before modifying a port value. More information: About Oracle Application Server Port Dependencies				
Component	Type	Port In Use	Suggested Port Range	Configure
DCM Object Cache	Cache Discovery Port		7100-7199	
home	JMS	12603	12601-12700	
home	RMI	12403	12401-12500	
home	AJP	12503	12501-12600	
Log Loader	Management		44000-44099	
OC4J_Portal	JMS	12604	12601-12700	
OC4J_Portal	RMI	12404	12401-12500	
OC4J_Portal	AJP	12504	12501-12600	

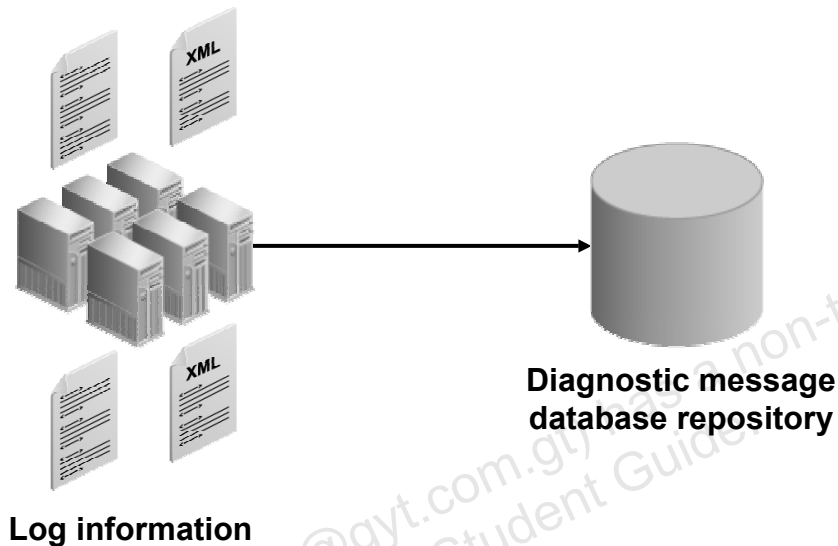
ORACLE

Copyright © 2005, Oracle. All rights reserved.

Application Server Ports Page

Use the Application Server ports page to view a list of all the ports currently in use by the components of this Oracle Application Server instance. This page is important when you are troubleshooting port conflicts among the various application server components. Port Range is the range of port numbers reserved for a component when it is installed. Port In Use is the port currently in use by the component. A link (pencil icon) to the appropriate configuration page is provided where you can modify the port settings for the component. If no link is provided, refer to the component administration documentation for more information.

Storing Log Information in the Diagnostic Message Database Repository



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Storing Log Information in the Diagnostic Message Database Repository

Each component of Oracle Application Server generates a set of log files. The log files contain messages that record all types of events, errors, warning messages, and additional information. As an application server administrator, you can identify and diagnose performance and configuration issues by using the information in the log files.

The Application Server Control Console lists log files and enables you to search them across the Oracle Application Server components. You can either view these files from the Application Server Control Console page or download a log file to your local client and then view it by using some file-viewing utility.

In Oracle Application Server 10g (9.0.4), only an XML file-based repository stored in the local file system was supported. In Oracle Application Server 10g Release 2, diagnostic log messages can also be stored in a diagnostic message database repository. The diagnostic message database repository can contain messages from multiple Oracle Application Server instances that belong to an Oracle Application Server Farm. The Diagnostic Message Loader (LogLoader) component collects the log messages from multiple Oracle Application Server component log files and loads these messages into the diagnostic message database repository. Although messages from multiple Oracle Application Server instances are stored in a single database, you can distinguish messages on the basis of the instance that originates the message.

Storing Log Information in the Diagnostic Message Database Repository (continued)

This feature is beneficial especially for a large Oracle Application Server installation, because by using this feature an administrator can easily manage an Oracle Application Server Farm.

However, if the LogLoader or the database repository fails, you can look at the local LogLoader log at each instance, and the alert logs and other diagnostic tools at the repository instance. The LogLoader logs its diagnostic messages to a local file. You can look at the LogLoader log files to diagnose any communication problems between the LogLoader and the repository.

You can view the connect string information for the database hosting the diagnostic message repository by clicking the Logs link on the Application Server Control Console page. You can use the `printlogs` command-line tool to query and display the contents of the repository.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Log Viewer

View Logs

Page Refreshed Sep 2, 2005 1:59:17 AM

Log Files Search Log Repository

The Log Files tab lists the log files for this application server. View a log file by clicking on the Log File name in the search results table.

Simple Search

Available Components

ADF Business Components

ASCLONE

Backup/Recovery

DCM

Enterprise Manager

HTTP_Server

LogLoader

OC4J_Portal

OPMN

Port Tunneling

Move

Move All

Remove

Remove All

Selected Components

home

Advanced Search

Search

Results: 10 Log Entries Retrieved

Component Type	Component Name	Log Type	Log File	Modified	Size (bytes)	OC4J Island	OC4J Island Process
OC4J Application	home	Application portietapp	application.log	August 24, 2005 11:14:31 PM PDT	1600	default_island	1
OC4J Application	home	Application default	global-application.log	August 24, 2005 11:16:02 PM PDT	1070	default_island	1
OC4J Application	home	Application BC4J	application.log	August 24, 2005 11:15:34 PM PDT	472	default_island	1

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Log Viewer

The Application Server Control's Log Viewer capabilities provide a unified view of logs from different components of Oracle Application Server.

From a single HTML interface, you can quickly select one or more components from which you would like to see log files. All associated log files are then displayed for the selected components, and you can directly search the log as needed.

Advanced search capabilities are also available that enable you to further restrict the list of log files to those that apply only to your current task or responsibility. For example, from an advanced search, you can choose to display only those log files that affect a particular J2EE application.

emctl Utility

- You can use **emctl** to start or stop Application Server Control, or to check its status:

```
$> emctl start iasconsole  
$> emctl stop iasconsole  
$> emctl status iasconsole
```

- When you start or stop Application Server Control, the management agent for Oracle Application Server is also started or stopped.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Command-Line Utility: **emctl**

Each Oracle Application Server installation has its own Application Server Control. The Application Server Control process is started at the end of your Oracle Application Server installation. You may need to start it manually after each system boot. You can also create a script to start it automatically during system boot.

To start Application Server Control, log in to the account that you used to install Oracle Application Server, and enter the following command:

```
$> $ORACLE_HOME/bin/emctl start iasconsole
```

When you start Application Server Control, the management agent for Oracle Application Server is also started automatically. Similarly, when you stop Application Server Control, the management agent for Oracle Application Server is also stopped.

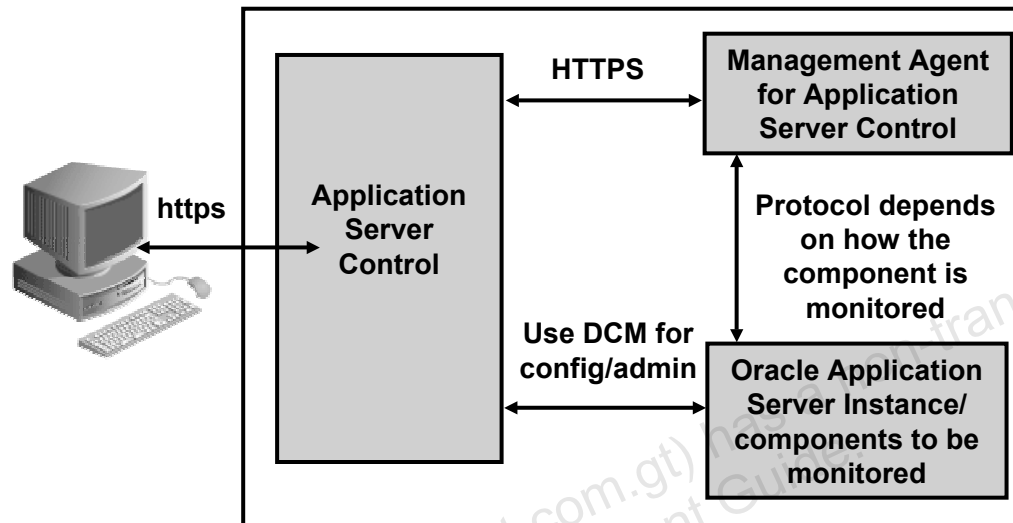
To stop Application Server Control, enter the following command:

```
$> $ORACLE_HOME/bin/emctl stop iasconsole
```

To check whether Application Server Control is functional, use the following syntax:

```
$> $ORACLE_HOME/bin/emctl status iasconsole
```

Enabling SSL for Application Server Control



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Enabling SSL for Application Server Control

You can access the Application Server Control Console through your Web browser using the nonsecure HTTP. In addition, communications between the local Management Agent and the Application Server Control Console are transferred over insecure connections.

To secure the communications between the Web browser and the Application Server Control Console, and between the Application Server Control Console and the Management Agent, you can use the `emctl secure iasconsole` command-line utility.

The communications (such as obtaining monitoring information and configuration and administration tasks) between the Management Agent and the application server being monitored, and Application Server Control and the application server being administered are not affected in any way when you use the `emctl secure iasconsole` utility. However, you can secure those communication paths by performing the application server security configuration steps for the particular path. In addition to their SSL configurations, some components also require you to perform configuration changes to the Application Server Control's Management Agent.

Enabling SSL for Application Server Control

You can enable SSL for Application Server Control for better security.

<pre>loracl@EDRSR16P1 bin\$./emctl stop iasconsole TZ set to US/Pacific Oracle Enterprise Manager 10g Application Server Control Copyright (c) 1996, 2005 Oracle Corporation. All rights reserved. http://EDRSR16P1:1810/emd/console/aboutApplication Stopping Oracle Enterprise Manager 10g Application Server Control loracl@EDRSR16P1 bin\$./emctl secure iasconsole TZ set to US/Pacific Oracle Enterprise Manager 10g Application Server Control Copyright (c) 1996, 2005 Oracle Corporation. All rights reserved. http://EDRSR16P1:1810/emd/console/aboutApplication Generating Standalone Console Root Key (this takes a minute)... Done. Fetching Standalone Console Root Certificate... Done. Generating Standalone Console Agent Key... Done. Storing Standalone Console Agent Key... Done. Generating Oracle Wallet for the Standalone Console Agent... Done. Configuring Agent for HTTPS... Done. EMD_URL set in /home/oracle/portal/sysman/config/emd.properties Generating Standalone Console Java Keystore... Done. Configuring the website ... Done. Updating targets.xml ... Done loracl@EDRSR16P1 bin\$./emctl start iasconsole TZ set to US/Pacific Oracle Enterprise Manager 10g Application Server Control Copyright (c) 1996, 2005 Oracle Corporation. All rights reserved. http://EDRSR16P1:1810/emd/console/aboutApplication Starting Oracle Enterprise Manager 10g Application Server Control started successfully. loracl@EDRSR16P1 bin\$</pre>	1. Stop Application Server Control.
	2. Secure Application Server Control.
	3. Start Application Server Control.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Enabling SSL for Application Server Control (continued)

To configure security for Application Server Control, perform the following steps:

1. Stop Application Server Control using the command:
\$ \$ORACLE_HOME/bin/emctl stop iasconsole
2. Secure Application Server Control using the command:
\$ \$ORACLE_HOME/bin/emctl secure iasconsole
Enterprise Manager secures Application Server Control.
3. Start Application Server Control using the command:
\$ \$ORACLE_HOME/bin/emctl start iasconsole

Test the security of Application Server Control using HTTPS instead of HTTP in the URL to access Application Server Control. For example:

<https://edrsr16p1.us.oracle.com:1156/>

Changing Oracle Enterprise Manager 10g Port Values

You can change the following Application Server Control ports associated with your Oracle Application Server instance:

- **Management Agent (Oracle home in your application server) port**
- **Application Server Control Console port**
- **OC4J Remote Method Invocation (RMI) port used by the Application Server Control OC4J instance**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Changing Oracle Enterprise Manager 10g Port Values

The Application Server Control Console is a Web application that runs in a Web container for a stand-alone OC4J instance. After Oracle Application Server is installed, you can view the current port values and the suggested port range for components on the Ports page of the Application Server instance home page. You can, however, change the port values of the Application Server Control components only from the command line because the Application Server Control user interface does not permit you to do so. You can change the following Application Server Control ports associated with your Oracle Application Server instance:

- Management Agent (Oracle home in your application server) port
- Application Server Control Console port
- OC4J Remote Method Invocation (RMI) port used by the Application Server Control OC4J instance

Changing Oracle Enterprise Manager 10g Port Values

```
$ cd /<ORACLE_HOME>/bin
```

```
$ ./emctl stop iasconsole
```

```
$ emctl config [agent port | iasconsole {port  
| rmiport}] port_number
```

```
$ ./emctl start iasconsole
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Changing Oracle Enterprise Manager 10g Port Values (continued)

To modify the ports used by the Management Agent, the Application Server Control Console, or the Application Server Control Console RMI by using the enhanced `emctl` script, perform the following steps:

1. Change directory to the `bin` directory in Oracle Application Server Home.
2. Execute the command to stop Application Server Control.
3. Use the command to change one of the Oracle Enterprise Manager 10g port values.
4. Start Application Server Control.

This script checks the status of the applicable components and stops them, if necessary, before the modification is made. When the agent port is changed, it is also necessary to stop Oracle Application Server Control.

Oracle Process Manager and Notification Server (OPMN)

- **Oracle Process Manager and Notification Server (OPMN) is the centralized process management mechanism of Oracle Application Server.**
- **OPMN manages all Oracle Application Server component processes, except OracleAS Metadata Repository or Application Server Control.**
- **OPMN consists of:**
 - **Oracle Process Manager**
 - **Oracle Notification Server**
 - **PM Modules**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Process Manager and Notification Server (OPMN)

OPMN consists of the following components that interpret and convey notification information that is sent between Oracle Application Server processes within the same or different OPMN servers:

- **Oracle Process Manager (PM)** is the centralized process management mechanism to manage Oracle Application Server processes. It starts, stops, restarts, and detects death of these processes. The Oracle Application Server processes that PM is configured to manage are specified in the `opmn.xml` file.
- **Oracle Notification Server (ONS)** is the transport mechanism for failure, recovery, startup, and other related notifications between components in Oracle Application Server.
- **PM Modules** implement the Oracle Application Server component-specific process management functionality. The PM Modules pass notification information returned by other Oracle Application Server component PM Modules within the same or different OPMN servers.

The PM uses the ONS to:

- Detect that a process has completed initialization and is ready to receive requests
- Determine what ports are in use
- Obtain component-specific run-time information

Application Server Control also uses PM to manage processes.

opmnctl Command

- **opmnctl** is the command-line interface of OPMN.
- Use Application Server Control or the **opmnctl** command-line utility to start or stop Oracle Application Server components.
- Examples of some **opmnctl** commands:

Purpose	Command
Status of all the managed processes	<code>opmnctl status</code>
Start the opmn process	<code>opmnctl start</code>
Start opmn and the managed processes	<code>opmnctl startall</code>
Stop opmn and the managed processes	<code>opmnctl stopall</code>
Start Oracle HTTP Server	<code>opmnctl startproc process-type=HTTP_Server</code>

ORACLE

Copyright © 2005, Oracle. All rights reserved.

opmnctl Command

opmnctl is the command-line utility for OPMN. The **opmnctl** command is located in the `ORACLE_HOME/opmn/bin` directory.

To get the status of the processes that are managed by OPMN, use the following command:

```
$ ./opmnctl status
```

opmnctl Command (continued)

The following table lists the processes that are running on `ias-instance` @ `host`:

Processes in Instance: `portal.edrsr16p1.us.oracle.com`

ias-component	process-type	pid	status

DSA	DSA	N/A	Down
LogLoader	logloaderd	N/A	Down
dcm-daemon	dcm-daemon	948	Alive
OC4J	home	6860	Alive
OC4J	OC4J_Portal	6861	Alive
WebCache	WebCache	N/A	Down
WebCache	WebCacheAdmin	N/A	Down
HTTP_Server	HTTP_Server	6819	Alive

In this table:

- `ias-component` represents the Oracle Application Server component
- `process-type` is a subcomponent of the `ias-component` entry. `process-type` defines the type of process to be run in association with a specific PM module.
- `pid` is the operating system integer value given for the process ID
- `status` shows the managed process status

Typical Startup Sequence

The following is a typical order to start up all instances:

1. Start the OracleAS Metadata Repository listener.
2. Start the OracleAS Metadata Repository database.
3. Use `opmnctl` to start the OracleAS Infrastructure instance.
4. Use `emctl` to start Application Server Control of the OracleAS Infrastructure instance.
5. You can start the components from the Application Server Control Console or use `opmnctl` from each OracleAS Middle Tier instance to start the processes.
6. Use `emctl` from each OracleAS Middle Tier instance to start Application Server Control.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Typical Startup Sequence

Generally, each installation, based on the functionality implemented and its operational controls, would require its own startup and shutdown sequences. However, it would be better to remember some important interdependencies of the products.

OracleAS Infrastructure functions as a service provider for the middle-tier installations. Therefore, it should be started before any middle tier that depends on the OracleAS Infrastructure services, such as single sign-on and clustering.

The slide describes a very typical startup sequence.

The following are the commands for a typical startup sequence:

1. Start the OracleAS Metadata Repository listener:
`$ lsnrctl start <listener_name>`
2. Start the OracleAS Metadata Repository database:
`$ sqlplus /nolog`
`SQL> connect username/password as SYSDBA`
`SQL> STARTUP`
3. Use `opmnctl` to start the OracleAS Infrastructure instance:
`$./opmnctl start`

Typical Startup Sequence (continued)

4. Use `emctl` to start Application Server Control of the OracleAS Infrastructure instance:

```
$ ./emctl start iasconsole
```

5. Use `emctl` from each OracleAS Middle Tier instance to start Application Server Control:

```
$ ./emctl start iasconsole
```

6. You can start the components from the Application Server Control Console or use `opmnctl` from each OracleAS Middle Tier instance to start the processes:

```
$ ./opmnctl startproc <process-type=name of process>
```

Note: Use the following command to start an entire application server instance that includes the infrastructure and the middle tier:

```
$ORACLE_HOME/bin/runstartupconsole.sh start all
```

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Typical Shutdown Sequence

The following is a typical order to shut down all instances:

1. Use `emctl` from each OracleAS Middle Tier instance to stop Application Server Control.
2. Use `opmnctl` from each OracleAS Middle Tier instance to stop the processes.
3. Use `emctl` to stop Application Server Control of the OracleAS Infrastructure instance.
4. Use `opmnctl` to stop the OracleAS Infrastructure instance.
5. Stop the OracleAS Metadata Repository database.
6. Stop the OracleAS Metadata Repository listener.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

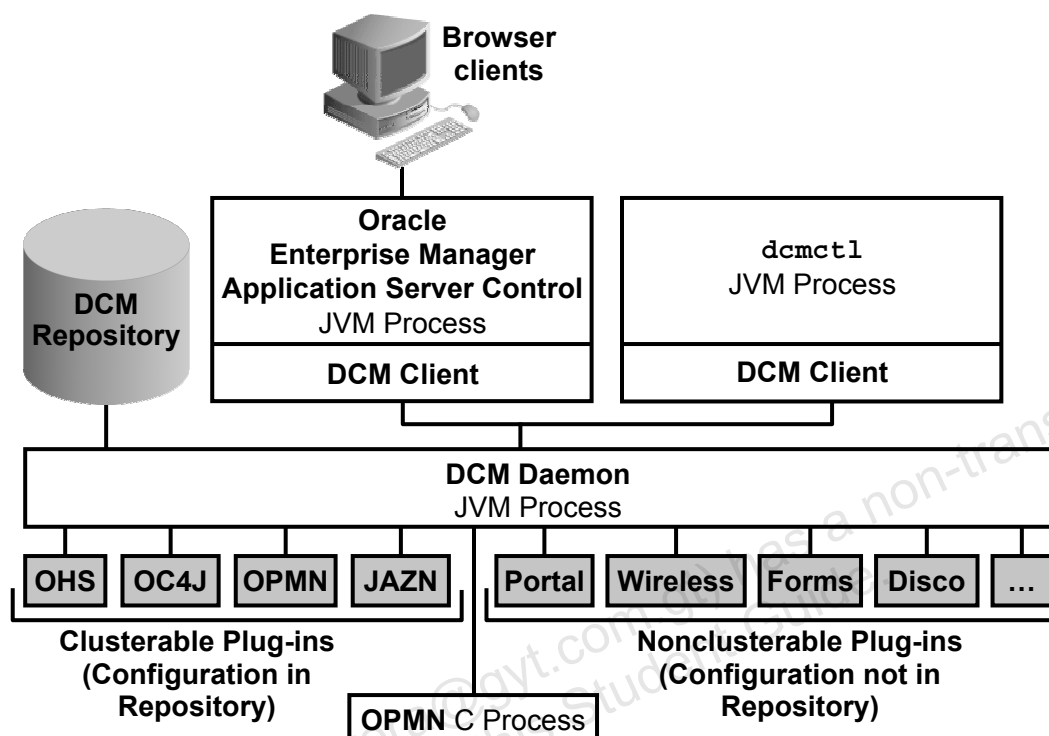
Typical Shutdown Sequence

The slide describes the sequence of shutting down the services and components of Oracle Application Server.

Note: Use the following command to stop an entire application server instance that includes the infrastructure and the middle tier:

```
$ORACLE_HOME/bin/runstartupconsole.sh stop all
```

Distributed Configuration Management



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Distributed Configuration Management (DCM)

Distributed Configuration Management is a management framework that enables you to manage the configurations of multiple Oracle Application Server instances.

DCM enables you to:

- Manage clusters and farms of Oracle Application Server instances
- Manage the configuration of individual components, such as Oracle HTTP Server instances, OC4J instances, OPMN, or Java Authentication and Authorization Service
- Perform clusterwide OC4J application deployment
- Manage versions of configurations

DCM is implemented as DCM daemon and has two interfaces:

- Application Server Control for browser clients
- The `dcmctl` utility for command-line operations

The DCM daemon is associated with two types of plug-ins:

- **Clusterable:** For Oracle HTTP Server, OC4J, and OPMN
- **Nonclusterable:** For other application server components

DCM and Metadata Repository

- The DCM repository contains the following:
 - Topology information about Oracle Application Server instances, clusters, and farms
 - Configuration files for Oracle HTTP Server, OC4J, OPMN, and Java Authentication and Authorization Service
 - Deployed J2EE applications
- The DCM repository is stored in two ways:
 - Database: In OracleAS Metadata Repository as the DCM schema
 - File based: In the file structure of the middle-tier instance
- You can access either type of repository by using the `dcmctl` utility.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

DCM and Metadata Repository

There are two types of DCM repositories, depending on the type of installation chosen for the Oracle Application Server instance:

- A database repository comprising the DCM schema is kept in OracleAS Metadata Repository.
- A file-based DCM repository is kept in a file structure in the `ORACLE_HOME/dcm/repository` directory. When a middle-tier instance is registered with an OracleAS Metadata Repository, the database repository is used as the repository.

Note: The DCM metadata repository also contains the following files:

- **app.bom:** Stores the names and versions of application objects deployed to an instance
- **conf.bom:** Stores the names and versions of configuration objects in an instance
- **cluster.bom:** Stores information about the cluster to which an instance belongs
- **tp.bom:** Stores information about the topology of an instance

Using dcmctl

- You can use the `dcmctl` command-line utility to manually manage configuration of your instance.
- You can use `dcmctl` to implement scripted control of your instance.
- Examples of `dcmctl` commands:

Purpose	Command
List instance components	<code>dcmctl listcomponents</code>
Refresh configuration information from metadata repository	<code>dcmctl resyncinstance</code>
Refresh configuration information to the metadata repository	<code>dcmctl updateconfig</code>
Create OC4J instance (of name <code>oc4j_test</code>)	<code>dcmctl createcomponent -ct oc4j -co oc4j_test</code>

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using dcmctl

The `dcmctl` utility is available in the `ORACLE_HOME/dcm/bin` directory in each Oracle Application Server installation.

Before you use the `dcmctl` utility, note the following:

- You must restart Oracle Application Server after you use `dcmctl` commands to manage clusters and farms.
- You must log in to the operating system with the username that was used to install Oracle Application Server in order to use `dcmctl`.
- The `dcmctl` commands operate on the instance in which the `dcmctl` executable is located. The value of the `ORACLE_HOME` environment variable does not determine the instance on which `dcmctl` operates. Ensure that you issue `dcmctl` commands in the Oracle home of the instance that you want to manage.
- The `dcmctl` commands and options are not case sensitive. Instance, component, and cluster names are case sensitive.

Using dcmctl in Batch Mode

- The **dcmctl** utility can be used to execute multiple command in a batch mode:
`dcmctl shell -f <script_file_name>`
- The batch mode of the **dcmctl** utility can be used to perform the following noninteractively:
 - Deploy applications and validate EAR files.
 - Archive instance configuration and deployed applications.
 - Restore the instance to a specific configuration.
- For more information about how to use **dcmctl** in batch mode, refer to the Oracle Application Server documentation set.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using the dcmctl Shell

You can execute **dcmctl** commands from within the **dcmctl** shell. Within the shell, it is not necessary to preface commands with **dcmctl**. To start the **dcmctl** shell, enter:

```
dcmctl shell
```

The following is the content of a script to create an application server cluster, join an Oracle Application Server instance to the cluster, create an OC4J instance in the cluster, deploy an application to the clustered instance, and stop the shell:

```
$ cat create_n_deploy.sh
createcluster -cl testcluster
joincluster -cl testcluster
start -cl testcluster
createcomponent -ct oc4j -co component1
start -co component1
deployapplication -f /stage/apps/app1.ear -a app1 -co
component1
exit
```

To execute the `create_n_deploy.sh` script, you can use the following command:

```
dcmctl shell -f create_n_deploy.sh
```

Management Tasks: Tools

	Application Server Control	dcmctl	opmnctl
Start/stop/restart instance and components	Yes		Yes
Start/stop/restart clusters	Yes		Yes
Create OC4J instance	Yes	Yes	
Create/join clusters	Yes	Yes	
Deploy/undeploy/redeploy applications	Yes	Yes	
Enable/disable components	Yes		
Status (up/down) of instance and components	Yes		Yes
Archive and restore versions of configurations		Yes	
Configure installed (but unconfigured) components	Yes		

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Management Tasks: Tools

The slide summarizes the appropriate tools for the management tasks.

Summary

In this lesson, you should have learned how to:

- **Compare Grid Control with Application Server Control**
- **Start and stop an Oracle Application Server instance or a component by using:**
 - **Application Server Control**
 - **Oracle Process Manager and Notification Server (OPMN)**
- **Use the `dcmctl` utility to obtain configuration information**
- **View, monitor, and control your application server processes by using the Topology Viewer**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Summary

- **View and explain all performance metrics being monitored**
- **Change the Oracle Enterprise Manager 10g port values**
- **Query from the diagnostic message database repository**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

6

Configuring and Managing Oracle HTTP Server

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Explain the Oracle HTTP Server processing model**
- **Describe the Oracle HTTP Server modules**
- **Configure and manage Oracle HTTP Server by using Oracle Application Server to:**
 - **Specify the server and file locations**
 - **Control the number of processes and connections**
 - **Manage network connections**
 - **Configure and use server log files**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Introduction to Oracle HTTP Server

Oracle HTTP Server provides a robust, reliable Web server, which:

- **Is based on Apache**
- **Serves static and dynamic content**
- **Supports content generation in many languages, such as Java, C, C++, PHP, PERL, or PL/SQL**
- **Is easily integrated with Oracle clustering, single sign-on, or Web Cache**
- **Offers plug-ins for the integration of Oracle Application Server with non-Oracle HTTP servers**

ORACLE

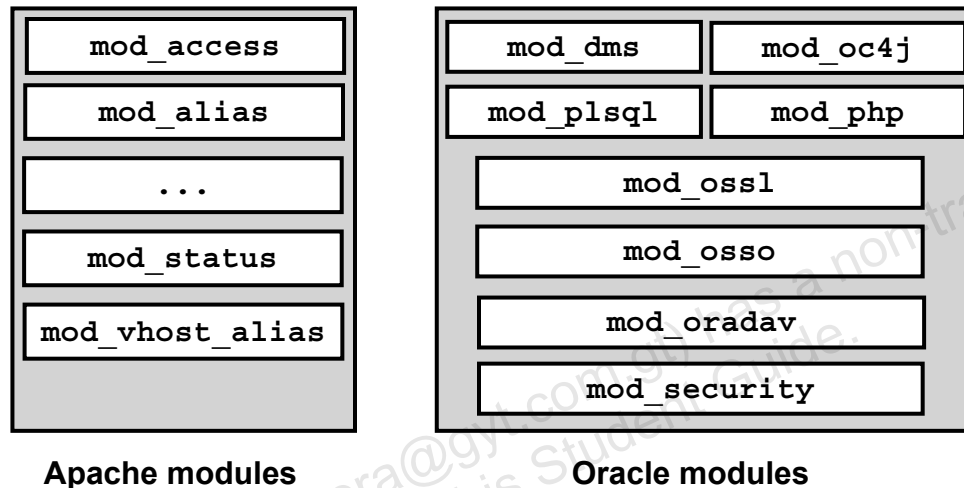
Copyright © 2005, Oracle. All rights reserved.

Overview

Oracle HTTP Server is based on the Apache Web server. It serves both static and dynamic content, and supports applications developed in Java, C, C++, PHP, PERL, or PL/SQL. Oracle HTTP Server supports single sign-on, clustered deployment and high availability, and Web Cache. In addition, plug-ins that are available as separate components enable integration of Oracle Application Server with non-Oracle HTTP servers.

Oracle HTTP Server Modules

Oracle HTTP Server extends the standard Apache distribution.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle HTTP Server Modules

One of the strengths of Apache is its modular structure. Only a core set of features exist within the main Apache executable. Everything else is provided by modules. Modules (mods) are dynamic shared objects loaded into Oracle HTTP Server that extend its functionality either by offering native services (for example, `mod_access`) or by dispatching requests to external processes (for example, `mod_oc4j` dispatching to OC4J JVMs). In addition to the compiled Apache mods provided by Oracle HTTP Server, several of the standard mods have been enhanced and Oracle-specific mods have been added.

For more information about Apache modules, see the Apache Web site at <http://www.apache.org/docs/mod/index.html>.

The Oracle-specific modules are the following:

- **mod_dms:** Enables you to monitor the performance of site components with Oracle's Dynamic Monitoring Service
- **mod_oc4j:** Routes requests from Oracle HTTP Server to Oracle Application Server Containers for J2EE (OC4J), providing HTTP for communication with the servlet engine. `mod_oc4j` is covered in detail later in this lesson.
- **mod_php:** Oracle Application Server 10g supports PHP version 4.3 (php-4.3.4), which is included in the release CDs. On Oracle HTTP Server, PHP support is provided through `mod_php` (a type of the Oracle HTTP Server module).

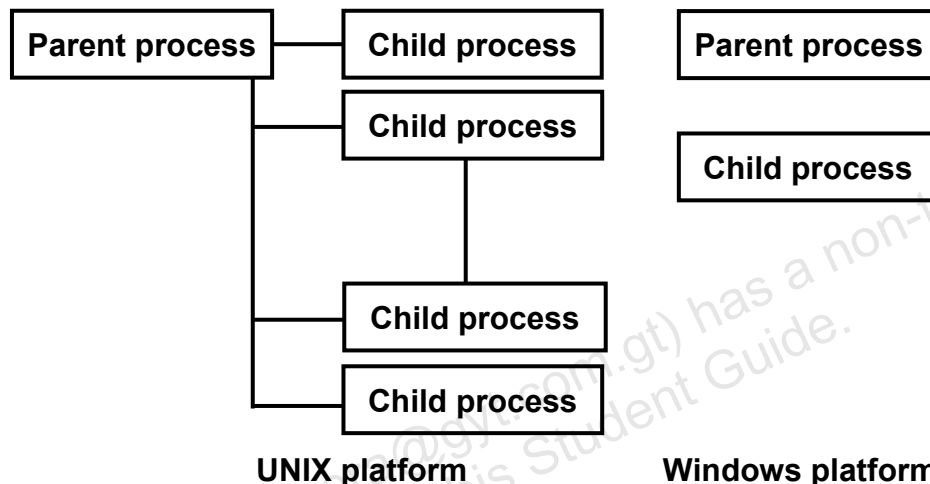
Oracle Application Server 10g R2: Administration I 6-4

Oracle HTTP Server Modules (continued)

- **mod_oradav:** Provides distributed authoring and versioning capability to Oracle HTTP Server. `mod_oradav` is based on `mod_dav`, the Apache Group's native implementation of the WebDAV specification. WebDAV is a protocol extension to HTTP 1.1 for managing Web content for multiple authors, enabling them to check out, edit, and check in files.
- **mod_oss1:** Enables strong cryptography for Oracle HTTP Server
- **mod_osso:** Provides single sign-on for Oracle HTTP Server. It examines the incoming requests and determines whether the resource requested is protected and, if so, retrieves the Oracle HTTP Server cookie for the user.
- **mod_plsql:** Enables you to create Web applications using Oracle stored procedures by connecting Oracle HTTP Server to the PL/SQL Gateway.
- **mod_security:** Is an open source engine that detects intrusion and acts as a prevention engine for Web applications. It is embedded into the Web server, and it operates as an Apache Web server module. It shields applications from intrusion attempts (such as cross-site scripting and attacks) and, thus, increases the security of Web applications.

Oracle HTTP Server Processing Model

The `httpd.pid` file contains the process ID for the parent process.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle HTTP Server Processing Model

After Oracle HTTP Server is started, the system is ready to listen and respond to requests. The request-processing model is different for Windows and UNIX. On Windows, the child processes are threads of a single child process.

On UNIX, when Oracle HTTP Server is started, a single control process launches several child processes that listen for and promptly respond to client requests. Each new process that is created in this way is a copy of the original Apache process. On UNIX, the process ID of the parent process is stored in the `httpd.pid` file that is located in the `$ORACLE_HOME/Apache/Apache/logs` directory by default. The main `httpd` parent process continues to run as the `root` user, but the child processes run as a less-privileged user. The `User` and `Group` directives are used to set the privileges for the child processes. The child processes must be able to read all the content that is served.

On Windows, Oracle HTTP Server launches a single control process and a single child process. The child process creates multiple threads that listen and respond to client requests.

Managing Processes and Connections

- **On UNIX and Linux:**
 - **StartServers**
 - **MaxClients**
 - **MaxSpareServers / MinSpareServers**
 - **MaxRequestsPerChild**
- **On Windows NT:**
 - **ThreadsPerChild**
- **On all operating systems:**
 - **KeepAlive**
 - **KeepAliveTimeout**
 - **MaxKeepAliveRequests**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing Processes and Connections

To control the number of processes on a UNIX or on a Linux system, use the following server-level directives from “Section 1: Global Environment” in your `httpd.conf` file:

- **StartServers:** Sets the number of child server processes created when Oracle HTTP Server is started. The default value is 5.
- **MaxClients:** Limits the number of requests that are handled simultaneously. The default value is 150.
- **MaxRequestsPerChild:** Each child version handles this number of requests and then dies. If the value is 0, the process lasts until the machine is rebooted. The default value is 30.
- **MaxSpareServers:** No more than this number of child servers should be left running and unused. The default value is 10.
- **MinSpareServers:** At least this number of child servers should be kept functional. The default value is 5. If fewer than this number exist, new ones are started at an increasing rate each second until the rate defined by `MAX_SPAWN_RATE` is reached (default: 32).

For example, if the number of requests to be handled simultaneously is 100, and the number of child server processes created is 10, then the maximum number of requests to be handled per child can be 10.

Oracle Application Server 10g R2: Administration I 6-7

Managing Processes and Connections (continued)

Controlling the Number of Processes and Connections

The Oracle HTTP Server implementation on Windows is multithreaded. The server handles each request internally rather than generating another instance of the `httpd` program.

The `ThreadsPerChild` directive limits the number of requests that are handled simultaneously.

The following directives can be applied to all operating systems:

- **KeepAlive:** HTTP is stateless, which means that each request and response pair between a Web browser and a server is independent. For example, if you visit a Web page that contains three embedded images, then your browser makes four separate connections to that Web server: one for the page itself and one for each of the images in turn. `KeepAlive` provides a persistent connection between the browser and the server so that the same connection can handle multiple requests and response pairs. The result is a drop in latency, or the time consumed by establishing a connection. When using Oracle Application Server Clusters, set `KeepAlive` to off.
- **KeepAliveTimeout:** This sets the number of seconds the server waits for a subsequent request before closing the connection. After a request has been received, the timeout value specified by the `Timeout` directive applies.
- **MaxKeepAliveRequests:** This limits the number of requests allowed per connection when `KeepAlive` is on. If it is set to 0, unlimited requests are allowed.

Starting, Stopping, and Restarting Oracle HTTP Server

The screenshot displays the Oracle Application Server 10g R2 Administration Console. The main window is titled "HTTP_Server" and has tabs for "Home", "Virtual Hosts", and "Administration". The "General" tab is selected, showing the status of the HTTP Server as "Up" with a start time of "Sep 6, 2005 4:41:01 AM". There are "Stop" and "Restart" buttons. Below the status, there are two pie charts showing resource usage: "Application Server (3%)", "Idle (33%)", and "Other (64%)" on the left; and "Application Server (55% 54)", "Free (1% 11MB)", and "Other (44% 444MB)" on the right. The "System Components" section at the bottom has buttons for "Start", "Stop", "Restart", and "Delete OC4J Instance". It also includes a table with columns for "Select Name", "Status", "Start Time", and "CPU Usage (%)".

Select Name	Status	Start Time	CPU Usage (%)
<input type="checkbox"/> home	↑	Sep 6, 2005 4:41:03 AM	0.00
<input type="checkbox"/> HTTP Server	↑	Sep 6, 2005 4:41:01 AM	0.59
<input type="checkbox"/> OC4J Portal	↑	Sep 6, 2005 4:41:03 AM	0.00

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting, Stopping, and Restarting Oracle HTTP Server

The Oracle Application Server components, such as Oracle HTTP Server, can be started using Application Server Control. You can start Oracle HTTP Server from the following:

- System Components table region
- Oracle HTTP Server home page (as shown partly in the slide)

The Oracle HTTP Server home page enables you to configure Oracle HTTP Server. You can modify directives, change log properties, specify a port for a listener, manage client requests, and edit server configuration files as explained in this lesson.

Starting and Stopping Oracle HTTP Server Manually

- Oracle HTTP Server is managed by OPMN.
- To start and stop Oracle HTTP Server, run:

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

- To obtain status information, run:

```
$> opmnctl status
```

ORACLE®

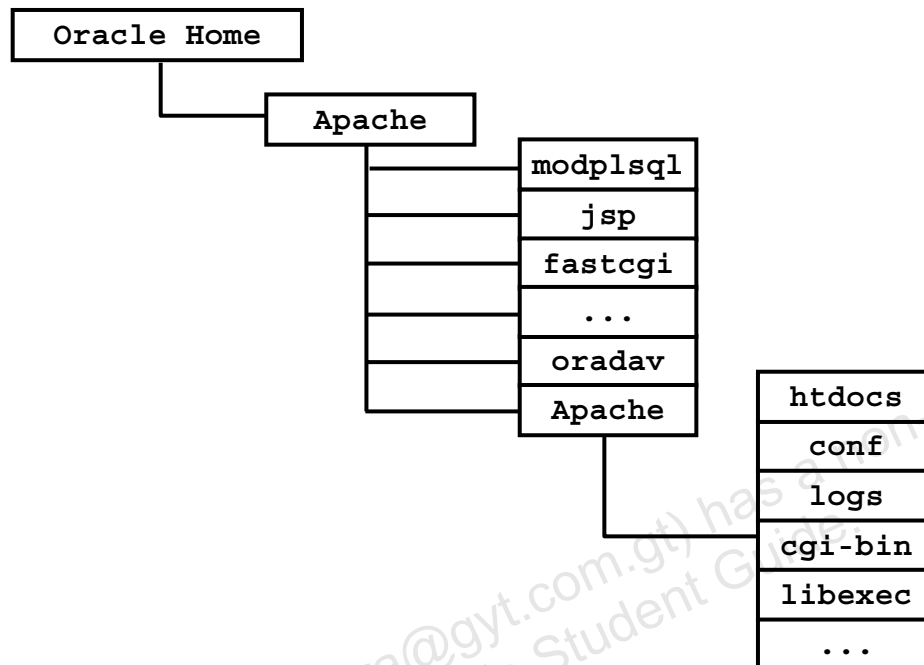
Copyright © 2005, Oracle. All rights reserved.

Starting and Stopping Oracle HTTP Server Manually

Oracle HTTP Server is managed by OPMN, which manages the Oracle Application Server processes. You can use `opmnctl` to start, stop, and restart Oracle HTTP Server. Change the directory to `$ORACLE_HOME/opmn/bin` before using the `opmnctl` commands.

- To start the Oracle HTTP Server process in the local instance:
\$> `./opmnctl startproc process-type=HTTP_Server`
- To stop the Oracle HTTP Server process:
\$> `./opmnctl stopproc process-type=HTTP_Server`
- To determine the state of Oracle HTTP Server:
\$> `./opmnctl status`
- To restart Oracle HTTP Server:
\$ `./opmnctl restartproc process-type=HTTP_server`

Directory Structure



Copyright © 2005, Oracle. All rights reserved.

Directory Structure

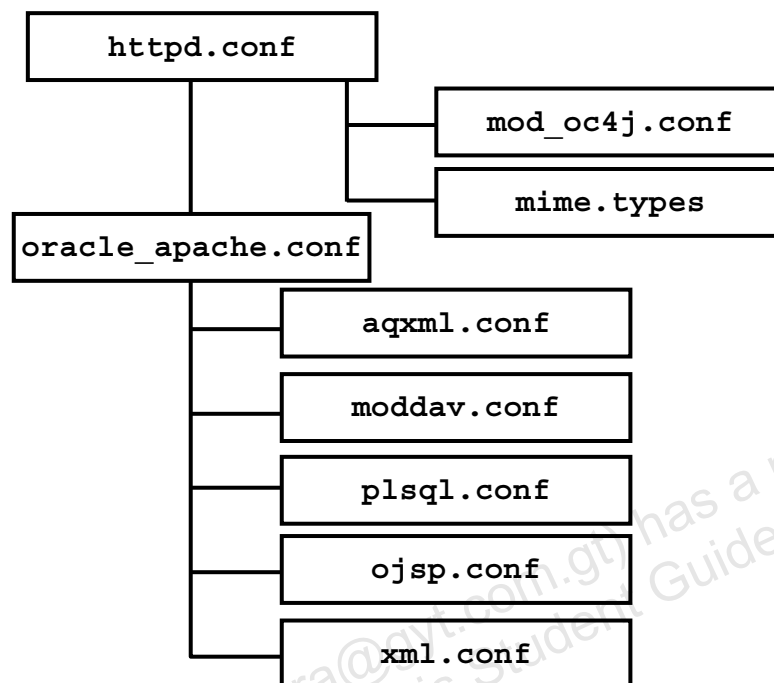
Oracle HTTP Server is installed in the `$ORACLE_HOME/Apache` directory and subdirectories for configuring modules. For example, the `modplsql` folder contains the subdirectories that are necessary to configure and run PL/SQL applications.

Apache

This is the base directory of the Apache server. It has the following subdirectories:

- **htdocs:** Contains HTML scripts. The `htdocs` directory and its subdirectories are accessible to anyone on the Web and, therefore, pose a severe security risk, if used for anything other than data that is available to public.
- **conf:** Contains configuration files
- **logs:** Contains log data, for both accesses and errors
- **cgi-bin:** Contains CGI scripts. These are programs or shell scripts that can be executed by Oracle HTTP Server on behalf of its clients.

Oracle HTTP Server Configuration Files



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle HTTP Server Configuration Files

The main configuration file is `httpd.conf`. The `httpd.conf` file includes the following:

- Reference to other configuration files, such as `oracle_apache.conf`
- Directives to include the configuration files

The following is a description of some of the configuration files:

- **`httpd.conf`:** Is the main configuration file. This contains directives and pointers to other configuration files, such as:
 - **`mod_oc4j.conf`:** Configures and loads the `mod_oc4j` module. The `mod_oc4j` Oracle module routes requests from Oracle HTTP Server to OracleAS Containers for J2EE (OC4J) and, therefore, contains routing information. This module is enabled by default.
 - **`mime.types`:** Controls the Internet media types that are sent to the client for the given file extensions. Sending the correct media type to the client is important so that the client knows how to handle the content of the file. You can add extra types in the MIME type file or add an `AddType` directive in the configuration file. For more information about working with MIME types, see the Web site at http://www.apache.org/docs/mod/mod_mime.html.

Oracle HTTP Server Configuration Files (continued)

- **oracle_apache.conf:** Is included in the main configuration file to store configuration files of supported modules:
 - **aqxml.conf:** Enables and configures Advanced Queuing
 - **moddav.conf:** Configures and loads the mod_oradav module to enable distributed authoring and versioning of Web documents
 - **plsql.conf:** Configures and loads the PL/SQL module. The file is located in the \$ORACLE_HOME/Apache/modplsql/conf directory.
 - **ojsp.conf:** Configures JavaServer Pages. The file is located in the \$ORACLE_HOME/Apache/jsp/conf directory.
 - **xml.conf:** Associates the .xsql extension with the XSQL servlet. The file is located in the \$ORACLE_HOME/xdk/admin directory.

The following example of the oracle_apache.conf file explains how configuration files are included and where to change the path information or the name:

```
# Advanced Queuing - AQ XML
include "/export/home0/ias20/rdbms/demo/aqxml.conf"
#
#Directives needed for OraDAV module
include "/export/home0/ias20/Apache/oradav/conf/moddav.conf"
include "/export/home0/ias20/Apache/modplsql/conf/plsql.conf"
include "/export/home0/ias20/Apache/jsp/conf/ojsp.conf"
#
include "/export/home0/ias20/xdk/admin/xml.conf"
#
```

Specifying File Locations

- The following directives control the location of the various server files and can be specified in the server configuration context:
 - `ServerRoot`
 - `PidFile`
 - `CoreDumpDirectory`
- The following directives can be used in the server configuration and virtual host contexts:
 - `DocumentRoot`
 - `ErrorLog`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specifying File Locations

- **ServerRoot:** This directive specifies the main directory where Apache stores its log files, configuration files, and HTML documents. Other directives, which are defined with a relative path, use the defined `ServerRoot` path as the default root to extend their relative paths. You should not change this directive.
- **PidFile:** Enables you to set and change the location of the PID file where the server records the process identification number. If the file name does not begin with a slash (/), then it is assumed to be relative to the `ServerRoot` directory.
- **CoreDumpDirectory:** Specifies the directory where the server stores core dumps. The default is the `ServerRoot` directory. This directive is applicable only to Linux/UNIX.
- **DocumentRoot:** Defines the directory from which Apache serves files. It can use any valid directory that is accessible to the server, even on another computer over network file system (NFS). The default setting in `httpd.conf` is `htdocs`, a directory relative to the server root directory. It is very common to change the setting of `DocumentRoot` to keep sensitive configuration information away from your public pages. You can specify either a relative path, which Apache looks up under the server root, or an explicit path that can be outside the server root.

Specifying File Locations (continued)

- **ErrorLog:** Sets the name of the file where Oracle HTTP Server logs any errors that it encounters. The default is `logs/error_log`. The most common log files are the access log and error log. The name of the error log either is an explicit path name starting with a slash (/) or is relative to the server root directory (the default).

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.


Oracle HTTP Server Home Page

HTTP_Server

[Home](#) [Virtual Hosts](#) [Administration](#)

Page Refreshed Sep 6, 2005

General



Status **Up**
Start Time **Sep 6, 2005 4:41:01 AM**

Default Server Configuration

Server Name **EDRSR16P1**
Document Root **/home/oracle/portal/Apache/**
Last Modification **Sep 6, 2005 4:39:04 AM**

Status

CPU Usage (%)	0.75
Memory Usage (MB)	124.16
Error Rate (%)	0.00
Active Connections	1
Connection Open Time (seconds)	7.31

Response and Load

Active Requests	1
Request Throughput (requests per second)	2.25
Request Processing Time (seconds)	0.02
Data Throughput (KB per second)	10.04
Data Processed (KB per request)	4.46

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle HTTP Server Home Page

The Oracle HTTP Server home page is the entry point in Application Server Control for managing and configuring Oracle HTTP Server. You can access the Oracle HTTP Server home page from the Oracle Application Server Instance home page of Application Server Control. Click the Oracle HTTP Server link in the components table of the Oracle Application Server instance page and the Oracle HTTP Server page is launched.

The Oracle HTTP Server page has three property pages: Home, Virtual Hosts, and Administration. The first interface to be invoked is the home page. Accordingly, the Oracle HTTP Server page is also referred to as the Oracle HTTP Server home page.

You can get the overall status of Oracle HTTP Server from the home page. Also, you can monitor the performance, and drill down through the metrics using the links in the Performance section. You can also use the home page to stop, start, or restart Oracle HTTP Server.

You can use the VirtualHosts property page to manage the virtual hosts that are configured with Oracle HTTP Server.

You can use the Administration property page to configure server properties (such as document root, ports, and number of processes) and to edit configuration files.

Configuring Oracle HTTP Server

- **Directives are used to configure Oracle HTTP Server to meet your needs.**
- **The server-level configuration directives apply to Oracle HTTP Server globally.**
- **The container directives create a limited scope for the directives that are defined within them.**
- **Per-directory configuration enables the server to act like a container with directory scope in the main configuration files. The default name for the per-directory configuration file is `.htaccess`.**
- **The configuration tiers are applied hierarchically.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Oracle HTTP Server

The directives that belong to the main Apache core are defined in `httpd.conf` or in application-specific configuration files. The configuration is logically divided as follows:

- The server-level configuration directives include those that make sense only in a global context, such as `StartServers`, or those where you want a default setting that can be overridden by container or per-directory directives, such as `ServerAdmin`.
- The container directives are used to modify the server-level configuration directives for the area of effect of the container. The main purpose of a container is to allow Oracle HTTP Server to include or ignore a given directive, depending on whether it is applicable to the scope defined by the container.
- Per-directory configuration is optional. The default name for the per-directory configuration file is `.htaccess`. The configuration files are located in the directories under `DocumentRoot`. Per-directory directives work like container directives. The `AllowOverride` directive allows to restrict the use of directives. The configuration tiers are applied hierarchically; each directive overrides the directive in the tier above it.

Controlling Access to the Application Server

Server and server administrator options can be set based on the main server or a virtual host:

```
Listen
UseCanonicalName On
ServerName
Port
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Setting Server and Administrator Functions

The following are the basic directives:

- **Listen:** This directive sets the server to use more than one IP address or port. By default, the server listens to requests on every IP address and the port number specified by the `Port` directive. If you specify a port, Oracle HTTP Server receives connection on that port for all installed network interfaces. To limit the scope to a specific address, use `IP:port`.
- **UseCanonicalName:** This directive indicates the host name and port to use when redirecting the URL to the same server. This forces Oracle HTTP Server to use `ServerName` and `Port` from the server configuration instead of using the `ServerName` and `Port` combination through the HTTP 1.1 header.
 - **on:** This is the default setting. For this setting, the server uses the host name and port values set in `ServerName` and `Port`.
 - **off:** For this setting, the server uses the host name and port that the user specifies in the request.
- **ServerName:** This directive defines the host name that the server uses when creating redirection URLs and for the `SERVER_NAME` variable. If you are using name-based virtual hosts, the `ServerName` directive inside a `<VirtualHost>` section specifies what host name must appear in the request's header to match this virtual host when using HTTP 1.1.

Setting Server and Administrator Functions (continued)

It is a common mistake to think that the server name is what Apache responds to, but it is the name used within responses. In fact, Oracle HTTP Server responds to any connection request on any network interface and port number to which it is configured to listen.

If you do not use virtual host containers, `ServerName` refers to the server to which Oracle HTTP Server responds. However, if you use name-based virtual hosting, then the browser must support HTTP 1.1. In this case, you must provide the `ServerPath` directive to support the HTTP 1.0 browser as well.

- **Port:** You can specify only one `Port` directive in a single configuration. If there is no `Listen` directive, the server accepts connections and redirects to the port specified in the `Port` directive. However, if there is a `Listen` directive, the port specified is used only for redirection. In such cases, the port refers to the IP and Port on the load balancer or the Web Cache that acts as the front end for Oracle HTTP Server.

Port numbers range from 0 through 65,535. Generally, on UNIX, all ports with port numbers lower than 1,024 are reserved for system use and only the root can bind a service to those ports. To use port 80, you must start the server from the root account. After binding to the port and before accepting requests, Apache changes to a low-privileged user as set by the `User` directive.

The `Port` directive is used to set the `SERVER_PORT` environment variable (for CGI) for server-side code to use in redirect requests. That is, it forms the Canonical Server name for redirect requests.

For more information about directives, refer to the Apache documentation at <http://httpd.apache.org/docs/mod/directives.html>.

Modifying Server Properties

HTTP_Server	
Home Virtual Hosts Administration	
Server Properties MIME Languages MIME Types MIME Encodings PL/SQL Properties Advanced Server Properties	Properties Inheritance HTTP Server administration each virtual host. Virtual host by the virtual host. Use the links on this page to or are common to multiple vi to override any values spec
Server Properties	
General	
Version	10.1.2
Server Root Directory	/home/oracle/portal/Apache/Apache
Configuration File	/home/oracle/portal/Apache/Apache/conf/httpd.conf
Process ID File	/home/oracle/portal/Apache/Apache/logs/httpd.pid
* Document Root	/home/oracle/portal/Apache/Apache/htdocs
Administrator E-Mail	you@your.address
User	oracle
Group	dba

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Modifying Server Properties

To modify Server Properties, perform the following steps:

1. Navigate to the Administration property page of the Oracle HTTP Server home page.
2. Click the Server Properties link. The Server Properties page is displayed.
3. You can change the Document Root field that refers to the directory from which the server serves the files. Enter a new path in the Document Root field to change the document root directory. Note that the document root directory is different from the server root directory that is used only to store the server files.
4. Enter the appropriate e-mail address in the Administrator E-Mail field. Oracle HTTP Server uses this e-mail address to issue notices and warnings.
5. User and group settings can be added or changed; both specify which privileges the child processes run with when Oracle HTTP Server is started by root.
6. Click Apply to accept the changes. If you do not click Apply, you lose your changes. If you make a mistake or want to undo any changes, click Revert. Oracle Application Server displays a confirmation page, which confirms that the appropriate configuration files have been updated.
7. Click Yes to restart Oracle HTTP Server so that the changes take effect. Click No to restart the server later.

Specifying a Listener Port

Server Properties

Page Refresh

General

Version

10.1.2

Server Root Directory

/home/oracle/portal/Apache/Apache

Configuration File

/home/oracle/portal/Apache/Apache/conf/httpd.conf

Process ID File

/home/oracle/portal/Apache/Apache/logs/httpd.pid

* Document Root

/home/oracle/portal/Apache/Apache/htdocs

Administrator E-Mail

you@your.address

User

oracle

Group

dba

Listening Addresses and Ports

Default Port

7778

Select Item and... Remove

Select All | Select None

Select Listening IP Address

Listening Port

☐

4444

☐

7779

☐

127.0.0.1

7201

Add Another Row

Port Dependencies

Changing a listening port
HTTP Server virtual host
Web Cache, Portal, Inter
one or more of these dep
required. For more inform
Configuration Dependenc

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Specifying a Listener Port

The port specification tells the server where to listen for requests. By default, the server listens on all networks, but only on one port that you specify. To specify a listener port from the Oracle HTTP Server Administration property page, perform the following steps:

1. Navigate to the Listening Addresses and Ports region of the Server Properties page.
2. Select the IP address and associated port number that you want to use for the listener, or create new port settings by clicking the Add Another Row button and making the appropriate changes. Click OK to return to the Server Properties page.
3. If applicable, select the IP address to be used as a self-referencing URL.
4. Scroll down to the end of the page, and click Apply to accept the changes. If you do not click Apply, you lose your changes. If you make a mistake or want to undo any changes, click Revert.

Oracle Application Server displays a confirmation page, which confirms that the appropriate configuration files have been updated.

5. Click Yes to restart Oracle HTTP Server so that the changes take effect. Click No to restart the server later.

For more information about using ports, refer to the *Oracle Application Server Administrator's Guide* and the *Oracle HTTP Server Administrator's Guide*.

Administrative Directives

To ensure that Oracle HTTP Server runs with appropriate privileges, you must define the following directives in your server configuration or virtual host context:

- **User**
- **Group**
- **ServerAdmin**
- **ServerTokens**

ORACLE

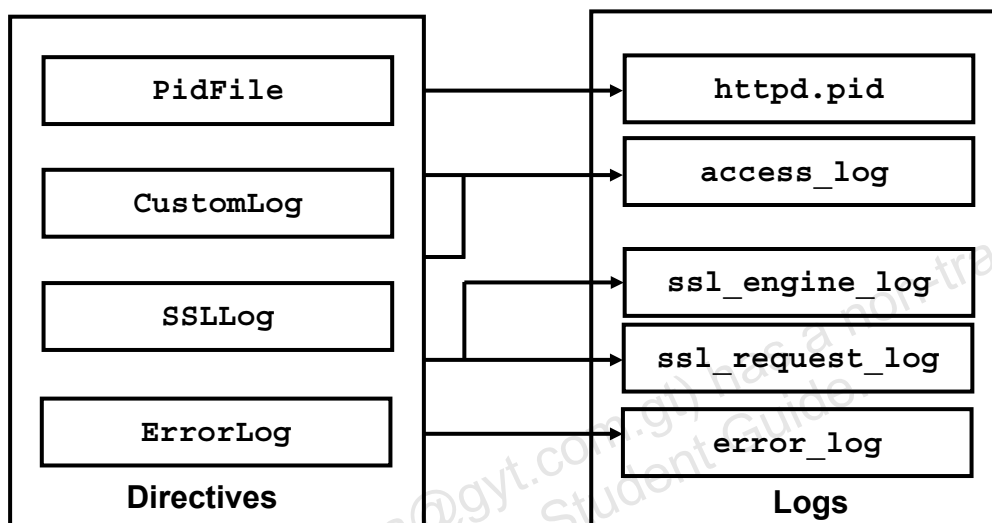
Copyright © 2005, Oracle. All rights reserved.

Administrative Directives

- **ServerAdmin:** Creates an e-mail address that is included with every error message that clients encounter. It is useful to create a separate e-mail address for this purpose.
- **ServerTokens:** Controls the server information that is returned to clients, such as in error messages. This information includes a description of the generic operating system type of the server and information about included modules:
 - **prod:** For this setting (product only), the server provides only the server name (Apache). You should prefer this setting to the default setting, which is **full**.
 - **min:** For this setting (“minimal”), the server provides the server name and version.
 - **OS:** For this setting, the server provides the server name, version, and operating system.
 - **full:** For this setting, the server provides the server name, version, operating system, and compiled modules.

Server Logs

`$ORACLE_HOME/Apache/Apache/logs`



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring and Using Server Logs

Oracle HTTP Server log files consist of:

- **httpd.pid:** On startup, Apache saves the process ID of the parent httpd process in this file. This file name can be changed with the `PidFile` directive. The process ID is used by the administrator to restart and terminate the daemon on UNIX. If the process dies (or is killed) abnormally, you must also kill the child httpd processes.
- **access_log:** The server typically logs each request in a transfer file.
- **error_log:** The server logs error messages to a log file. The file name can be set using the `ErrorLog` directive for the server configuration; different error logs can be set for different virtual hosts.

The `ErrorLog` directive sets the name of the file to which the server logs any errors that it encounters. For example, setting `ErrorLog /dev/null` effectively turns off error logging.

Configuring and Using Server Logs (continued)

- **ssl_engine_log** and **ssl_request_log**: These directives specify the location of their specific log files. These files are created when Oracle HTTP Server is started in SSL mode and the SSLLog directive is enabled in the server configuration context or for a virtual host.

The SSLLog directive can be used in the context of the server configuration as well as for virtual hosts when you want to use different locations for each virtual host.

PidFile can be used only once in the server configuration context, that is, in Section 1 or Section 2. The following is an example of how to use these directives from httpd.conf:

```
### Section 1: Global Environment
# PidFile: The file in which the server should record its
# process identification number when it starts.
#
PidFile logs/httpd.pid

### Section 2: 'Main' server configuration
# The directives in this section set up the values used by
# the 'main' server. SSLLog are not set per #default in this
# section but could be set like this:
#SSLLog      logs/ssl_engine_log

### Section 3: Virtual Hosts
#
#<VirtualHost ip.address.of.host.some_domain.com>
#   ServerAdmin webmaster@host.some_domain.com
#   DocumentRoot /www/docs/host.some_domain.com
#   ServerName host.some_domain.com
#   ErrorLog logs/host.some_domain.com-error_log
#   </VirtualHost>
```

LogLevel Directive

- The **LogLevel** directive applies to the context of server configuration and virtual hosts.
- It controls the number of messages.
- It can be set to one of the following: **Emerg, Alert, Crit, Error, Warn, Notice, Info, Or Debug.**

An example from `httpd.conf`:

```
### Section 2: 'Main' server configuration
#
ErrorLog logs/error_log
LogLevel warn
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using the LogLevel Directive

The **LogLevel** directive adjusts the verbosity of the messages that are recorded in the error logs. The following levels are available, in the order of decreasing significance:

Level	Description	Example
Emerg	Emergency: System is unusable Action must be taken immediately	“Child cannot open lock file. Exiting” Alert “getpwuid: couldn’t determine user name from uid”
Crit	Critical condition	“socket: Failed to get a socket, exiting child”
Error	Error condition	“Premature end of script headers”
Warn	Warning condition	“child process 1234 did not exit, sending another SIGHUP”
Notice	Normal but significant condition	“httpd: caught SIGBUS, attempting to dump core in ...”
Info	Informational	“Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)...”
Debug	Debug-level message	“Opening config file ...”

Log Formats

The default format is the Common Log Format (CLF):

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

- **%h: Remote host**
- **%l: Remote log name, if supplied**
- **%u: Remote user**
- **%t: Time in common log format**
- **%r: First line of request**
- **%s: Status**
- **%b: Bytes sent, excluding HTTP headers**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specifying Log Formats

The default for the log format is the standard Common Log Format (CLF). You can use the `LogFormat` directive to set a different format. Alternatively, the log file can be customized (and if multiple log files are used, each can have a different format). Custom formats are set with `LogFormat` and `CustomLog`.

`LogFormat` specifies the information included in the log file and the manner in which it is written. The CLF format is `host ident authuser date request status bytes`, where:

- `host` is the client domain name or its IP number
- `ident` is the client identity information, if `IdentityCheck` is enabled and the client machine runs `identd`
- `authuser` is the user ID for a password-protected site
- `date` is the date and time of the request in the `day/month/year:hour:minute:second` format
- `request` is the request line, in double quotation marks, from the client
- `status` is the three-digit status code returned to the client
- `bytes` is the number of bytes, excluding headers, returned to the client

Changing Error Log Properties

Logging		Related Links
Error Log Filename	<input type="text" value="/home/oracle/portal/Apache/Apache/bin/rotatelog /hom"/>	Error Rate (%) <u>0.00</u>
Error Logging Level	<input type="text" value="Warning"/>	Error Log
IP Address Translation	<input type="text" value="None"/>	Access Logs
Select Access Log and... <input type="button" value="Remove"/>		
Select Client Access Log Filename	Log Format	
<input type="text" value="/home/oracle/portal/Apache/Apache/bin/rotatelog /hom"/>	<input type="text" value="common"/>	
<input type="button" value="Add Another Row"/>		
Client Request Handling		Related Links
Maximum Requests Processed Simultaneously	<input type="text" value="150"/>	Active Processes <u>1</u>
Request Timeout (seconds)	<input type="text" value="300"/>	Active Requests <u>1</u>
<input type="checkbox"/> Limit Requests Handled by each Child Server Process		
Client Connection Handling		Related Links
<input checked="" type="checkbox"/> Allow Multiple Requests per Connection		Active Connections <u>4</u>
Connection Timeout (seconds)	<input type="text" value="15"/>	
<input checked="" type="checkbox"/> Limit Requests per Connection	<input type="text" value="100"/>	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Changing Error Log Properties

The error log file is an important source of information for maintaining a well-performing server. The error log file records all the information about problem situations so that the system administrator can diagnose and fix the problems.

Note: To provide access to this file, without providing access to the other configuration files, you may need to move the error log file to a directory that is accessible.

To customize your error log, perform the following steps:

1. Navigate to the Logging region of the Server Properties page.
2. In the Error Log Filename field, enter the full path and the file name of the file where you want the errors to be logged. A relative path name is assumed to be relative to the `ServerRoot` directory.
3. Select the error-logging level from the Error Logging Level drop-down list. The logging level indicates the severity of the error reported in the error log.
4. Set the IP Address Translation type. This setting tells the server how to handle DNS lookups.
5. Scroll down to the end of the page, and click **Apply** to accept the changes.

Adding an Access Log File

Logging		Related Links
Error Log Filename	/home/oracle/portal/Apache/Apache/bin/rotatelog /hom	Error Rate (%) <u>0.00</u>
Error Logging Level	Warning	Error Log
IP Address Translation	None	Access Logs
Select Access Log and... Remove		
Select Client Access Log Filename	Log Format	
<input type="radio"/> /home/oracle/portal/Apache/Apache/bin/rotatelog /hom	common	
Add Another Row		
Client Request Handling		
Maximum Requests Processed Simultaneously	150	Related Links
Request Timeout (seconds)	300	Active Processes <u>1</u>
<input type="checkbox"/> Limit Requests Handled by each Child Server Process		Active Requests <u>1</u>
Client Connection Handling		
<input checked="" type="checkbox"/> Allow Multiple Requests per Connection		Related Links
Connection Timeout (seconds)	15	Active Connections <u>4</u>
<input checked="" type="checkbox"/> Limit Requests per Connection	100	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Adding an Access Log File

The access log file contains a detailed list of accesses to Oracle HTTP Server. It contains the remote host name, remote log name, remote user, time, request, response code, and bytes transferred. This information can be used to generate statistical reports about the server usage patterns. To add an access log file, perform the following steps:

1. Navigate to the Logging region of the Server Properties page.
2. In the Select Access Log region, select the client access log file that you want to relocate, or click the Add Another Row button.
3. In the field provided, enter the full path name and file name of the access log file that you want to create. For example, you can enter the following location:
 /private2/ias/Apache/Apache/logs/access_log
 Or, enter the relative path and file name: logs/access_log. A relative path is assumed to be relative to the Server Root directory.
4. Set the log format; you can select to use an existing format or specify a new format by entering a new format name. For a full description of the available log formats, click Help.
5. Scroll down to the end of the page, and click Apply to accept the changes.
6. Click Yes to restart Oracle HTTP Server so that the changes take effect.

Log Rotation

Log rotation:

- Is necessary to reduce the size of log files
- Is achieved by moving or deleting existing logs
- Is achieved by using the `rotatelog`s program

```
NewLog "|bin/rotatelog /var/logs/logfile
86400" common
NewLog "|bin/rotatelog /var/logs/logfile 5M"
common
```

```
rotatelog logfile [rotatetime [offset]] |
[filesizeM]
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Log Rotation

The quantity of information stored in log files is very large. It is, therefore, necessary to periodically rotate the log files by moving or deleting the existing logs. This cannot be done while the server is running, because Apache continues writing to the old log file as long as it holds the file open. Instead, the server must be restarted after the log files are moved or deleted so that it opens new log files.

By using a graceful restart, the server can be instructed to open new log files without losing any existing or pending connections from clients. However, in order to accomplish this, the server must continue to write to the old log files while it finishes serving old requests. It is, therefore, necessary to wait for some time after the restart before processing log files. An important requirement is to rotate the log files without having to restart the server.

This is achieved through the use of piped logs. Instead of writing error and access log files directly to a file, `httpd` is also capable of writing error and access log files through a pipe to another process. To write logs to a pipe, replace the file name with the pipe character "|", followed by the name of the executable that should accept log entries on its standard input. Apache starts the piped-log process when the server starts.

Log Rotation (continued)

In the first example in the slide, create `/var/logs/logfile.nnnn`, where `nnnn` is the time at which log starts (after 24 hours). At the end of each rotation time, a new log is started.

In the second example, the configuration rotates the log file whenever it reaches a size of 5 megabytes.

The usage of the `rotatelog` command is as follows:

```
rotatelog logfile [rotationtime [offset] ] | [filesizeM]
```

where

`logfile` is the path and the base name of the log file

`rotationtime` is the time between log file rotations in seconds

`offset` is the number of minutes offset from UTC

`filesizeM` is the maximum file size in megabytes followed by the letter M to specify size instead of the time

Managing Client Requests and Connection Handling

Logging		Related Links
Error Log Filename	/home/oracle/portal/Apache/Apache/bin/rotatlogs /hom	Error Rate (%) 0.00
Error Logging Level	Warning	Error Log
IP Address Translation	None	Access Logs
Select Access Log and... Remove		
Select Client Access Log Filename	Log Format	
/home/oracle/portal/Apache/Apache/bin/rotatlogs /hom	common	
Add Another Row		
Client Request Handling		Related Links
Maximum Requests Processed Simultaneously	150	Active Processes 1
Request Timeout (seconds)	300	Active Requests 1
<input type="checkbox"/> Limit Requests Handled by each Child Server Process		
Client Connection Handling		Related Links
<input checked="" type="checkbox"/> Allow Multiple Requests per Connection		Active Connections 4
Connection Timeout (seconds)	15	
<input checked="" type="checkbox"/> Limit Requests per Connection	100	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing Client Requests and Connection Handling

On the Oracle HTTP Server home page, you can specify how child processes and connections should initialize resources during the processing phase of the server. Child processes and connection settings have an impact on the ability of the server to process requests. To maintain a well-performing server, you may need to modify these settings as the number of requests increases or decreases.

To modify the settings, perform the following steps:

1. On the Server Properties page, navigate to the Client Request Handling region.
2. Modify the child process and connections directives by changing the default values in the appropriate fields.
3. Scroll down to the end of the page, and click Apply to accept the changes.
4. Restart the server to commit the changes.

For more information about setting the client request and connection-handling parameters, refer to the *Oracle HTTP Server Administrator's Guide*.

Advanced Server Properties

HTTP_Server
[Home](#) [Virtual Hosts](#) [Administration](#)

[Server Properties](#)
[MIME Languages](#)
[MIME Types](#)
[MIME Encodings](#)
[PL/SQL Properties](#)
[Advanced Server Properties](#)

Properties Inheritance
Configuration Files

File Name	Location
httpd.conf	/home/oracle/portal/Apache/Apache/conf
oracle_apache.conf	/home/oracle/portal/Apache/Apache/conf
moddav.conf	/home/oracle/portal/Apache/oradav/conf
ajsp.conf	/home/oracle/portal/Apache/jsp/conf
mod_oc4j.conf	/home/oracle/portal/Apache/Apache/conf
cache.conf	/home/oracle/portal/Apache/modplsql/conf
portal.conf	/home/oracle/portal/portal/conf
dms.conf	/home/oracle/portal/Apache/Apache/conf
plsql.conf	/home/oracle/portal/Apache/modplsql/conf
oradav.conf	/home/oracle/portal/Apache/oradav/conf
dads.conf	/home/oracle/portal/Apache/modplsql/conf
ultrasearch.conf	/home/oracle/portal/ultrasearch/webapp/config
ssl.conf	/home/oracle/portal/Apache/Apache/conf
mod_osso.conf	/home/oracle/portal/Apache/Apache/conf
oiddas.conf	/home/oracle/portal/dap/das
agxml.conf	/home/oracle/portal/rdbms/demo
uix.conf	/home/oracle/portal/uix

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Advanced Server Properties

The Oracle HTTP Server Advanced Properties page enables you to access the Oracle HTTP Server configuration files directly. These files are used to customize the features of your server. To access one of the Oracle HTTP Server configuration files directly, perform the following steps:

1. On the Oracle HTTP Server home page, scroll down to the Administration region.
2. Click Advanced Server Properties. This takes you to the Configuration Files region of the Advanced Server Properties page.
3. Select the configuration file that you want to edit. A text editor appears.
4. Make the appropriate changes and click Save Changes. When you are finished, click OK to return to the Oracle HTTP Server home page.
5. Scroll down to the end of the page and click Apply to accept the changes. If you do not click Apply, you lose your changes. If you make a mistake or want to undo any changes, click Revert.
6. Navigate to the Oracle HTTP Server home page and restart the server to commit the changes. You do not have to restart the server if the changes have been reverted.

For more information about using the Oracle HTTP Server configuration files to customize your server settings, refer to the *Oracle HTTP Server Administrator's Guide*.

Editing Server Configuration Files

Configuration Files	
File Name	Location
httpd.conf	/home/oracle/portal/Apache/Apache/conf
oracle_apache.conf	/home/oracle/portal/Apache/Apache/conf
moddav.conf	/home/oracle/portal/Apache/oradav/conf
ojsp.conf	/home/oracle/portal/Apache/jsp/conf
mod_oc4j.conf	/home/oracle/portal/Apache/Apache/conf
cache.conf	/home/oracle/portal/Apache/modolsql/conf
portal.conf	
dms.conf	
plsql.conf	
oradav.conf	
dads.conf	
ultrasearch.conf	
ssl.conf	
mod_ossso.conf	
oiddas.conf	
aqxml.conf	
uix.conf	

Edit httpd.conf	
Configuration File	/home/oracle/portal/Apache/Apache/conf/httpd.conf
## ## httpd.conf -- Apache HTTP server configuration file ## # # Based upon the NCSA server configuration files originally by Rob McCool. # # This is the main Apache server configuration file. It contains the # configuration directives that give the server its instructions. # See <URL:http://www.apache.org/docs/> for detailed information about # the directives. # # Do NOT simply read the instructions in here without understanding # what they do. They're here only as hints or reminders. If you are unsure # consult the online docs. You have been warned.	

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Editing Configuration Files

You can use Application Server Control to perform minor configuration changes in the configuration files. The advantage of making these changes using Application Server Control is that Enterprise Manager ensures that Oracle HTTP Server is stopped and restarted to effect the configuration changes immediately.

Note that this interface does not have elaborate editing capabilities.

It is advisable to back up the original configuration files before editing them so that in case the revisions fail, you can retrieve the operational configuration.

Getting the Server Status

- **Change your `httpd.conf` file to allow access from specific IP addresses or machine:**

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 139.185.35.125
</Location>
```

- **Set the directive to show extended status to `on` or `off` in section 1 of `httpd.conf`:**

```
ExtendedStatus On
```

- **Restart Oracle HTTP Server.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Enabling the Server Status

A server administrator uses the `Status` module to find out how well a server is performing. An HTML page is presented that gives the current server statistics in an easily readable form.

To enable status reports only for browsers from the `foo.com` domain, add the following code to your `httpd.conf` configuration file:

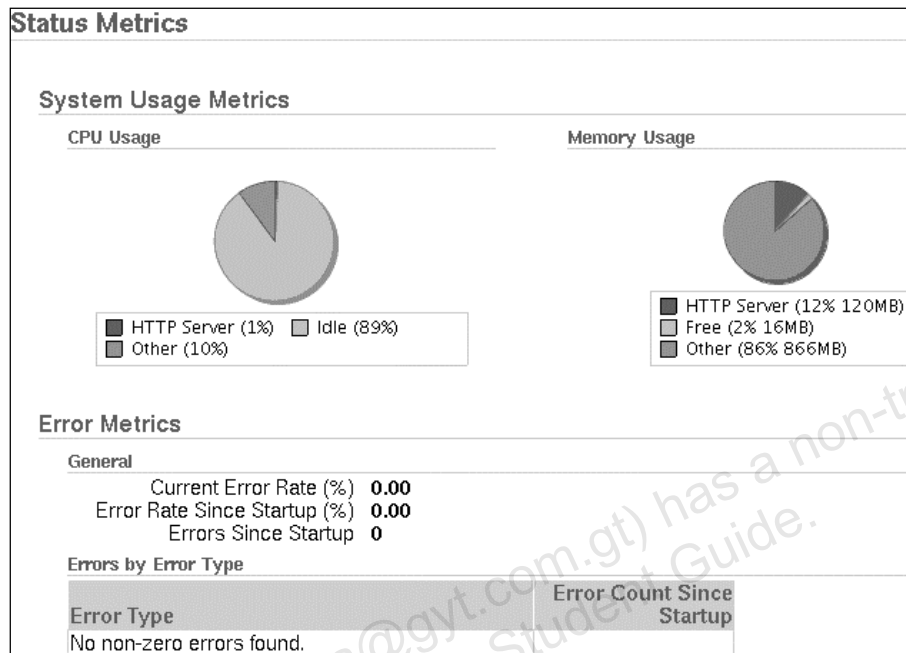
```
<Location /server-status>
    SetHandler server-status
    order deny,allow
    deny from all
    allow from .foo.com
</Location>
```

You can now access server statistics by using the following URL:

`http://<host name>:<port>/server-status`

You can get the status page to update itself automatically if you have a browser that supports the `Refresh` command. To refresh the page every *N* seconds, access the page using the URL `http://<host name>:<port>/server-status?refresh=N`.

Monitoring Oracle HTTP Server



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Monitoring Oracle HTTP Server

All Oracle Application Server components automatically generate usage and performance statistics. These statistics are periodically polled and analyzed by Oracle Application Server Administration Service. In general, metrics focus on three factors: volumes, rates, and durations. An example of how this applies to Oracle HTTP Server is monitoring Web server request activity. The aggregate Oracle HTTP Server request activity is monitored by the following metrics:

- Active Requests measures total current active requests.
- Request Throughput measures requests processed per second.
- Request Processing Time measures the average number of seconds required to process requests.

All monitoring is performed in real time using live data provided by the monitored component. Oracle HTTP Server is monitored not only for request activity but also for connections, errors, and data throughput. The percentage of CPU and memory usage is also monitored in real time.

Using Stand-Alone Oracle HTTP Server Based on Apache 2.0

The core enhancements in the Apache 2.0 version include:

- Support for Internet Protocol version 6 (IPv6)
- Improved performance on Windows
- Enhanced modules

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using Stand-Alone Oracle HTTP Server Based on Apache 2.0

Oracle HTTP Server is shipped with Oracle Application Server and is also available as a stand-alone component. There are two versions of Oracle HTTP Server available, based on Apache 1.3 and Apache 2.0. Oracle HTTP Server based on Apache 1.3 is available both in stand-alone and in customary integrated manner. Oracle HTTP Server based on Apache 2.0 is available only as a stand-alone component.

Use Oracle HTTP Server based on Apache 2.0, when you do not require:

- Java on the Web server for security reasons
- Clustering
- Application Server Control

Apache 2.0 is nearly a complete rewrite of Apache 1.3. Therefore, Apache 2.0 version has improved interfaces and performance, and reduced resource requirements. More significantly, the core enhancements include:

- Support for Internet Protocol version 6 (IPv6). On systems where IPv6 is supported by the underlying Apache Portable Runtime library, Apache gets IPv6 listening sockets by default. Additionally, the `Listen`, `NameVirtualHost`, and `VirtualHost` directives support IPv6 numeric address strings.
- Enhanced modules
- Improved performance on Windows

Oracle Application Server 10g R2: Administration I 6-36

Using Stand-Alone Oracle HTTP Server Based on Apache 2.0 (continued)

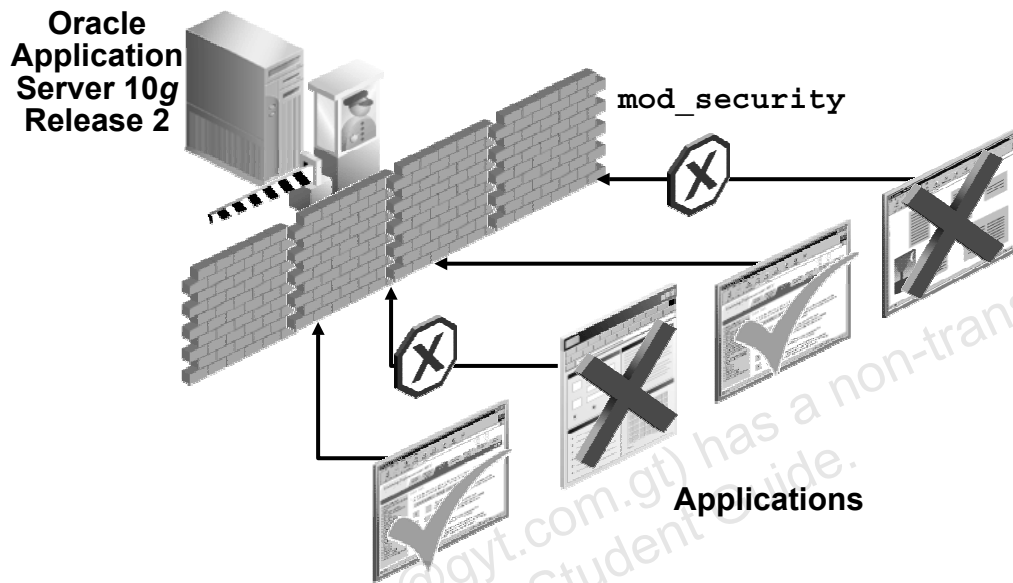
In addition, the stand-alone Oracle HTTP Server also supports:

- Log file rotation
- SSL session renegotiation
- nCipher SSL support
- Death detection and restarting of failed processes

Note: IPv6 is an updated version of IP version 4. Because of a growing shortage of IPv4 addresses (used by the Internet now), new addresses are needed by all new machines that are added to the Internet. IPv6 fixes a number of problems in IPv4, such as the limited number of available IPv4 addresses. It also adds many improvements to IPv4 in areas such as routing and network autoconfiguration. This version has the following renovations:

- Simpler header format
- Authentication and security extensions
- Increase in the size of IP addresses to 128 bits
- Simpler autoconfiguration of IP addresses
- Improved multicast routing due to the addition of a scope field to the multicast addresses

What Is mod_security?



ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is mod_security?

mod_security is an open source engine that detects intrusion and acts as a prevention engine for Web applications. Embedded into the Web server, it operates as an Apache Web server module. It shields applications from intrusion attempts, such as cross-site scripting and attacks, and thus increases the security of Web applications.

mod_security protects vulnerable applications by detecting and optionally rejecting HTTP requests that appear suspicious in the processing stream. In addition to filtering requests, it can also create Web application audit logs. You can thus use mod_security to protect your applications, instead of making changes to your applications.

mod_security, however, does not block all forms of attacks, such as those that attack SSL or HTTP. It deals with Web attacks in the following ways:

- Analyzes incoming requests when they come in and before they are handled by the Web server or other modules
- Normalizes paths and parameters before analysis, such as removing multiple slash characters, removing directory self-references, and decoding URL-encoded characters
- Performs very specific and fine-granulated filtering
- Intercepts the contents transmitted by using the POST method
- Logs details about every request for later analysis
- Accesses request data after decryption takes place

Oracle Application Server 10g R2: Administration I 6-38

Installing and Configuring mod_security

To install and configure mod_security, perform the following steps:

1. Log in as the user who installed Oracle HTTP Server.
2. Download the latest version of mod_security.
3. Extract the archive and change the directory to Apache home.
4. Compile the module.
5. Edit the httpd configuration file and save the changes.
6. Restart Oracle HTTP Server.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Installing and Configuring mod_security

To install mod_security, perform the following steps:

1. Log in as the user who installed Oracle HTTP Server.
2. Download the latest version of mod_security by using the following URL:
http://www.modsecurity.org/download/mod_security-1.8.6.tar.gz
3. Untar the archive and change the directory:

```
tar zxvf mod_security-1.7.4.tar.gz
```

```
cd mod_security-1.8.6
```

Change the directory to Apache home:

```
cd Apache
```
4. Compile the module:

```
/usr/local/apache/bin/apxs -cia mod_security.c
```
5. Edit the httpd.conf file. Search for the `<IfModule mod_security.c>` directive.

Installing and Configuring mod_security (continued)

Add the following:

```
<IfModule mod_security.c>
    # Turn the filtering engine On or Off
    SecFilterEngine On

    # Change Server: string
    SecServerSignature " "

    # Make sure that URL encoding is valid
    SecFilterCheckURLEncoding On

    # This setting should be set to On only if the Web
    # site is using the Unicode
    # encoding. Otherwise it may interfere with the
    # normal Web site operation.
    SecFilterCheckUnicodeEncoding Off

    # Only allow bytes from this range
    SecFilterForceByteRange 1 255

    # The audit engine works independently and
    # can be turned On or Off on the per-server or
    # on the per-directory basis. "On" will log
    # everything,
    # "DynamicOrRelevant" will log dynamic requests or
    # violations,
    # and "RelevantOnly" will only log policy
    # violations
    SecAuditEngine RelevantOnly

    # The name of the audit log file
    SecAuditLog /var/log/httpd/audit_log

    # Should mod_security inspect POST payloads
    SecFilterScanPOST On

    # Action to take by default
    SecFilterDefaultAction "deny,log,status:500"

    # Require HTTP_USER_AGENT and HTTP_HOST in all
    # requests
    SecFilterSelective "HTTP_USER_AGENT|HTTP_HOST"
    "^$"
```

Installing and Configuring mod_security (continued)

```
# Prevent path traversal (..) attacks
SecFilter "../"

# Weaker XSS protection but allows common HTML
#tags
SecFilter "<[[:space:]]*script"

# Prevent XSS attacks (HTML/Javascript injection)
SecFilter "<(.|n)+>"

# Very crude filters to prevent SQL injection
#attacks
SecFilter "delete[[:space:]]+from"
SecFilter "insert[[:space:]]+into"
SecFilter "select.+from"

# Protecting from XSS attacks through the PHP
#session cookie
SecFilterSelective ARG_PHPSESSID "!^[0-9a-z]*$"
SecFilterSelective COOKIE_PHPSESSID "!^[0-9a-z]*$"
</IfModule>
```

Save the file.

6. Restart Oracle HTTP Server.

Summary

In this lesson, you should have learned how to:

- Explain the Oracle HTTP Server processing model
- Describe the Oracle HTTP Server modules
- Configure and manage Oracle HTTP Server by using Oracle Application Server to:
 - Specify the server and file locations
 - Limit the number of processes and connections
 - Manage the network connections
 - Configure and use server log files
 - Edit server configuration files

ORACLE

Copyright © 2005, Oracle. All rights reserved.

7

Configuring Directives and Virtual Hosts

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

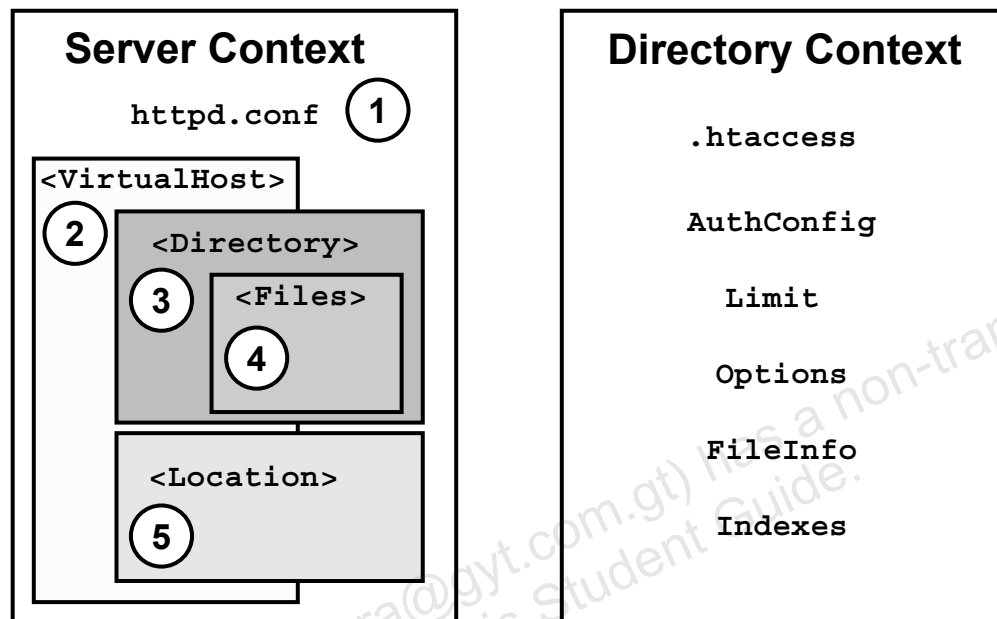
After completing this lesson, you should be able to do the following:

- **Describe configuration directives and their scopes**
- **Describe the process of merging containers and contents**
- **Configure directories and enable directory indexes**
- **Set up virtual hosts**
- **Use configuration directives, such as `Option`, `Alias`, and `ScriptAlias`**

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Configuration Contexts



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuration Contexts

The configuration tiers are applied hierarchically, which means that each directive overrides the directive in the preceding tier, enabling progressive refinement of the Oracle HTTP Server behavior in increasingly more specific areas.

The per-server context applies to the `httpd.conf` configuration file:

1. Directives outside any sections that are applied to the default or main server (depending on a particular directive) may be inherited by other sections.
2. Virtual host sections contain directives that are applied to a particular virtual server, and are distinguished by unique IP address–IP port pairs.
3. Directory sections contain directives that are applied to a particular directory (and its subdirectories), and are distinguished either by plain directory paths or by regular expressions matching directory paths.
4. File sections contain directives that are applied to particular files, and are distinguished by either plain file names or regular expressions matching file names.
5. URL sections contain directives that are applied to a particular URL and its subareas, and are distinguished by either plain relative URLs or regular expressions matching relative URLs.

Configuration Contexts (continued)

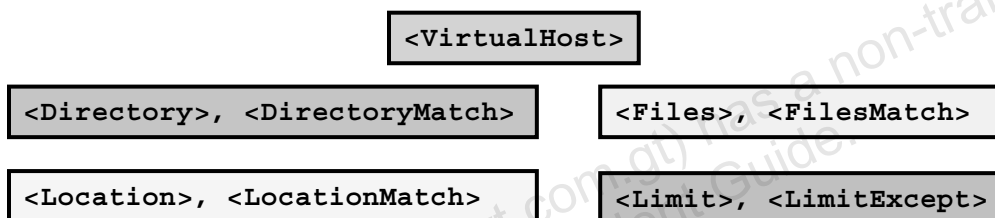
The directory context applies to the local `.htaccess` configuration files. They are read by Oracle HTTP Server as it discovers the file system structure. This context is also divided into five subcontexts that are enabled with the `AllowOverride` directive in the `httpd.conf` file. In the `.htaccess` file, there is no order of precedence and the directives are applied in the order they are defined:

- `AuthConfig` contains directives that control authorization (`mod_auth`).
- `Limit` contains directives that control access restrictions (`mod_access`).
- `Options` contains directives that control specific directory features (`http_core`).
- `FileInfo` contains directives that control document attributes (`mod_mime`).
- `Indexes` context contains directives that control directory indexing (`mod_index`).

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Container Directives

- **Container directives have opening and closing tags that surround other directives.**
- **Every directive within a container's tag affects only what that container refers to.**
- **Any directive that does not appear within a container applies to the entire server.**



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Container Directives

The container directives limit the scope of the directives they contain:

- **VirtualHost** enables one configuration to serve multiple sites from a single set of resources.
- **Directory** matches a physical location of the specified directory.
- **Location** matches a virtual path.
- **Files** matches specific file types.
- **DirectoryMatch**, **FilesMatch**, and **LocationMatch** enable the use of regular expressions in the matching string.
- **Limit** and **LimitExcept** restrict authentication and restrict specific HTTP methods that are seldom used.

Block Directives

The block directives limit the scope of application of other directives within them.

```
<IfModule mod_userdir.c>  
    UserDir public_html  
</IfModule>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Block Directives

The block directives limit the application of other directives within them to operate on particular virtual hosts, directories, or files. Container and block directives come in pairs.

The `<IfModule> ... </IfModule>` section is used to mark directives that are conditional on the presence of a specific module. In the example in the slide, the `IfModule` directive is a block directive that applies only to the `mod_userdir.c` module. `mod_userdir` is an Apache module that provides for user-specific directories. This module is contained in the `mod_userdir.c` file and is compiled by default.

Specifying the Location of the Directives

- **The directives can be specified within:**
 - **A server-level configuration section**
 - **A virtual host container**
 - **A Directory (including Location and Files) container**
 - **An .htaccess file**
- **The Limit or LimitExcept container may not include other containers, but may include any other directive.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specifying the Location of the Directives

Not all directives can be specified in each container. For example, the `ServerName` directive is allowed in the server configuration or a virtual host container, but nowhere else. Oracle HTTP Server does not start if you put it into a `Directory` container.

An exception is the `Limit` directive. The `<Limit>` directive restricts the effect of access controls to the nominated HTTP methods, and only in this case the directive takes effect.

So long as the `Limit` or `LimitExcept` directive is placed within a container that is acceptable to the directive, any directive is allowed inside a `Limit` or `LimitExcept` container.

Specifying the Location of the Directives (continued)

In terms of refining the scope of a container, note that no container directive can be nested within a directive of the same type. Therefore, you cannot nest a `Directory` or `Location` directive to refine the scope of successive definitions.

Instead, you can have the `Directory`, `Files`, and `Location` containers defined separately, but refer to the same areas of the file system.

The `Limit` directive and the `Files` container are notable exceptions to the rule that containers cannot enclose another container.

A `Files` container is allowed inside a `Directory` container, and, most importantly, the `VirtualHost` containers allow all of the other container types.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

<Directory> Directive

The <Directory> directive contains a group of directives that apply to the named directory and subdirectories:

```
<Directory />  
    Options none  
    AllowOverride none  
</Directory>  
  
<Directory /home/www/*>  
    AllowOverride all  
</Directory>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

<Directory> Directive

<Directory> encloses directives that apply only to the named directory and subdirectories of that directory. Any directive that is allowed in a directory context may be used. The <Directory> containers cannot be nested inside each other, but can refer to directories in the document root that are nested:

- <Directory /home/*/public_html> refers to the public_html subdirectory under any directory in /home.
- <Directory /> operates on the whole file system.
- <DirectoryMatch> should be used when specifying regular expressions, instead of using the tilde form of <Directory> with wildcards in the directory specification. The following two examples have the same result, matching directories that start with web and end with a number from 1 through 9:

```
<Directory ~/web[1-9] />  
<DirectoryMatch "/web[1-9] /">
```

<Files> and <Location>

- **<Files> matches files instead of directories:**

```
<Directory /ias20/public/images>
  <Files *.gif>
    SetHandler /cgi-bin/process-image.cgi
  </Files>
</Directory>
```

- **<Location> applies to a URL:**

```
<Location /server-info>
  SetHandler server-info
</Location>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Container Directives: <Files> and <Location>

- The <Files *file*> and </Files> directives support access control by a file name. They are comparable with the <Directory> and <Location> directives. The directives given within this section can be applied to any object with a base name (the last component of the file name) matching the specified file name. The <Files> sections are processed in the order that they appear in the configuration file, after the <Directory> sections and the .htaccess files are read, but before the <Location> sections. Note that the <Files> directives can be nested inside the <Directory> sections to restrict the portion of the file system to which they apply.
- The <FilesMatch> directive provides access control by file name, just as the <Files> directive does. However, it accepts a regular expression. For example, <FilesMatch "\.(gif|jpe?g|png)\$"> matches the most common Internet graphics formats: .gif, .jpg, .jpeg, and .png.
- The <Location> directive limits the application of the directives within a block to those URLs that are specified, rather than to a physical file location, such as the <Directory> directive. The <Location> sections are processed in the order that they appear in the configuration file, after the <Directory> sections and the .htaccess files are read, and after the <Files> sections.

Container Directives: <Files> and <Location> (continued)

<Location> accepts wildcard directories and regular expressions with the tilde character. The example in the slide shows the entry, if you want to have the server display server information whenever the URL is called.

- <LocationMatch> works in an identical manner to <Location> and you can use it for specifying regular expressions instead of the tilde form of Location with wildcards in the location specification. For example,
<LocationMatch "/(extra|special)/data">
matches URLs that contained the /extra/data or /special/data substring.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Defining Virtual Hosts

In a virtual host, additional hosts and Web sites can be defined alongside the main server:

- IP-based and name-based virtual hosts are defined with the `VirtualHost` container directive.
- The `VirtualHost` container includes a set of alternative directives to the main server, such as:

```
ServerAdmin  
ServerName  
DocumentRoot  
ErrorLog  
CustomLog  
Directory  
Location
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Defining Virtual Hosts

The term *virtual host* refers to the practice of maintaining more than one server on one machine, as differentiated by their apparent host names. Oracle HTTP Server supports both IP- and name-based virtual hosts with the `VirtualHost` container directive. Early versions of HTTP (including many other protocols, such as FTP) required a different IP address for each virtual host on the server. On some platforms, this can limit the number of virtual hosts that you can run. The `VirtualHost` directive is not only limited to IP addresses but also accepts host names. However, note that this puts an extra burden on the server because it requires name server lookups and makes Oracle HTTP Server vulnerable to domain name server (DNS) faking attacks.

`<VirtualHost>` and `</VirtualHost>` are used to enclose a group of directives that apply only to a particular virtual host. Any directive that is allowed in a virtual host context can be used. When the server receives a request for a document on a particular virtual host, it uses the configuration directives that are enclosed in the `<VirtualHost>` section. These directives consist primarily of `ServerAdmin`, `ServerName`, `DocumentRoot`, `ErrorLog`, and `CustomLog`.

Defining Virtual Hosts (continued)

Note that if you do not define one of these directives (with the exception of `ServerName`) in your virtual host, then the setting is inherited from the main server-level configuration. Note that `ServerName` in a virtual host has nothing to do with the name to which the virtual host responds. Moreover, the name to which the virtual host responds is defined by the IP address or host name in the `VirtualHost` directive itself. Rather, `ServerName` gives the name of the host when creating redirection URLs; without it, Oracle HTTP Server is forced to perform a DNS lookup of the virtual host's IP address to discover the name.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Using IP-Based Virtual Hosts

A virtual host can be IP based:

```
<VirtualHost 130.35.174.159 205.134.38.199>  
    ServerName www.oracle.com  
    ServerAdmin Webmaster@oracle.com  
    DocumentRoot /oras/oracle/www  
    ErrorLog /oras/oracle/logs/error_log  
    TransferLog /oras/oracle/logs/access_log  
</VirtualHost>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using IP-Based Virtual Hosts

You can configure more than one name or IP address on each host. This is one way to ensure that a large machine consists of many smaller machines, with one machine responding to the same name on different interfaces. For example, you want to have a `VirtualHost` directive that is available to hosts on an internal (intranet) as well as external (Internet) network. This approach is also used in high-availability failover situations.

The `VirtualHost` directive in the configuration file is used to set the values of the `ServerAdmin`, `ServerName`, `DocumentRoot`, `ErrorLog`, and `TransferLog` or `CustomLog` configuration directives to different values for each virtual host. It is recommended that you use an IP address instead of a host name (see domain name server [DNS] look-up caveats described earlier).

Almost any configuration directive can be put in the `VirtualHost` directive, except for those directives that are allowed to be used only in the server configuration context, such as the following: `ServerType`, `StartServers`, `MaxSpareServers`, `MinSpareServers`, `MaxRequestsPerChild`, `BindAddress`, `Listen`, `PidFile`, `TypesConfig`, `ServerRoot`, and `NameVirtualHost`.

Using Name-Based Virtual Hosts

A virtual host can be name based:

```
<NameVirtualHost 205.134.38.199
VirtualHost www.host1.com>
    DocumentRoot /usr/virtual/htdocs/customers
    ServerName www.host1.com
    ErrorLog /usr/virtual/h1/logs/error_log
</VirtualHost>
<VirtualHost www.host2.com>
    DocumentRoot /usr/virtual/htdocs/internal
    ServerName www.host2.com
    ErrorLog /usr/virtual/h2/logs/error_log
</VirtualHost>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using Name-Based Virtual Hosts

The notable difference between IP- and name-based virtual host configurations is the `NameVirtualHost` directive that specifies an IP address that should be used as a target for name-based virtual hosts.

Note: When you specify an IP address in a `NameVirtualHost` directive, requests to that IP address are served only by matching the `<VirtualHost>` directives. The main server is never served from the specified IP address. If you start to use virtual hosts, you should stop using the main server as an independent server and use it as a place for configuration directives that are common for all your virtual hosts. In other words, you should add a `<VirtualHost>` section for every server (host name) that you want to maintain on your server.

For more information about virtual hosts, see the Web site at <http://www.apache.org/docs/vhosts/>.

Configuring Virtual Hosts

HTTP_Server						
Home Virtual Hosts Administration						
<div style="text-align: right;">Page Refreshed</div> <div> Create </div> <div> Create Like Delete </div>						
Select	Server Name	Port	IP Address	Type	Protocol	Average Response Time (seconds)
<input checked="" type="radio"/>	127.0.0.1	7201	127.0.0.1	IP-based	http	0.005
<input checked="" type="radio"/>	EDRSR16P1	4444		default	https (SSL)	Unavailable
Home Virtual Hosts Administration						

Welcome General **Addresses** Ports Protocol Error Log More

Create Virtual Host: Addresses

[Cancel](#) [Back](#) Step 3 of 7 [Next](#)

Enter the server name, server aliases, and IP address to be used with this name-based virtual host.

Server Name and Aliases

* Server Name

Select row and... [Remove](#)

Select All | Select None

Select Server Alias

☐ ☐

[Add Another Row](#)

TIP Values entered for Server Name and Server Alias should be valid DNS names. If you set name1.mydomain.com as the Server Name, some typical Server Aliases include www.name1.mydomain.com and name1.

IP Address

☒ Listen on all the main server IP addresses

☐ Listen on a specific IP address

[Cancel](#) [Back](#) Step 3 of 7 [Next](#)

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Virtual Hosts

To create an IP- or name-based virtual host container by using Application Server Control, perform the following steps:

1. Navigate to the Oracle HTTP Server home page. Scroll down to the Virtual Host region.
2. Click the Create button. This opens the Virtual Host Wizard Welcome page. Click the Next button to proceed to the next window.
3. In the General window, you can specify the path for the document root directory, the e-mail address for ServerAdmin, and the virtual host container type that you want to create. Enter information and click the Next button.
4. Now you define the server name and server aliases, as well as the IP-address that your virtual host listens on. Click the Next button.
5. On the next page, you can select the port setting that should be applied to this virtual host. To proceed, click the Next button.

Note: You must select the protocol (HTTP or SSL) that you want to use for transferring data to and from the virtual host. However, the SSL protocol is not supported for name-based virtual hosts.

Configuring Virtual Hosts (continued)

6. In this step, you can specify the path to the error log and select the logging level. To access the Summary page, click the Next button.
To apply the configuration, review your settings and click the Finish button.
Application Server Control displays a confirmation page, which confirms that the appropriate configuration file has been updated.
7. To restart Oracle HTTP Server and for the changes to take effect, click Yes.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Controlling Allowed Features

- **Use Options to enable and disable features:**

```
Options ExecCGI FollowSymLinks
```

- **Use AllowOverride to control overrides:**

```
AllowOverride FileInfo Indexes
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Controlling Allowed Features

Options and AllowOverride are the key directives to control what features Oracle HTTP Server allows, and how many control files outside the server configuration are allowed:

- The Options directive gives far-reaching control over what users get and the way Apache regards the file system, with each parameter controlling a different aspect of Apache's handling of files. For example, with the ExecCGI option, the server recognizes files as CGI scripts in this container, whereas FollowSymLinks allows your server to follow symbolic links.
- The AllowOverride directive tells your Oracle HTTP Server which directives in a per-directory .htaccess file can override the server configuration, including the Options directive. This mechanism enables you to exert finer control over what is done in the .htaccess files.

Options Parameters

- **All**
- **ExecCGI**
- **FollowSymLinks**
- **SymLinksIfOwnerMatch**
- **Includes**
- **IncludesNOEXEC**
- **Indexes**
- **MultiViews**
- **None**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Options Parameters

The following is a complete list of Options parameters:

- **All:** Enables all options by default, except for the MultiViews directive and those that are mutually exclusive, such as FollowSymLinks, SymLinksIfOwnerMatch, Includes, and IncludesNoExec. Typically, a Web master changes this parameter to something more desirable.
- **ExecCGI:** Permits the execution of CGI scripts (or prohibits it if the parameter is not set). Except for those directories defined with ScriptAlias, ExecCGI must be defined for any executable content to function. ExecCGI is discussed later in more detail.
- **FollowSymLinks:** Enables Apache to follow symbolic links defined on the file system for files or directories. You need to have Read privileges for successful reference. This option has no effect inside Location containers.
- **SymLinksIfOwnerMatch:** Operates in an identical manner as FollowSymLinks, but has higher priority. The option specifies that the server follows only those symbolic links for which the target file or directory is owned by the same user ID as the link.
- **Includes:** Controls the execution of server-side includes (SSI)
- **IncludesNOEXEC:** Permits server-side includes, but disallows the execution of CGI scripts through the #exec and #include commands

Options Parameters (continued)

- **Indexes:** Specifies that if a URL is requested that maps to a directory, and there is no corresponding index file identified with the `DirectoryIndex` directive, a formatted listing of the directory contents is created and returned
- **MultiViews:** Supports content-negotiated multiple views. This directive is not enabled with `All`. For more details about content negotiation, refer to the Apache documentation.

The options can be preceded by `+` or `-`, in which case they are added or removed. If there are multiple `Options` directives for the same directory, then they are merged together. Having specified `Options` directives for the same directory in a `Directory` container in the server configuration and an `.htaccess` file in the directory, has the same result: the two directives are merged.

If no options are set, and there is no `<Limit>` directive, then the effect is as if `All` has been set. Alternatively, a directory inherits options from the directories above it if it does not have an explicit `Options` directive set. Those inherited directives can be modified rather than simply overridden by using the `+` and `-` modifier prefixes; an `.htaccess` file in a subdirectory can modify the settings as well.

Because Apache looks for inherited directives all the way up the directory tree, in terms of both scope and `.htaccess` files, a succession of modifiers can be applied.

If you want to clear all inherited and incremental settings, specify an option without a prefix. This ensures that only the `Options` directives in either a `Directory` container or an `.htaccess` file for that directory are changed.

Using Options

Example A:

```
# Using Absolute Options
<Directory /web/docs>
    Options Indexes FollowSymLinks
</Directory>
<Directory /web/docs/spec>
    Options Includes
</Directory>
```

Example B:

```
# Using Relative Options
<Directory /web/docs>
    Options Indexes FollowSymLinks
</Directory>
<Directory /web/docs/spec>
    Options +Includes -Indexes
</Directory>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using Options

Normally, if multiple `Options` directives could apply to a directory, then the most specific one is used; the options are not merged. However, if all the options in the `Options` directive are preceded by a `+` or `-` symbol, then the options are merged. Any options that are preceded by the `+` symbol are added to the options that are currently being used, and any options preceded by the `-` symbol are removed from the options that currently being used.

For example, the configuration in example A sets only the `Includes` options for the `/web/docs/spec` directory.

However, in example B, the `Options` directive uses the `+` and `-` symbols:

```
<Directory /web/docs>
    Options Indexes FollowSymLinks
</Directory>
<Directory /web/docs/spec>
    Options +Includes -Indexes
</Directory>
```

In this case, the `FollowSymLinks` and `Includes` options are also set for the `/web/docs/spec` directory.

Overriding Directives with the Per-Directory Configuration

- **Oracle HTTP Server allows the server configuration to be supplemented with the following per-directory configuration files:**
 - `.htaccess` file
 - `AllowOverride`
`All, AuthConfig, Limit, FileInfo, Indexes, Options, None`
- **Using directives outside of the standard configuration files may cause the configuration repository to be out of sync.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Controlling Overrides with `AllowOverride`

Oracle HTTP Server allows the server configuration to be supplemented with per-directory configuration files. By default, if there is a readable file called `.htaccess` in a directory inside your path, Oracle HTTP Server treats the directives in them as if they were in a `Directory` container for that directory. (The name is not necessarily `.htaccess`; you can change it with `AccessFileName`.)

The `AllowOverride` directive is valid only in `Directory` containers. This directive specifies whether the directives that are retrieved from `.htaccess` are to be considered or not. The default setting is `All`, which means all the `.htaccess` files are read and parsed. From each `.htaccess` file for the same directory level, the server merges the directives found in any `Directory` containers (excluding `Directory` containers using regular expressions). The order of merging directives together is that the directives in lower `Directory` containers and `.htaccess` files have precedence over those in higher ones. After checking all directories and merging the directives in them, the server processes other containers whose scope covers the URL, as discussed earlier in this lesson.

Controlling Overrides with AllowOverride (continued)

Only certain directives can be defined in a per-directory configuration file. With the `AllowOverride` directive, parts of the subset can be enabled or disabled. The following is a complete list of override options:

- **All:** Enables all overrides. This default setting causes security risks.
- **AuthConfig:** Allows the use of directives belonging to user authentication, such as `AuthName`, `AuthType`, `AuthUserFile`, and `require`
- **FileInfo:** Can be specified to control file types such as `AddType`, `DefaultType`, and others
- **Indexes:** Permits directives controlling directories. It is not the same as the `Indexes` option, because it enables or disables the overriding, not the appearance, of directory indexes. For example, to allow directory indexes but prevent configuration in the `.htaccess` files, enable the option but disable the override.
- **Limit:** Allows the use of `allow`, `deny`, and `order` to control host access
- **Options:** Enables the use of `Options`. It is a good approach to disable the overriding of `Options` to prevent the `.htaccess` files from enabling the use of CGI scripts and SSIs in places where the server configuration denies them.
- **None:** Ensures that the `.htaccess` files are ignored. To improve both security and, particularly, performance, you should consider using `None` for most Oracle HTTP servers.

The process of inheritance for `AllowOverride` follows the same rules as for `Options`, and allows the inherited overrides to be modified with `+` and `-` in the same manner.

Directory Indexing

- **Enable or disable directory indexing:**

```
Options +Indexes
```

- **Use DirectoryIndex to change the default file displayed:**

```
DirectoryIndex index.html index.htm
```

- **Specify a nonrelative URL as a last resource to prevent the generation of an index for the directory:**

```
DirectoryIndex index.html /cgi-bin/error404.cgi
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Directory Indexing

Oracle HTTP Server can do one of the following when asked for a directory rather than a file:

- Return a default file in the directory
- Generate an HTML page of the contents of the directory
- Return an error stating that the file is not found
- Return an error stating that permission is denied

The `Indexes` option specifies whether to generate an HTML page with a directory listing. For example, `Options +Indexes` adds indexing to the list of active options. It is generally a good idea to disable indexing unless you need it, because it can be used by unwanted visitors to discover things about the Web site and the files in it, making other security weaknesses (such as backups of CGI scripts) easy to find.

In the example in the slide, the `DirectoryIndex` directive instructs Oracle HTTP Server to append `index.html` to the end of any URL that resolves a directory and return the resource of that name if it finds it. Using a nonrelative URL as the last option of `DirectoryIndex` prevents the creation of an index by ensuring that at least one resource in the `DirectoryIndex` directive is found.

DirectoryIndex Directive

A nonrelative URL specified as a last resource prevents the generation of an index of the directory:

```
DirectoryIndex index.html /cgi-bin/error404.cgi
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

DirectoryIndex Directive

In the example in the slide, the `DirectoryIndex` directive tells Oracle HTTP Server to append `index.html` to the end of any URL that resolves to a directory and return the resource of that name if it finds it. An index of the directory is generated instead, if none of the resources specified by `DirectoryIndex` are found, and if indexes are enabled. To prevent this, specify a nonrelative URL as the last option of `DirectoryIndex`.

In the example, the `error404.cgi` script is run for any requested directory that contains neither `index.html` nor `index.htm`.

Controlling Directory Listings with IndexIgnore

You can prevent files from appearing in the directory listing by using the IndexIgnore directive:

```
IndexIgnore .??* *~ *# *.bak HEADER* README*
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Controlling Directory Listings with IndexIgnore

Often, you do not want users to see all files in a directory (for example HEADER and README files). Less obvious examples are backup files, subdirectories containing file revision archives, and dot files (such as .htaccess, .cshrc, and .profile).

The IndexIgnore directive enables you to prevent files from appearing in file listings. This directive is followed by a list of files or wildcards describing the files to be ignored.

The directive explained in the slide ignores anything that looks like a backup file, a header, or a read-me file, and any file whose name starts with a dot and is three or more characters long. It still allows . . , so that clients can navigate to the enclosing directory.

Because the current directory is suppressed automatically in directory listings, and it is not possible to get mod_autoindex to display it, you do not need to specify it.

Oracle HTTP Server merges together multiple IndexIgnore directives, both those in the same directory and those inherited from higher-level directories.

Note: If an IndexIgnore directive is specified in the server configuration, it cannot be overridden by an .htaccess file. A file that has been suppressed by IndexIgnore once is always ignored. There is no way to reinstate such a file.

Error and Response Handling

- **Error and response codes:**

Category	Meaning
100+	Informational
200+	Client request successful
300+	Client request redirected, further action necessary
400+	Client request incomplete
500+	Server errors

- **The ErrorDocument directive:**

```
ErrorDocument 404 "Sorry, document not found"  
ErrorDocument 404 /errors/notfound.html  
ErrorDocument 500 /errors/fake404.cgi
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Error and Response Handling

This section does not cover error handling in general, but describes how to control Oracle HTTP Server behavior to determine what clients see when an error occurs. Before seeing how to handle errors, note the kinds of responses Apache can generate. Errors are actually just one kind of response code defined by HTTP. When Oracle HTTP Server encounters a problem processing a client request, it logs the error in the `error_log` file and returns an error response to the client. By default, Apache generates a short HTML document containing the error code and the reason for it. You may prefer to have Apache respond in a way you choose, or possibly pretend that nothing has happened. You can customize the response of the server to errors by using the `ErrorDocument` directive.

In the event of a problem or error, you can use `ErrorDocument` to perform one of the following:

- Output a simple hard-coded error message, as in the first example.
- Output a customized message, as in the second example.
- Run a CGI script, as in the last example.

Error and Response Handling (continued)

`ErrorDocument` takes two parameters: the error code to handle and the action to take, which can be either a customized error message or a URL. Note that the first example in the slide does not include a determining double quotation mark; if you put one, it appears in the message.

The last example shows how to deal with a 500 - Internal Server Error error, which is always an embarrassing thing to appear on a user's screen. You can use a CGI script to turn this error into a Not Found Error message, which is identical to the message that Oracle HTTP Server ordinarily generates.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Expires Header

The **Expires** headers are used to control the caching behavior for Web content.

- To enable the sending of the **Expires** headers:

```
ExpiresActive on
```

- To set a default expiration time:

```
ExpiresDefault A2419200
ExpiresDefault M86400
ExpiresDefault "access plus 1 month"
```

- To set expiration times by media type:

```
ExpiresByType image/gif A2419200
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Expires Header

The `mod_expires` module controls the **Expires** header. The proxies and client-browser caches consider a document being current based on the **Expires** header of the document. The value of the **Expires** header is a date beyond which the document is considered out-of-date. With the `ExpiresActive` directive, you switch on or off sending of the **Expires** header.

The `ExpiresDefault` directive defines a default expiry time for all files on the server. For files that change very rarely, such as archived documents, a setting as shown in the example in the slide may be useful because Oracle HTTP Server would send an **Expires** header so that documents expire 2,419,200 seconds (28 days) after the file is accessed by the client. The second example tells Oracle HTTP Server to send an **Expires** header so that documents expire 86,400 seconds (one day) after the date they are last modified, which is useful for pages that are updated daily. However, there are two important caveats:

- If the modification date of the page is ever more than one day in the past, then the page is never cached because the document is deemed to have already expired.
- An **Expires** header is not set if the source of the document is not a file on disk, because there is no modification time on which to base it.

Expires Header (continued)

`ExpiresByType` enables you to differentiate expiry criteria, with one expiry time for HTML documents, another for GIF images, and so on. Both `ExpiresDefault` and `ExpiresByType` understand an alternative verbose format for expiry times that are more human readable. For more information about `mod_expires`, see the Web site at http://httpd.apache.org/docs/mod/mod_expires.html.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Alias, AliasMatch, and ScriptAlias

Aliases allow accesses to resources from a location other than the DocumentRoot directory:

- **Use Alias to store documents elsewhere:**

```
Alias /soapdocs/ /ias/soap/
```

- **AliasMatch allows you to use regular expressions:**

```
AliasMatch /images/(.*)\.gif$ /ias/images/$1.gif
```

- **Use ScriptAlias to store scripts elsewhere and mark them as CGI scripts:**

```
ScriptAlias /cgi-bin/ /ias/Apache/Apache/cgi-bin/
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Alias, AliasMatch, and ScriptAlias

The `mod_alias` module provides for mapping different parts of the host file system in the document tree, and for URL redirection. The `Alias` and `ScriptAlias` directives are used to map between URLs and file system paths.

Aliases allow documents to be stored somewhere in the file system other than in the `DocumentRoot` directory. The URL is translated into a different location on the disk without the client being aware of it. The `Alias` example used in the slide substitutes the part of the URL that starts with `/soapdocs/` with `/ias/soap/` before retrieving the requested file. Some limitations of the `Alias` directive are that it cannot, for example, alias part of a URL that does not start at `DocumentRoot`, nor can it alias URLs based on the file extension.

The `AliasMatch` directive enables you to replace the URL prefix of `Alias` with a regular expression. The example ensures that any reference to an `images` directory in a URL is redirected to the desired real `images` directory.

The `ScriptAlias` directive controls which directories contain server scripts. `ScriptAlias` is essentially the same as `Alias`, except that documents in this directory are treated as applications and are run by the server when requested, rather than treated as documents and sent to the client. `ScriptAlias` is also the only means of enabling CGI script execution without specifying the `ExecCGI` option. It is, therefore, a popular choice for servers with user accounts and a policy of not allowing user-written CGI scripts.

Summary

In this lesson, you should have learned how to:

- Use configuration directives and their scopes
- Configure directories and enable directory indexes
- Set up virtual hosts
- Use configuration directives, such as `Option`, `Alias`, and `ScriptAlias`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

8

Configuring and Managing OracleAS Web Cache

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

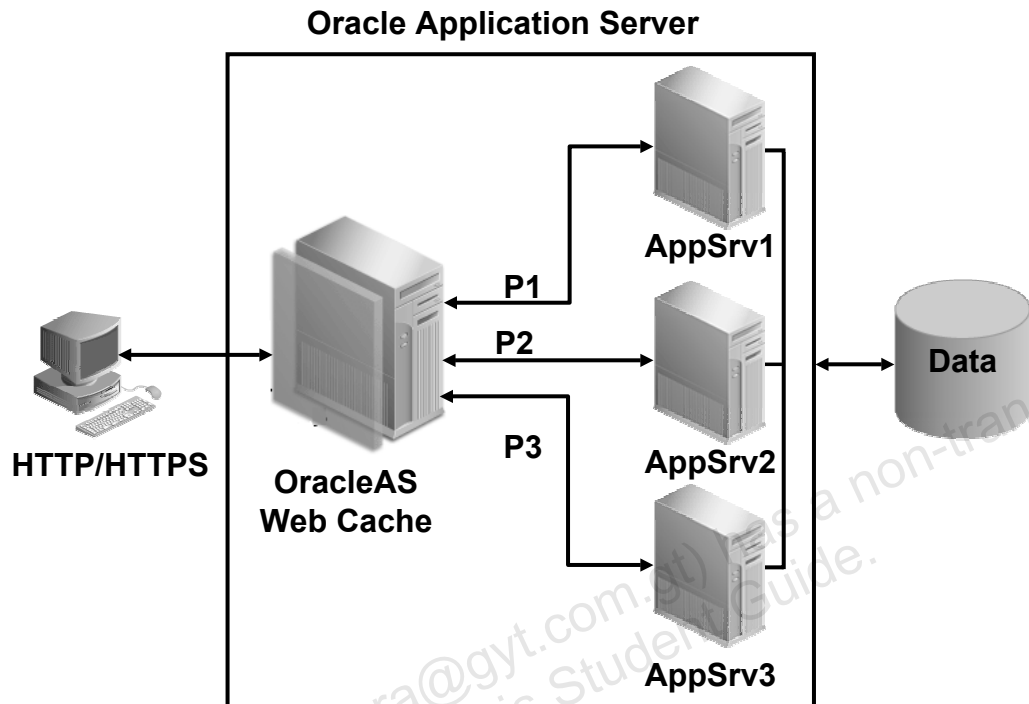
After completing this lesson, you should be able to do the following:

- **Start, stop, and restart OracleAS Web Cache**
- **Change passwords for administrative users and listener ports**
- **Specify site-to-server mappings**
- **Create and configure caching rules**
- **Set up basic invalidation mechanism**
- **Set up expiration rules**
- **Configure access and event logs**
- **Obtain basic performance statistics**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is OracleAS Web Cache?



ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is OracleAS Web Cache?

OracleAS Web Cache functions as an external cache and load balancer, and includes the following features:

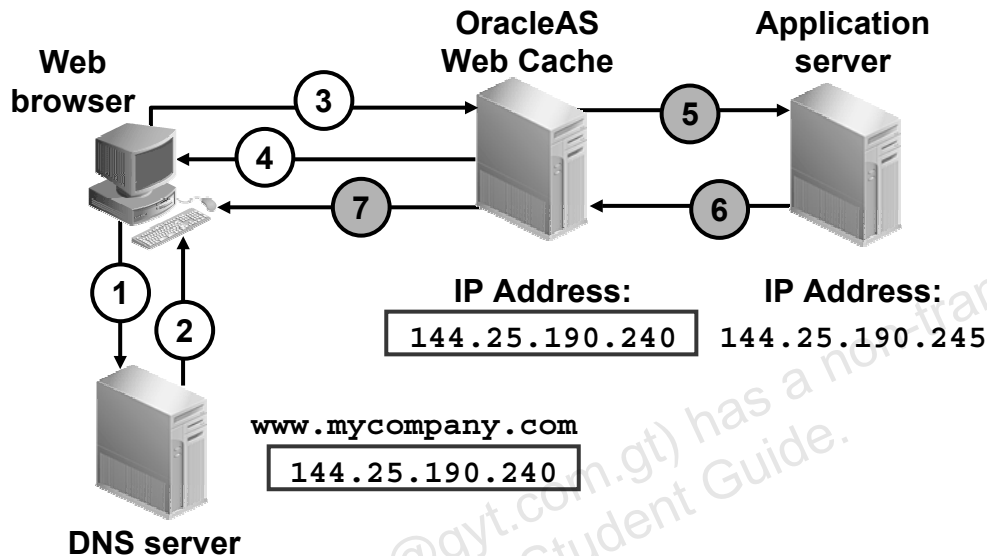
- Static content caching (HTML files and image files such as GIF or JPEG files)
- Dynamically created read-only content caching (for example, a JSP-driven product catalog)
- Consistency management with HTTP-based invalidation messages and declarative expiration
- Content-aware caching and routing based on HTTP header information, including cookies
- Self-tuning logic that protects Web servers during traffic spikes
- Web server load balancing and failover
- Compression of large files for faster delivery
- Ability to run on the same or a separate machine from Web servers
- Clustering support
- Support of partial page caching using Edge Side Includes (ESI) technology

What Is OracleAS Web Cache? (Continued)

The previous slide shows the basic architecture of OracleAS Web Cache. When Web browsers access a Web server, they send HTTP or HTTPS protocol requests to OracleAS Web Cache. OracleAS Web Cache, in turn, acts as a virtual server to the application Web servers. This Web cache stores the content. If the requested content is unavailable, the Web cache forwards the request to Oracle Application Server. If the requested content has been expired or invalidated, OracleAS Web Cache retrieves the new content from the application Web servers.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

How Does OracleAS Web Cache Work?



Copyright © 2005, Oracle. All rights reserved.

How Does OracleAS Web Cache Work?

1. A browser sends a request to a Web site with the URL www.mycompany.com. This request, in turn, generates a request to the domain name system (DNS) for the IP address of the Web site.
2. If an entry is found, DNS returns the IP address; in this case, it is 144.25.190.240.
3. The browser sends the request for the Web page, 144.25.190.240, to OracleAS Web Cache. (This needs to be configured, because the default port is 7777 or whatever that is assigned during installation.)
4. If the requested content is in its cache, then OracleAS Web Cache sends the content directly to the browser. This is called a cache hit.
5. If OracleAS Web Cache does not have the requested content or if the content is stale or invalid, then it passes the request to the application Web server. Each request that cannot be satisfied by OracleAS Web Cache is called a cache miss.
6. The application Web server sends the content to OracleAS Web Cache.
7. OracleAS Web Cache sends the content to the client and makes a copy of the page in the cache. A page stored in the cache is removed when it becomes invalid or outdated, as described later in this lesson.

OracleAS Web Cache Concepts

- **Populating OracleAS Web Cache**
- **Cache freshness and performance assurance:**
 - Expiration (rule based)
 - Invalidation (event based)
- **Cache hit and cache miss responses**
- **Caching dynamically generated content**
- **Edge Side Includes (ESI):**
 - Partial page caching
 - Content assembly
- **Edge Side Includes for Java (JESI)**
- **Content assembly and partial page caching**
- **Cache clustering**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Web Cache Concepts

The main concepts of OracleAS Web Cache involve the following:

Populating OracleAS Web Cache

When OracleAS Web Cache is first configured with a caching rule for a set of documents, those documents are not placed in the cache until there is a browser request for them. When a request comes in, OracleAS Web Cache sends the request to the application Web server. If the requested document is specified as one of the documents to cache, OracleAS Web Cache caches the document for subsequent requests.

Cache Freshness and Performance Assurance

The cached objects lose their relevance either over time or because of changes at the origin server. Invalidation and expiration policies ensure consistency between the cache and the content on the application Web servers:

- The invalidation mechanism is initiated by the user or administrator. You can use the invalidation mechanism to mark cache documents as invalid as and when you determine that the cached documents need to be refreshed.
- The expiration mechanism enables automation of the cache freshness. Documents can be marked as invalid after a certain amount of time in the cache.

OracleAS Web Cache Concepts (continued)

Cache Hit and Cache Miss Responses

For each requested document from the cache, OracleAS Web Cache adds cache hit or cache miss information to the Server response-header field of the HTTP response message.

Caching Dynamically Generated Content

For dynamically generated pages, browsers pass information about themselves to the origin server, enabling the origin server to serve appropriate content to the browser. HTTP has a way for browsers and origin servers to share information, such as session or category information, in message headers that browsers pass with every request to the origin server. One approach is to use cookies. The other method is to use embedded URL parameter. OracleAS Web Cache is able to recognize both cookies and embedded URL parameters and apply caching rules for dynamically generated pages.

Edge Side Includes (ESI)

ESI is an XML-like markup language that enables dynamic content assembly of fragments by OracleAS Web Cache.

A template page is configured with ESI markup tags that fetch and include dynamic HTML fragments. The fragments themselves can also contain ESI markup.

You can assign caching rules to the template page and HTML fragments. By enabling page assembly in OracleAS Web Cache, rather than in the application Web server, you can increase cache hit rates and improve performance.

ESI for Java (JESI)

OC4J provides the JESI tag library as an interface to ESI tags and functionality. JESI facilitates the programming of JavaServer Pages (JSPs) by using ESI. You have the option of using ESI tags directly in any Web application, but JESI tags provide additional functionality in a JSP environment. ESI and JESI are open standards, and you can use the JESI tag library in any standard JSP environment as long as an ESI processor, such as OracleAS Web Cache, is available.

Content Assembly and Partial Page Caching

OracleAS Web Cache provides dynamic assembly of Web pages with both cacheable and noncacheable page fragments. OracleAS Web Cache enables Web pages to be broken down into fragments of differing caching profiles. These fragments are each maintained as separate elements in OracleAS Web Cache. The fragments are assembled into HTML pages as appropriate when requested by end users.

Cache Clustering

A collection of OracleAS Web Cache instances working together to provide a single logical cache is OracleAS Cluster (Web Cache). Cache clusters provide failover detection and failover of caches, thus increasing the availability of your Web site.

Administering OracleAS Web Cache

- **Setting up caching rules**
- **Starting and stopping OracleAS Web Cache**
- **Invalidating documents in the cache**
- **Evaluating event logs**
- **Evaluating access logs**
- **Monitoring OracleAS Web Cache statistics**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Administering OracleAS Web Cache

Administrative tasks, besides starting, stopping, and restarting OracleAS Web Cache, include setting up the caching rules, invalidating the whole cache, and evaluating event and access logs:

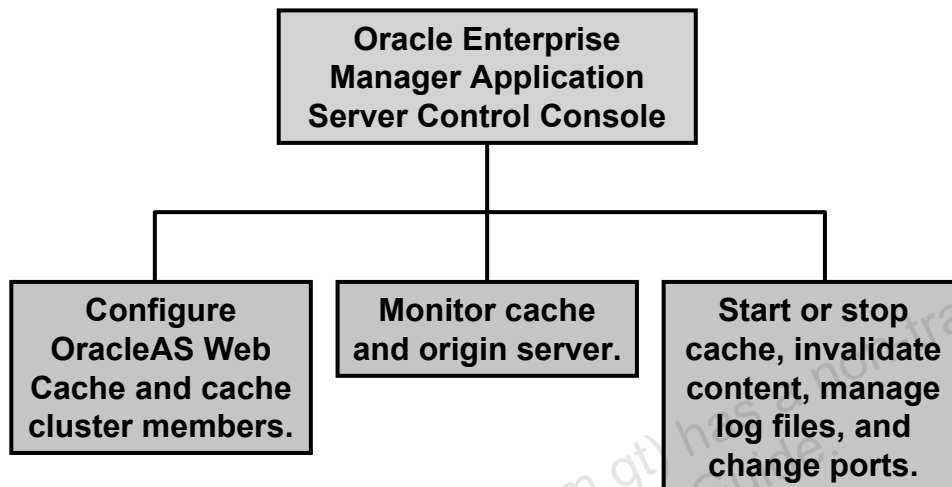
- The caching rules enable you to define various URL expressions. These URL expressions can represent one or more documents. If the expression represents several documents, such as complete directories and subdirectories, it is called a subtree of URLs.
- Invalidating documents in the cache: Invalidation messages are sent to an OracleAS Web Cache invalidation listening port through HTTP POST messages. The invalidation messages identify the documents to be invalidated. For a Web administrator, it may be necessary to:
 - Change the invalidation port number
 - Initiate sending invalidation messages
- OracleAS Web Cache events and errors are stored in an event log. The event log can help you determine which documents or objects have been inserted into the cache. It can also identify listening port conflicts or startup and shutdown issues. A Web administrator should monitor the event log on a regular basis.

Administering OracleAS Web Cache (Continued)

- Each Web site that OracleAS Web Cache supports has its own access log. An access log contains information about the HTTP requests sent to OracleAS Web Cache for a Web site. According to this, it is also useful for a Web administrator to monitor the access log regularly.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Managing OracleAS Web Cache with Application Server Control Console



ORACLE

Copyright © 2005, Oracle. All rights reserved.

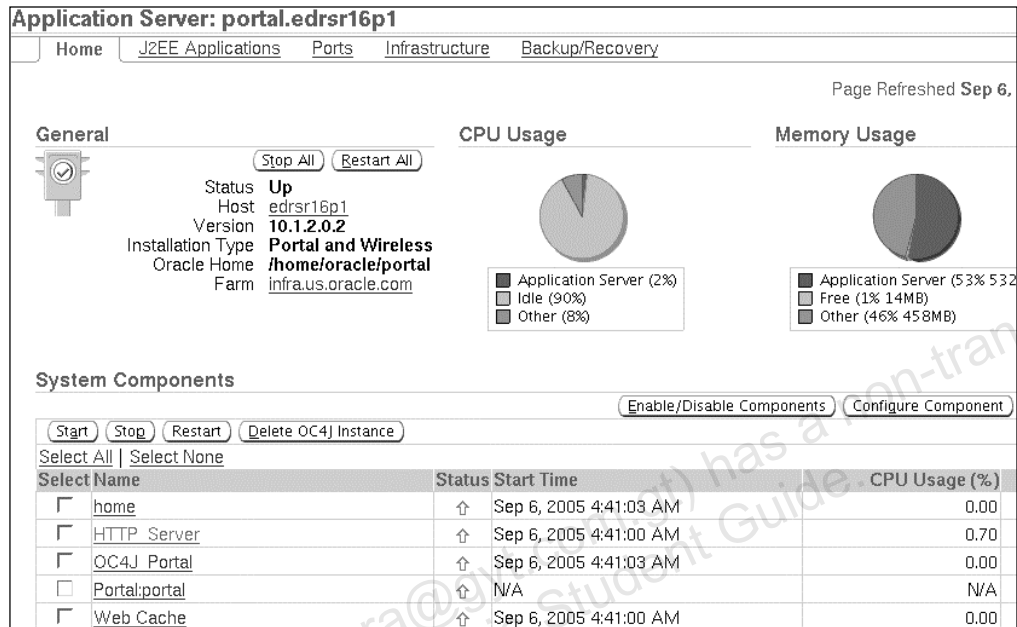
Managing OracleAS Web Cache with Application Server Control Console

In Oracle Application Server 10g Release 2, you can fully manage and administer OracleAS Web Cache instances by using the Application Server Control Console. There is no need to launch and log in to a separate tool (OracleAS Web Cache Manager) to administer OracleAS Web Cache. You can perform all administration tasks from the Application Server Control Console.

From the Application Server Control Console, you can configure OracleAS Web Cache, monitor cache and origin server status, and perform operational tasks (such as starting and stopping the cache, invalidating content, propagating configuration among cache cluster members, and managing log files).

Note: You can also install OracleAS Web Cache in a stand-alone environment, without it being part of an Oracle Application Server installation.

Using Application Server Control to Start and Stop OracleAS Web Cache



Using Application Server Control to Start and Stop OracleAS Web Cache

To start, restart, or stop OracleAS Web Cache, you can use the Application Server Control Console. Select Web Cache in the System Components table, and click the appropriate button to start, restart, or stop your Web Cache.

It is also possible to start, restart, or stop OracleAS Web Cache from the Web Cache home page of the Application Server Control Console.

Using `opmnctl` to Start and Stop OracleAS Web Cache

- You can use the `opmnctl` utility to start, stop, and restart OracleAS Web Cache processes:

```
$ opmnctl startproc ias-component=WebCache
```

```
$ opmnctl stopproc ias-component=WebCache
```

```
$ opmnctl restartproc ias-component=WebCache
```

- In a stand-alone OracleAS Web Cache installation, you can use `webcachectl` to start and stop OracleAS Web Cache:

```
$ webcachectl start
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using `opmnctl` to Start and Stop OracleAS Web Cache

On UNIX/Linux platforms, OracleAS Web Cache manifests two processes: the Web Cache admin process and the Web Cache server process. On Windows, they appear as two threads within a single process.

You can start, stop, and restart OracleAS Web Cache processes by using `opmnctl`. To determine the status of OracleAS Web Cache, use the `opmnctl status` command. For example:

```
$ opmnctl status
ias-component | process-type | pid | status
-----+-----+-----+-----
WebCache      | WebCache      | 24286 | Alive
WebCache      | WebCacheAdmin | 24278 | Alive
```

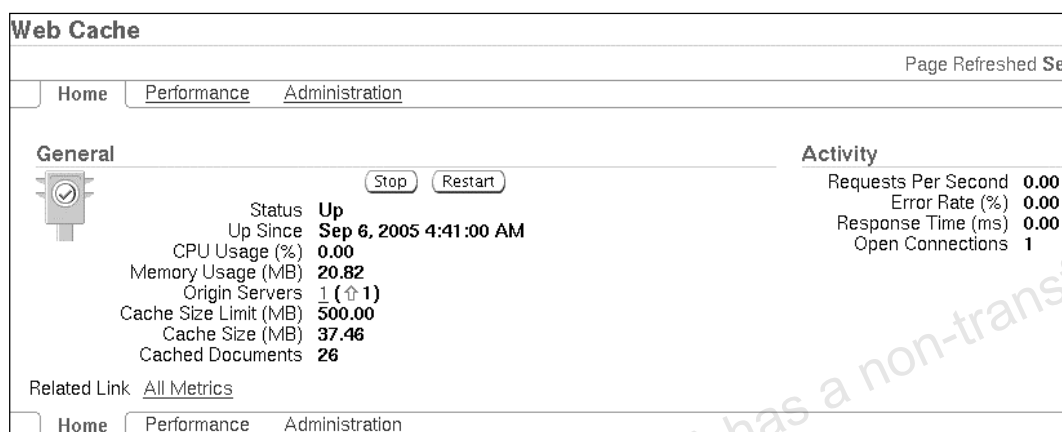
The status column in the example indicates that both the OracleAS Web Cache admin server (WebCacheAdmin) and the cache server (WebCache) are running (status = Alive).

If the processes are not running, then you can start OracleAS Web Cache by using the command shown in the slide.

In an OracleAS Web Cache stand-alone environment, you can use the `webcachectl` utility to start and stop OracleAS Web Cache.

Note: The Web Cache admin process manages the OracleAS Web Cache Manager interface, and the Web Cache server process manages the cache.

OracleAS Web Cache Home Page



OracleAS Web Cache Home Page

The Application Server Control Console is the starting point to manage or configure OracleAS Web Cache. Navigate to the System Components table, and click the Web Cache link to access the OracleAS Web Cache home page. This page is divided into two regions, each providing a different functionality.

You can gather general information about the status of your server and the time when the cache is started. In addition, you can also obtain high-level information about CPU and memory usage. The performance statistics for OracleAS Web Cache includes the following:

- The percentage of CPU that is being used for OracleAS Web Cache. As traffic increases, CPU utilization increases. If the CPU usage reaches beyond 80%, it is an indication that either OracleAS Web Cache is very busy or some other processes running on the same machine are competing for resources with OracleAS Web Cache. To improve the efficiency of OracleAS Web Cache, consider disabling any components that you are not currently using as part of this application instance. If the load is very high, you can also consider upgrading the cache computer.
- The memory usage parameter, which defines the number of megabytes being utilized for cache memory. To avoid swapping documents in and out of the cache, you must configure enough memory for the cache. Generally, the amount of memory, which is also the maximum cache size, must be set to at least 256 MB.

Oracle Application Server 10g R2: Administration I 8-13

OracleAS Web Cache Home Page (continued)

- Number of origin servers and their statuses
- The maximum cache size as specified by the Maximum Cache Size (MB) field on the Resource Limits and Timeouts page
- Size of the valid documents in the cache defined by the cache size (MB)
- Number of cached documents

If you have not configured the Web server or Web Cache correctly, the response time can be slower than expected. You can monitor the following settings from the Activity section:

- Number of requests that OracleAS Web Cache is serving each second
- Percentage of total requests that OracleAS Web Cache responds to by serving error pages due to a network or a busy Web site
- Difference between the time it takes OracleAS Web Cache to send responses and receive requests
- Number of incoming open connections to the OracleAS Web Cache server and outgoing open connections to the origin servers

The Performance page provides more sophisticated performance data about the Web server activity, the application Web server itself, and popular cached documents.

The Administration page provides access to administration and configuration tasks of OracleAS Web Cache.

Application Server Control Console: Web Cache Performance Page

All Sites <input type="button" value="Go"/>			
All Sites			
Requests		Errors	
Current Per Second	0.00		
Average Per Second	0.00		
Maximum Per Second	29.00		
Total Requests	1,057		
Current Origin Server Backlog	0		
Maximum Origin Server Backlog	0		
Related Links	Origin Servers		
	Popular Requests		
Hits		Misses	
	Current Per Second	Current Requests (%)	Average Per Second
Fresh Hits	0.00	0.00	0.00
Stale Hits	0.00	0.00	0.00
Total Hits	0.00	0.00	0.00
	Current Per Second	Current Requests (%)	Average Per Second
Cacheable Misses	0.00	0.00	0.00
Noncacheable Misses	0.00	0.00	0.00
Errors	0.00	0.00	0.00
Refreshes	0.00	0.00	0.00
Total Misses	0.00	0.00	0.00

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Application Server Control Console: Web Cache Performance Page

From the Web Cache Performance page, you can monitor the overall cache performance and the efficiency of the cache. You can specifically view the high-level metrics such as requests served in the Requests section, cache hits and misses in the Hits and Misses sections, compression performance in the Compression section, and errors in the Errors section.

Use the Performance page to tune OracleAS Web Cache, to monitor status and performance of the origin servers, and to view the most popular cache requests.

Note: For additional information about how to tune OracleAS Web Cache, refer to the *Web Cache Administrator's Guide 10g Release 2 (10.1.2)* and the Web Cache section of the *Oracle Application Server Performance Guide 10g (10.1.2)*.

Modifying Security Settings

Web Cache	
Home Performance Administration	
Operations Invalidation Rollover Log Files	
Cluster Properties Members and Properties Cluster Operations	
Properties	
Application Origin Servers Sites Sessions Rules	
Web Cache Ports Resource Limits and Timeouts Security Auto-Restart	

Security	
Administrator User Password The administrator is permitted to start, stop, and restart Web Cache, change configuration settings in Web Cache Manager, send invalidation requests with Web Cache Manager, and send statistics monitoring requests. All caches in the cluster must have the same administrator password.	
User	administrator
Old Password	*****
New Password	*****
Confirm New Password	*****

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Modifying Security Settings

When OracleAS Web Cache is first installed, it is set up with default passwords for administration and invalidation. The passwords are stored in the `webcache.xml` file in an encrypted form.

Administration user: The OracleAS Web Cache administrator user can perform the configuration and operational tasks. The administrator user (`ias_admin`) has the same password as that of the `ias_admin` user. You can change the password for the administrator as follows:

1. Click Administration on the Web Cache property page.
2. Click Security in the Properties section. The Security page appears.
3. On the Security page, enter the password for the `ias_admin` user in the Old Password field and a new password between 4 and 20 characters in the New Password and Confirm New Password fields.
4. Click OK.
5. Click Restart Web Cache to apply the security changes.

Modifying Security Settings (continued)

Invalidation user: Optionally, change the password for the invalidation administrator. The invalidation administrator has the `invalidator` user ID, whose default password `invalidator` is set up during installation.

Current trusted subnets: You can change the trusted subnet or trusted host from which OracleAS Web Cache and invalidation administration can take place. By default, the computer on which you installed OracleAS Web Cache is the trusted host. To change the trusted subnet or trusted host, perform the following steps:

1. On the Security page, select one of the following options:
 - **Trust all computers in all the subnets on the network:** Allows administration and invalidation requests from all computers in all the subnets in the network
 - **Trust only this machine:** Allows administration and invalidation requests from this computer only
 - **Trust only these IP addresses:** Allows administration and invalidation requests from all IP addresses that you enter in a comma-separated list
2. Click OK.
3. Click Restart Web Cache to restart OracleAS Web Cache to reread the configuration.

You can change the user ID and group ID for OracleAS Web Cache executables on UNIX. By default, the user that performed the installation is the owner of OracleAS Web Cache executables. Only this user can execute the `webcachectl start` and `stop` commands.

The security settings and the operational ports affect the functioning of other dependent components, such as OracleAS Portal. After you make a change to the security settings or the ports, you should ensure that the dependent components, such as OracleAS Portal, are synchronized with the changes.

Configuring Listening Ports for Requests

Operations
[Invalidation](#)
[Rollover Log Files](#)

Cluster Properties
[Members and Properties](#)
[Cluster Operations](#)

Properties
Application
[Origin Servers](#)
[Sites](#)
[Sessions](#)
[Rules](#)
Web Cache
[Ports](#)
[Resource Limits](#)
[Security](#)

Ports

Listen Ports

Web Cache receives browser requests on listen ports. Changing listen ports may affect other Oracle Application Server components. Ports that are not in conflict. In particular, page redirects in the origin server may require that ports are not in conflict.

IP Address ▲	Port	Protocol	Require Client-Side Certificates For HTTPS	Wallet For HTTPS
*	7778	HTTP ▼	Not Required ▼	
		HTTP ▼	Not Required ▼	

Add a row

ORACLE

Copyright © 2005, Oracle. All rights reserved.

The default port for OracleAS Web Cache depends upon the operating system in use. This port cannot be in use by the server. By default, OracleAS Web Cache is configured to listen for the HTTP protocol on port 7778. If this port is in use, the installation procedure attempts to assign other port numbers from a range of possible port numbers. You can add ports, if necessary.

- Oracle Application Server 10g R2: Administration I 8-18

Changing Operations Ports

Ports
Page Refreshed Sep 6, 2005 9:2

Listen Ports

Web Cache receives browser requests on listen ports. Changing listen ports may affect settings in site definitions. Changing address for Web Cache may affect other Oracle Application Server components. Parameters in other components may need to be changed if that ports are not in conflict. In particular, page redirects in the origin server may require reconfiguration.

IP Address ▲	Port	Protocol	Require Client-Side Certificates For HTTPS	Wallet For HTTPS	Delete
*	7778	HTTP ▼	Not Required ▼		
		HTTP ▼	Not Required ▼		

Operation Ports

Web Cache also receives administration, invalidation, and statistics monitoring requests on operation ports. These ports are used unless there are port conflicts.

	IP Address	Port	Protocol	Require Client-Side Certificates For HTTPS	Wallet For HTTPS
Administration	*	9400	HTTP ▼	Not Required ▼	
Invalidation	*	9401	HTTP ▼	Not Required ▼	
Statistics	*	9402	HTTP ▼	Not Required ▼	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Changing Operations Ports

For a Web administrator, it may be necessary to change the operation port numbers according to port conflicts. This can be achieved by using Application Server Control:

1. On the Application Server Control Web Cache Administration page, click Ports in the Web Cache section.
2. On the Ports page, scroll down to the Operation Ports section.
3. Change the desired port numbers, and click OK to apply any changes.

Note: Port numbers less than 1024 are reserved for use by privileged processes on Linux/UNIX. If you want to configure OracleAS Web Cache on a port such as 80, then run the webcached executable with root privileges:

- From \$ORACLE_HOME/webcache/bin, execute webcache_setuser.sh setroot user_ID, where user_ID is the user that performed the installation.
- Log out and log in as the user that performed the installation, and start OracleAS Web Cache.

Specifying Origin Server Settings

Origin Servers

Configure the location of the origin servers. Web Cache sends requests to origin servers for internal sites and proxy servers for external sites protected by a firewall. These settings are required for load balancing, failover, and site-to-server mappings.

Page Refreshed Sep 6, 2005 9:42:31 AM

Edit Delete

Select Host ▲

EDRSR16P1

Create Origin Server

Specify the settings for the origin server. In order for Web Cache to forward requests to an origin server, you must map a site to the server on the Sites page.

Cancel OK

* Host

* Port

Protocol HTTP ▼

* Capacity 100

Maximum concurrent connections

☒ Routing Enabled

Controls whether the cache sends requests to this origin server.

Failover

If another cache cluster member repeatedly fails to respond, Web Cache will assume that the cache server process for that member has failed. Then, it will periodically ping the cache server until it receives a successful response.

Fallover Threshold 5

Number of failures before the server is assumed down

Ping URL

Periodically ping this URL to see if the origin server is back up.

Ping Frequency (seconds) 10

How often to ping origin server

Proxy Web Server

Select proxy server for external sites protected by a firewall. If the proxy server does not require a username and password, leave those fields blank.

☐ Proxy Web Server

User

Password

Confirm Password

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specifying Origin Server Settings

You can configure OracleAS Web Cache with the application from where it gets its contents. By default, the listening port and host name of Oracle HTTP Server are configured. OracleAS Web Cache only forwards requests to a configured origin server if the server is mapped to a Web site on the Site-to-Server Mapping page.

To configure OracleAS Web Cache with application Web server information, perform the following steps:

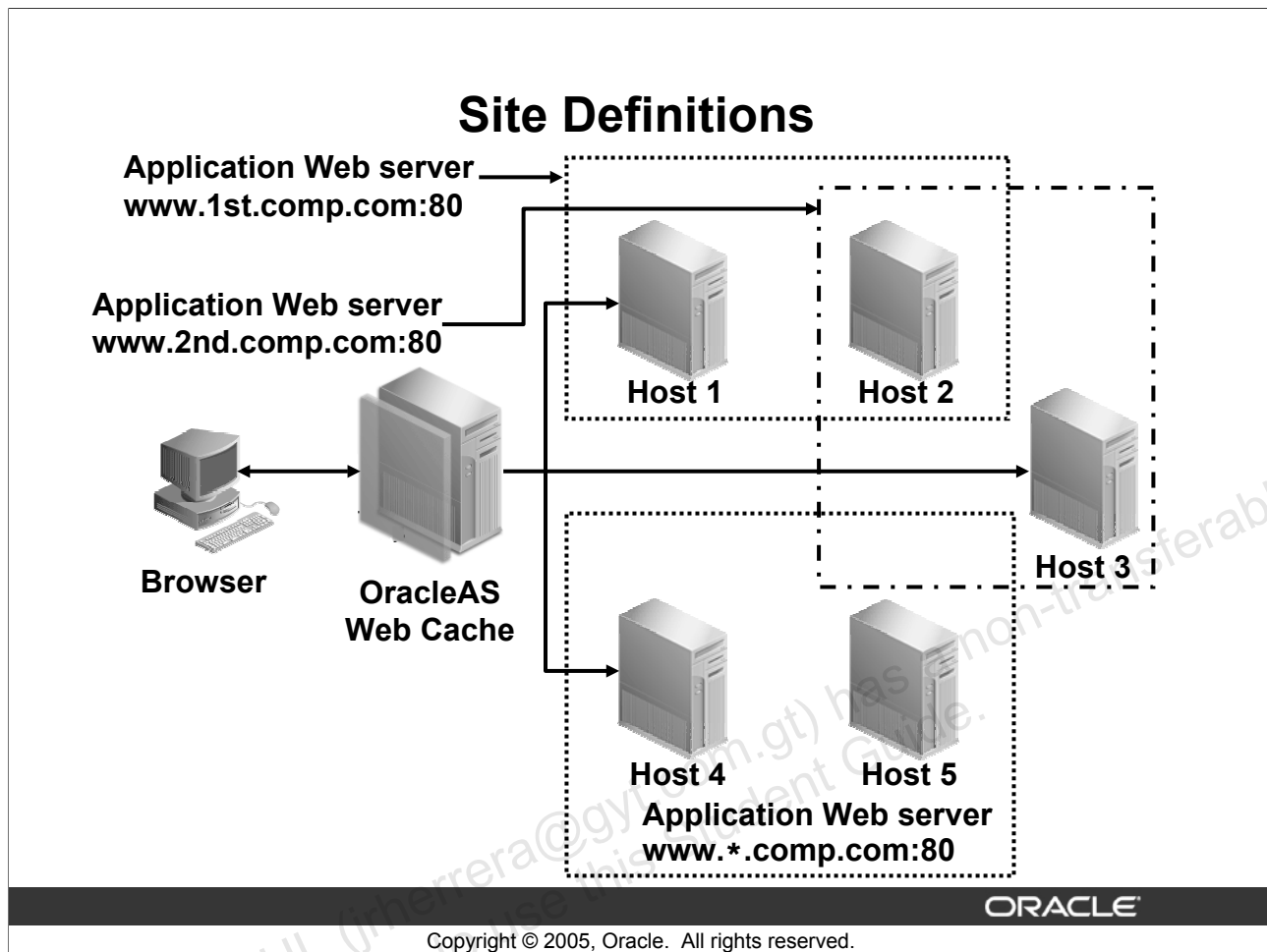
1. Click Origin Servers in the Properties section to display the origin server page.
2. Click Create to display the Create Origin Server page.
3. In the Host field, enter the host name of the application or proxy server.
4. In the Port field, enter the listening port from which the application or proxy server receives OracleAS Web Cache requests.
5. In the Capacity field, enter the maximum number of concurrent connections that the application or proxy server can accept.
The maximum number of concurrent connections that a server can handle is determined by load testing the application Web server or proxy server until it runs out of CPU, or responds slowly, or until a back-end database reaches full capacity.
6. From the Protocol list, select either HTTP to send HTTP requests on the port or HTTPS to send HTTPS requests on the port.

Oracle Application Server 10g R2: Administration I 8-20

Specifying Origin Server Settings (continued)

7. In the Failover Threshold field, enter the number of allowed continuous request failures before OracleAS Web Cache considers the origin server down. The default is five requests.
8. In the Ping URL field, enter the URL that OracleAS Web Cache uses to poll an origin server that has reached its failover threshold.
Instead of using a static URL, you should use a URL that checks the health of the application logic on the origin server and returns the appropriate HTTP 200 or 500 status codes.
9. In the Ping Frequency (seconds) field, enter the time in seconds for which OracleAS Web Cache polls an origin server that has reached its failover threshold. The default is 10 seconds.
10. Click OK.

Note: OracleAS Web Cache only forwards requests to a configured application Web server or proxy server if the server is mapped to a Web site on the Site-to-Server Mapping page, which is explained in the following slide.



Site Definitions

OracleAS Web Cache caches and assembles dynamic content for one or more Web sites. When OracleAS Web Cache receives a browser request for a document, it determines the destination site by using one of the following elements:

- Host request-header field from the request
- Host portion of the requested URL
- `src` attribute of the ESI `<esi:include>` tag

OracleAS Web Cache then looks up the configured site settings and mappings to determine whether the site is supported, and the application Web servers or proxy servers and caching rules for the site.

The illustration in the slide shows Web sites `www.1st.comp.com:80` and `www.2nd.comp.com:80` that have site aliases of `1st.company.com:80` and `2nd.company.com:80`, respectively.

The site-to-application Web server mappings are as follows:

- `www.1st.company.com` maps to application Web servers `host1` and `host2`.
- `www.2nd.company.com` maps to application Web servers `host2` and `host3`.
- `www.*.company.com` maps to `host4` and `host5`.

Configuring Site Definitions and Mapping to the Origin Server

The screenshot displays the Oracle Web Cache Administration console. On the left is a navigation pane with sections: **Operations** (Invalidation, Rollover Log Files), **Cluster Properties** (Members and Properties, Cluster Operations), and **Properties** (Application, Origin Servers, Sites, Sessions, Rules). The **Sites** section is selected. The main content area shows the 'Named Sites Definitions' page with a 'Create Named Site' dialog box open. The dialog has tabs for 'General' and 'Advanced'. The 'General' tab is active, showing fields for '* Host' (with a hint: '(Example: www.company.com. Do not use wildcards.)'), '* Port', and 'Prefix'. Below these is the 'Origin Servers' section, which includes radio buttons for 'Application Web Servers' and 'Proxy Web Servers', and a list of 'Available Origin Servers' with 'EDRSR16P1:7779' selected. A 'Selected Origin Servers' list is also present. The dialog has 'Cancel' and 'OK' buttons. A watermark 'HERPEDIA RAUL (www.herpedia.com.91) Guide' is visible across the image.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Site Definitions and Mapping to the Origin Server

To configure a site definition, perform the following steps:

1. On the Web Cache Administration page of the Application Server Control Console, scroll down to the Properties section, and click Sites. The Sites page appears.
2. On the Sites page, under Named Sites Definitions, click Create to add a new site.
3. In the Host field, enter the host name of the site, such as `myserver.us.oracle.com`. Do not use the wildcard `*` to represent multiple sites.
4. In the Port field, enter the port number from which the Web site is listening for incoming HTTP requests. The port number should be of the port used in browser requests.
5. In the Prefix field, enter the path prefix of the URLs to distinguish this site from another site that shares the same host name. Do not include the file name or embedded URL parameters in the prefix and make sure that the prefix starts with `/`. When defining a site definition, you can specify a URL path prefix for those sites that share the same host name. For example, by specifying `/hrapp` and `/financeapp` as prefixes, you can treat `http://www.myserver.com/hrapp` and `http://www.myserver.com/financeapp` as two distinct sites.

Configuring Site Definitions and Mapping to the Origin Server (continued)

After specifying a site definition, you must map the site to the origin servers:

6. In the Origin Servers section, select the application Web server or proxy server for the site. From the Available Origin Servers list, select the origin server. Click Move to move the server to the Selected Origin Servers list.
7. In the Alias section, click Add Another Row to add an alias for this site.
Many sites are represented by one or more aliases. OracleAS Web Cache recognizes and caches requests for a site and its aliases.
For example, the site `www.mycompany.com:80` may have an alias of `company.com:80`. By specifying this alias, OracleAS Web Cache caches the same content from either `company.com:80` or `www.company.com:80`. If a request includes a site alias that is not configured, then OracleAS Web Cache sends the request to the default site.
8. Click OK.
9. Restart Web Cache.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Caching Rules: Overview

- **Caching rules specify whether or not to cache content, and determine what content to cache:**
 - **Static documents**
 - **Multiple-version URLs**
 - **Personalized pages**
 - **Pages that support session tracking**
 - **HTTP error messages**
 - **URLs that match with regular expressions**
 - **URL trees that contain a document or a subtree**
- **Caching is based on priority rules (top is the highest).**
- **Rules also specify the caching of static versus dynamic content.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Caching Rules: Overview

Caching rules enable you to define various URL expressions. These URL expressions can represent one or more documents. If the expression represents several documents, such as complete directories and subdirectories, it is called a subtree of URLs.

The priority at the top overrules the expressions under it.

You can specify caching rules in the following formats:

- Regular expressions
- File extension
- Path prefix

This enables easier specification of caching rules.

Surrogate-Capability

- **OracleAS Web Cache appends a Surrogate-Capability request-header field to an object's HTTP request message to detect OracleAS Web Cache.**
- **OracleAS Web Cache adds version and diagnostic information to the Server response-header field.**

```
Surrogate-Control:[content=content_type, content_type,...] [no-store] [no-store-remote] [max-age=expiration_time[+removal_time]] [vary=headers(header header...)] [cookie(cookie_name cookie_name...)] [compress=yes|no]
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Surrogate-Capability

Caching rules determine which objects are cached. When a caching rule for a particular URL is first created, the objects contained within the URL are not cached until the browser requests for them. When an object is first requested, OracleAS Web Cache appends a Surrogate-Capability request-header field to the object. The Surrogate-Capability request-header field identifies that the object passed through the cache. OracleAS Web Cache then sends the request to the origin server. If the requested object is specified as one of the objects to cache, then OracleAS Web Cache caches the object for subsequent requests. On subsequent requests for the object, OracleAS Web Cache serves the object from its cache to the browser.

In addition to configuring a caching rule with Application Server Control, you can configure the caching attributes for a specific object within a Surrogate-Control response-header field. OracleAS Web Cache uses the following priority to determine object caching:

- Surrogate-Control response header
- Caching rules
- Other HTTP headers such as Authorization
- Cookie values
- Cache-Control response header

Surrogate-Capability (continued)

The Surrogate-Control response-header field enables the origin server to override the caching rules. When both a Surrogate-Control response header and a caching rule for the same object are present, OracleAS Web Cache merges the two. If there is a conflict between the Surrogate-Control response header and a caching rule, then OracleAS Web Cache uses the settings from the Surrogate-Control response header.

In addition to, or as an alternative to, creating caching rules, application developers can store many of the caching attributes in the header of an HTTP response message.

The following attributes are not supported when storing caching attributes in the header:

- ESI Output Permission
- Session/Personalized attribute-related caching rules
- HTTP error caching

To enable this feature, configure the HTTP response with:

```
Surrogate-Control:[content=content_type, content_type,...] [no-store] [no-store-remote] [max-age=expiration_time[+removal_time]] [vary=headers(header header...)] [cookie(cookie_name cookie_name...)] [compress=yes|no]
```

where:

- `content` denotes the type of processing
- `no-store` means do not cache the object
- `max-age` denotes the time (in seconds) to expire the object after it enters the cache
- `vary` denotes the HTTP request headers or cookies from which OracleAS Web Cache uses to cache and identify multiple-version objects
- `compress` indicates that OracleAS Web Cache can serve compressed objects

Predefined Caching Rules

Rules						
Rules instruct the cache how to react to each request. A rule has two parts: Selector and Instructions. A request is compared with the Selectors. If there is a match, then the cache follows the Instructions. Rules are ordered; only the first matching rule is honored. Page Refreshed Sep 6, 2005 9:57:16 AM						
View Site: <input type="text" value="EDRSR16P1:7778"/> View Columns: <input type="text" value="Overview"/> Create Reorder						
Create Like Edit Delete						
Select	Order	Name	Selector Summary	Enabled	Instructions	
	1	cache wireless rm	Reg: /ptg/rm	✓	✓	Expires: As Specified in HTTP Header, Refresh: Immediately Sessions: PAsid, PAconnxn, PAuserid
	Global 1	cache image	Reg: \.(gif jpe?g png bmp)\$	✓	✓	Expires: Max Time in Cache, 60 Minutes, Refresh: Within 6 Minutes
	Global 2	cache compress css	Ext: .css	✓	✓	Compress, except Netscape 4.x Expires: Max Time in Cache, 5 Minutes, Refresh: Immediately
	Global 3	cache uix-idev js	Reg: /jsLibs/.*.js\$	✓	✓	Expires: Max Time in Cache, 5 Minutes, Refresh: Immediately

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Predefined Caching Rules





In Application Server Control, on the Web Cache Administration page, scroll down to the Properties section and click Rules to see the caching rules. The following rules are predefined by Oracle:

- **/ptg/rm:** Caches the default Oracle Application Server Wireless servlet. This rule is necessary for OracleAS Wireless to use OracleAS Web Cache to cache transformations. If you change the servlet mount point to something other than /ptg/rm, then you must update this rule.
- **\.(gif|jpe?g|png|bmp)\$:** Caches all files ending with .gif, .jpeg or .jpg, .png, and .bmp
- **.css:** Caches cascading style sheet files
- **./jsLibs/.*.js\$:** Caches all UIX Oracle JDeveloper .js (JavaScript) files without compression
- **\.js\$:** Caches .js (JavaScript) files

The slide illustrates a portion (from rule 1 to rule 4) of the predefined caching rules.

Rules for Caching, Personalization, and Compression

The Order column specifies the priority in which the rules are processed:

Select	Order	Name	Selector Summary	Enabled	Instructions	
					Cache Summary	
	1	cache wireless_rm	Reg: /ptg/rm	✓	✓	Expires: As Specified in HTTP Header, Refresh: Immediately Sessions: PAsid, PAconnxn, PAuserid
	Global 1	cache image	Reg: \.(gif jpe?g png bmp)\$	✓	✓	Expires: Max Time in Cache, 60 Minutes, Refresh: Within 6 Minutes
	Global 2	cache compress css	Ext: .css	✓	✓	Compress, except Netscape 4.x Expires: Max Time in Cache, 5 Minutes, Refresh: Immediately
	Global 3	cache uix-idev.js	Reg: /jsLibs/*\js\$	✓	✓	Expires: Max Time in Cache, 5 Minutes, Refresh: Immediately

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Rules for Caching, Personalization, and Compression

In the Caching Rules table, the second column from the left specifies the rule's priority, that is, the order in which the rules are applied. The first match is applied and subsequent rules are not followed.

For example, apply the rules in the following order:

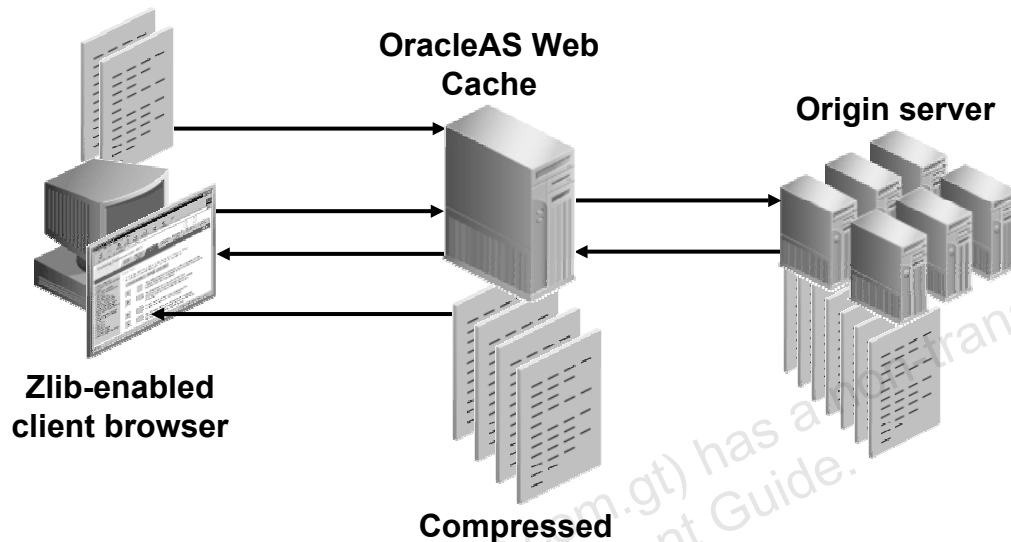
1. `^/abc/. *index\.htm$` to cache `index.htm[1]` files
2. `^/abc/bc/index\.htm$` defined because you do not want the index file from this specific directory to be cached

The result is that OracleAS Web Cache caches the `index.htm` file in both directories, because the first rule overrides the second. Because the first rule matches, the second rule is not applied.

Alternatively, applying the rules in the following order has the desired result:

1. `^/abc/bc/index\.htm$` should not be cached.
2. `^/abc/. *index\.htm$` should be cached.

Streamed Delivery of Compressed Content



Copyright © 2005, Oracle. All rights reserved.

Streamed Delivery of Compressed Content

Compression is a technology for reducing the content. By enabling compression, you are instructing OracleAS Web Cache to compress pages by using the zlib compression technique. Most browsers support zlib compression. Browser support for compression information is included in the Accept-Encoding HTTP header. For example, if a browser supports compression, its HTTP request includes a header similar to:

Accept-Encoding: zlib

OracleAS Web Cache determines the value of the Accept-Encoding header and checks that it contains zlib. Thus, OracleAS Web Cache sends compressed content according to the value provided in the Accept-Encoding field in the client request. If the client browser does not support zlib, OracleAS Web Cache sends the content back uncompressed.

Using zlib, on average, OracleAS Web Cache can compress text files, such as HTML and XML, by a factor of 10. Because compressed objects are smaller in size, they require less bandwidth to transmit and can be delivered faster to browsers. Compression reduces transmission costs, and end users enjoy more rapid response times.

Streamed Delivery of Compressed Content (continued)

For cacheable content that an administrator or developer chooses to compress, OracleAS Web Cache stores both compressed and uncompressed versions in the cache. If an object retrieved from the origin Web server already contains a Content-Encoding response header, which is typically used to denote compression, then OracleAS Web Cache does not compress it. Noncacheable responses can also be compressed dynamically if the administrator chooses this configuration option.

In Oracle Application Server 10g (9.0.4), only after all the responses are compressed, OracleAS Web Cache sends it over to the client. Instead, compressed content must be sent at regular intervals; that is, the content must be streamed to the client. The benefit of this is to eliminate large memory usage for noncacheable miss documents and to eliminate spikes in response times. Another benefit is faster response time for smaller documents. An example to explain this is that the CPU time required for compression at the fastest level for an HTML document of 900 KB is around 500 milliseconds. Thus, if a request for a small document (irrespective of whether it is cacheable or noncacheable) comes just after the request for this 900 KB document, then OracleAS Web Cache can start processing the request for the small document at an earlier interval.

Streaming is the capability of OracleAS Web Cache to start sending the response to the client even though the whole response is not yet compressed in OracleAS Web Cache or received from the operating system. This is applicable to both compressed and noncompressed data.

Compression is performed in an improved manner in Oracle Application Server 10g Release 2. OracleAS Web Cache now has the ability to stream compressed content. The end user does not need to wait for the entire document to arrive at the browser before viewing it. The size of the compressed content does not change; however, users can see the content earlier.

Oracle recommends not compressing images (such as GIFs and JPEGs), as well as executables, and files that are already zipped with utilities such as WinZip and GZIP.

Compressing these files incurs additional overhead without the benefits of compression. In addition, Oracle recommends not compressing JavaScript and cascading style sheet.

Even if compression is turned on, OracleAS Web Cache does not compress objects containing the following:

- Content-Encoding header, which is typically used to denote compression
- Content-Disposition header, which is typically used for attachments

Creating Caching Rules

Create Rule	
<p>Rules instruct the cache how to react to each request. A rule has two parts: Selector and Instructions. First the request is compared with the Selector. If there is a match, then the cache follows the Instructions. Rules are ordered; only the first matching rule is honored.</p>	
<div> <div>General</div> <div>Advanced Caching Instructions</div> </div>	
* Name	Site <input type="text" value="EDRSR16P1:7778"/>
Description	<input checked="" type="checkbox"/> Enabled
Selector To match this rule, a request must match all the parts of the Selector. Match URL By <input type="text" value="File Extension"/> *	
▶ Show HTTP Methods and Parameters	
Instructions	
Caching <input checked="" type="radio"/> Cache Expiration of Cached Response <input type="text" value="Never"/> <input type="radio"/> Do not cache Related Link Expiration Policies	
Compression Web Cache can compress responses whether or not they are cached. Do not compress objects which are already compressed, such as JPEG files. Netscape 4.x browsers are unable to uncompress files which are included in HTML content, such as JavaScript files.	
<input checked="" type="radio"/> Do not compress <input type="radio"/> Compress for all browsers <input type="radio"/> Compress for all browsers except Netscape 4.x	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Creating Caching Rules

To specify caching rules, perform the following steps from the Application Server Control Web Cache Administration page:

1. On the Administration page, scroll down to the Properties section. Click Rules. The Rules page appears.
2. On the Caching Rules page, click Create to create a new rule if no rule exists. If rules already exist, select a rule, and then click Edit to change the settings for the rule. Click Reorder to change the priority order of the rules. The Create Rule window appears.
3. In the Name field, enter a string that uniquely defines the caching rule.
4. In the Description field, enter a description for the caching rule.
5. From the Site list, select the site for which to apply this rule.
6. Select the Enabled check box to enable the caching rule. By enabling or disabling a rule, you do not have to delete or remove a rule and then re-create it. You can retain the copy of the rule and enable it as and when required.

Creating Caching Rules (continued)

7. In the Match URL By drop-down list, select:
 - File Extension to apply the caching rule to objects ending in a particular file extension
 - Path Prefix to apply the rule to objects matching a path prefix
 - Regular Expression to apply the caching rule to object matching regular expression syntax

In the adjacent field, based on the selection you made for Match URL By, enter a corresponding expression. For regular expression syntax, remember to use ^ to denote the start of the URL and \$ to denote the end if necessary.

8. Select “Cache” or “Do not cache” for the documents contained within the URL.
9. The Compression field enables you to compress documents. If a document retrieved from the application Web server already contains a Content-Encoding header, which is typically used to denote compression, then OracleAS Web Cache does not compress it.
10. Click Advanced Caching Instructions to specify additional instructions.

Edit Caching Rules

Edit Rule: cache compress html

Rules instruct the cache how to react to each request. A rule has two parts: Selector and Instructions. First the request is compared with the Selector. If there is a match, then the cache follows the Instructions. Rules are ordered; only the first matching rule is honored. Cancel OK

General **Advanced Caching Instructions**

Name **cache compress** Site **All Sites**
html

Cache Multiple Versions of an Object

For Selected Cookies
The content of a response might depend upon a cookie value. For example, a page might use customer information cookie. You cache multiple versions of this page for each customer by selecting the customer information cookie.

Cookie Name	Cache If Absent	Delete
(No Cookies found)		

Add a Cookie

For Other HTTP Headers
The content of a response might depend upon a HTTP request header. For example, a page may provide English, Spanish, and Japanese versions through the Accept-Language header request field. You cache multiple versions of this page by selecting the Accept-Language header.

Available Headers

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language
- Authorization
- Referer
- User-Agent

Selected Headers

Move Move All Remove Remove All

More headers
Enter headers separated by commas or spaces.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Edit Caching Rules

To edit the already existing rules, for example for HTML files, select the rule on the Rules page, and click Edit. The Edit Rule page is displayed. The first part contains exactly the same fields that are discussed before. Click Advanced Caching Instructions to specify additional instructions. Here you can specify settings for the following additional fields:

- **Cache Multiple Versions of a Document**

The reason for caching multiple versions of a document is that pages with the same URL may actually have content that varies slightly based on the cookies sent by the browser. That is, the pages vary depending on the cookie value. Therefore, you need to specify the cookie value so that OracleAS Web Cache can distinguish (and determine) one cacheable copy from another:

- **For Selected Cookies:** Select the required cookies to cache multiple-version objects that depend on cookie values. If you do not see a cookie that can be applied to these objects, click Add a Cookie.
- **For Other HTTP Headers:** Select the HTTP request headers whose values OracleAS Web Cache uses to cache and identify multiple-version objects. If you require other headers, enter them in the More Headers field.

Note: OracleAS Web Cache does not interpret the values of these HTTP request headers. If the values for two pages are different, OracleAS Web Cache caches both pages separately.

Oracle Application Server 10g R2: Administration I 8-34

Edit Caching Rules (continued)

- **Avoid Unwanted Copies**

This section lists the embedded URL or POST body parameters for which OracleAS Web Cache ignores the values. If you need to define additional embedded URL or POST body parameters for which OracleAS Web Cache ignores the values, then click the link next to Global URL Parameters to Ignore for global parameters to apply to all sites, or Site-Specific URL Parameters to Ignore to apply to site-specific parameters. By configuring OracleAS Web Cache to ignore the value of embedded URL or POST body parameters, you enable OracleAS Web Cache to serve once-cached object to multiple sessions requesting the same page.

- **Session-Related Caching**

To cache or serve objects based on session or personalized attribute information contained within a cookie, embedded URL parameter, or POST body parameter, select the session and corresponding policy.

- **Cache Error Responses**

A particular request from a client may result in an error response. Error responses are not normally cached. If there is a problem in the origin server that does not result in a 200 OK HTTP response status for a rule, then OracleAS Web Cache must be provided with an option to serve a cached HTTP error. This saves origin server resources. However, the origin server also must generate this HTTP error itself. Select the error codes from the Available Error Codes list, or enter the error code number in the Additional Error Responses to Cache field to cache responses for errors.

Caching Dynamic and Partial Pages

Caching dynamic pages:

- **Cookies or embedded URLs enable OracleAS Web Cache to recognize caching rules for pages with:**
 - **Multiple versions of the same document**
 - **Personalized attributes**
 - **Session information**

Caching partial pages:

- **OracleAS Web Cache provides dynamic assembly of Web pages with both cacheable and noncacheable page fragments by using ESI tags.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Caching Dynamic and Partial Pages

Caching Dynamically Generated Content

Most Web pages today are dynamically generated before delivery to the browser. Web developers frequently use database-driven technologies for complex Web sites that are easier to modify and maintain. Examples of pages that are dynamically generated include:

- A product catalog, where information on pricing and inventory might vary from one moment to the next
- Auction views, which must be regenerated after each successful bid is processed
- Search results, which can change as catalog items are added and removed

Because of invalidation, OracleAS Web Cache knows which documents are valid and which documents are invalid. This is especially important for dynamically generated content that changes frequently.

Most static caches and content distribution services have no mechanism to verify the consistency of dynamically generated Web pages with the data sources used to create them. Therefore, it is difficult for these services to know when content has changed.

Caching Dynamic and Partial Pages (continued)

Caching Dynamically Generated Content (continued)

For dynamically generated pages, browsers pass information about themselves as a browser to the application Web server, enabling the application Web server to serve appropriate content to the browser.

The HTTP protocol has a way for browsers and application Web servers to share information, such as session or category information, in message headers that browsers pass with every request to the application Web server. This message header can contain a cookie.

Cookies are stored on the browser's file system and are often used for identifying users who revisit Web sites. Many users choose to disable cookies in their browsers out of privacy concerns. For this reason, application Web servers often embed parameter information in the URL.

OracleAS Web Cache can recognize both cookies and embedded URL parameters, enabling it to follow caching rules for pages with:

- Multiple versions of the same document
- Personalized attributes
- Session information

Content Assembly and Partial Page Caching

The new Edge Side Includes (ESI) functionality enables OracleAS Web Cache to aggregate portions of Web pages and reassemble them for individual users. ESI is the result of a joint development effort between Oracle and Akamai, and has now been proposed as an open standard. ESI is a simple markup language that application developers use to identify content fragments for dynamic assembly in edge servers, such as OracleAS Web Cache and third-party content delivery networks (CDNs). The partial-page caching functionality that ESI enables is especially useful for Web pages that contain targeted banner advertisements, individual account information, or other user-unique elements that should not be cached. With ESI, the edge server can store all the common elements of a Web page and query the database or other content repositories only for any highly personalized objects. By uniquely identifying common elements (such as stock quotes, weather reports, news, or graphics) that can be shared among different Web pages, only one copy of each element needs to be cached, invalidated, and revalidated, thus saving valuable resources across all the layers of OracleAS Infrastructure. More HTML content can be cached, assembled, and delivered by OracleAS Web Cache when requested. Furthermore, page assembly can be conditional, based on the information provided in HTTP request headers or end-user cookies.

Expiration Rules

- **When a cached object has a predictable time for usefulness, you can specify the expiration rule for that object.**
- **There are three options for setting expiration rules:**
 - Specified as per the HTTP Expires header
 - Specified time after entry into cache
 - Specified time after the document is created
- **Expired objects can be processed in two ways:**
 - Refresh immediately
 - Refresh on demand and no later than the specified time after expiration

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Expiration Rules

You would specify the expiration rule for a cacheable object if you can predict its duration for usefulness. When an object expires, OracleAS Web Cache marks it as invalid. There are three ways to set expiration rules with OracleAS Web Cache:

- **As Specified in the HTTP Expires Header**

This is the default option. Expiration is based on the Expires header that is generated by the origin Web server.

- **Max Time in Cache**

Expiration is based on when the object is inserted into the cache.

- **Max Time Since Document Created**

Expiration is based on when the object is created. This option relies on the Last-Modified header generated by the origin Web server.

A Web site that displays weather forecasts and current climate conditions is an example of an application that would benefit from invalidation using the expiration policies. The Web pages relating to the climate conditions could be set to expire 30 minutes after the pages are created, thereby ensuring that users never receive outdated information.

Defining Expiration Rules

Expire:

- **Based on the HTTP header**
- **After cache entry**
- **After document creation**

Remove documents:

- **Immediately**
- **Based on stale versus fresh**

The screenshot shows the 'Create Expiration Policy' dialog box. It contains a description of expiration and two main sections: 'Objects Expire' and 'Expired Objects Are Refreshed'. In the 'Objects Expire' section, 'As Specified in the HTTP Expires Header' is selected. In the 'Expired Objects Are Refreshed' section, 'Immediately' is selected. There are also input fields for 'Max Time Limit' and 'Removal Time Limit', both set to 'Seconds'.

Create Expiration Policy

Expiration determines an object's lifespan in the cache. A new copy of the object is fetched from the origin server either immediately or based on origin server availability. To set the expiration policy for an object, define a rule to match the request and associate the desired expiration policy.

Objects Expire

- ☒ As Specified in the HTTP Expires Header
- ☐ Max Time in Cache
- ☐ Max Time since Object Created

Max Time Limit Seconds

Expired Objects Are Refreshed

- ☒ Immediately
- ☐ Within a Time Limit, Based on Origin Server Availability

Removal Time Limit Seconds

Cancel OK

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Defining Expiration Rules

To create expiration rules, perform the following steps:

1. On the Application Server Control Web Cache Administration page, click Rules.
2. Scroll down and click Expiration Policies.
3. In the Oracle Web Cache Navigator pane, select Administering Web Sites > Caching Rules > Expiration Rules, and click Add.
4. In the Create Expiration Policy region, specify when documents should expire by selecting one of the options. The first option recognizes the expiration policy established for the documents that are already programmed with an HTTP Expires header. This is the default. To use this option, documents must be programmed to use the HTTP Expires header. The other two options enable you to set the expiration for rules that are specific to OracleAS Web Cache.
5. In the Expired Objects Are Refreshed region, specify how you want OracleAS Web Cache to process documents after they have expired:
 - Select Immediately to mark documents as invalid, and then refresh them immediately with updated content from the application Web servers.
 - Or, select Within a Time Limit, Based on Origin Server Availability, and enter the maximum amount of time for which the documents can reside in the cache.

Performance Assurance and Surge Protection

OracleAS Web Cache uses a patent-pending performance-assurance logic to ensure that:

- **Invalidation of a large number of objects in the cache does not result in a surge**
- **Load on the Web server and database is dampened**
- **Capacity heuristics are based on:**
 - **Request queue length**
 - **Document popularity**
 - **Document validity**
 - **Invalidation age**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

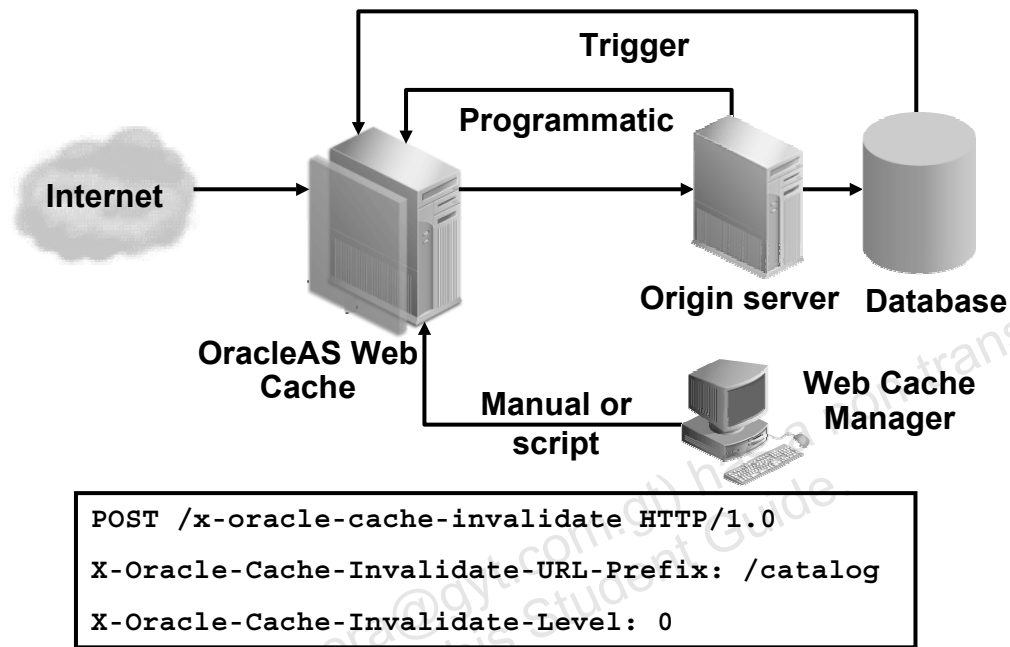
Performance Assurance and Surge Protection

When faced with the choice of serving some stale content or no content at all, most Web site administrators opt for the former. The result is that the overall Web site performance remains constant at the higher throughput levels sustainable by the cache, even with frequent content changes on the origin Web server and database.

OracleAS Web Cache uses a patent-pending performance assurance logic that determines which objects to refresh and which objects to serve stale, with minimal tradeoff between Web site performance and content consistency. The input for the heuristic algorithm is provided in part by the OracleAS Web Cache administrator and in part by the statistics gathered by OracleAS Web Cache during normal operations. The queue order of documents is based on the popularity of documents and the validity of documents assigned during invalidation. If the current load and capacity of the application Web server are not exceeded, the most popular and least valid documents are refreshed first.

OracleAS Web Cache passes requests for noncacheable or stale documents to the application Web servers. To prevent an overload of requests on the application Web servers, OracleAS Web Cache has a surge protection feature that enables you to set a limit on the number of concurrent requests that the application Web servers can handle. When the limit is reached, subsequent requests are queued.

Invalidation Messages



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Invalidation Messages

Invalidation mechanism is useful where the changes are more frequent and not predictable. OracleAS Web Cache provides a privileged `INVALIDATOR` user that can interactively mark cache contents invalid. You can also enable invalidation messages to be sent by the Web applications, by integrating the invalidation message format and grammar into the applications.

OracleAS Web Cache invalidation messages are HTTP POST requests that include XML data. The contents of the XML message body tell the cache which URLs to mark as invalid. The message in the slide, for example, invalidates all URLs starting with `/catalog` with a severity of 0 (that means, never serve these pages stale). Invalidation messages can be sent by using one of the following methods:

- Using the Application Server Control Web Cache Administration page or OracleAS Web Cache Manager
- Inline invalidation implemented as part of ESI template pages
- Using database triggers, scripts, or applications

Invalidation Messages (continued)

Manual Invalidation Using Application Server Control Web Cache Administration Page

The Web Cache Administration page provides an easy-to-use browser interface for invalidating cached objects. The advantage of the browser approach is that the administrator is isolated from the intricacies of the HTTP and XML formats and, consequently, there is less chance for error. The administrator specifies only which objects to invalidate and how invalid those objects should be.

Manual Invalidation Using Telnet

Manual invalidation can be performed through telnet. This involves generating an HTTP POST message containing the host name of the OracleAS Web Cache machine, the invalidation listening port number, authentication data, and the invalidation instructions.

Automatic Invalidation Using Database Triggers

Database triggers are procedures that are stored in the database and activated (or “fired”) when an INSERT, UPDATE, or DELETE statement is issued against a table. A trigger stored in the database can include SQL and PL/SQL or Java statements to be executed as a unit. Specifically, a trigger can be set so that when a database table is updated, an HTTP invalidation message is sent to OracleAS Web Cache. (Any database that supports triggers and HTTP can be used to invalidate the content stored in OracleAS Web Cache.)

Automatic Invalidation Using Scripts

Many Web sites use scripts for uploading new content to databases and file systems. For example, a large online book retailer might run a PERL script once per day to load new book listings and price changes into its catalog database. The retailer would want the price changes and availability listings to be reflected in the item views and search results currently cached in OracleAS Web Cache. To achieve this, the PERL script can be modified so that when the bulk-loading operation completes, the script sends an invalidation message to the cache invalidating all catalog views and search results.

Automatic Invalidation Using Applications

Invalidation messages can also originate from a Web site’s underlying application that is used to design Web pages. OracleAS Web Cache is shipped with an invalidation API that enables sites by using JSP and Java servlets to take advantage of automatically generated invalidation messages. With only moderate code changes, almost any application can automatically generate the XML and HTTP code required to invalidate cached content.

Invalidation Using Secondary Key

In previous releases, invalidation requests needed to specify either exact URLs or a set of URLs and headers matching a regular expression in order to invalidate cached objects. Because it can be difficult for applications to map URLs to the underlying data used to generate those URLs, OracleAS Web Cache invalidation has been extended to support search keys. Cached objects can now be associated with multiple application-specified search keys, with the URL-based key being the primary key.

Basic Content Invalidation

Invalidation

Indicate the objects you want to invalidate by specifying selection criteria. All the objects must have the same URL prefix. Invalidation is a way to remove obsolete content from the cache. Use these pages to invalidate objects based on their URLs. To ensure that you are removing only the objects you want to remove, you can preview the list of objects that meet the selection criteria you specify.

Invalidate these cached objects

- ☐ Specified objects
- ☒ All objects
- ☐ The single object that matches this URL

Cluster members to receive invalidation **All**

Preview Objects

Before submitting the invalidation operation, you can optionally preview the list of objects that will be invalidated. There may be a delay before the preview page is displayed, depending on the number of objects in the cache.

Invalidation: Removal Time

Specify when to remove objects from the cache.

- ☒ Remove objects immediately. Refresh each object as soon as there is a browser request for it.
- ☐ Remove objects as soon as possible. If the origin server is too busy to process requests, the cache may serve stale objects for a limited time.

Time Limit for Serving Stale objects Seconds

Copyright © 2005, Oracle. All rights reserved.

Basic Content Invalidation

Refreshing cache content by using expiration rules is sufficient when the changes are predictable, such as a weather forecast that needs to be refreshed every 30 minutes. The other method is to use invalidation for unpredictable changes, such as emptying a shopping cart after purchasing an order. To invalidate cache content, use Application Server Control:

1. On the Application Server Control Administration page, click Invalidation in the Operations region.
2. Select from the following options:
 - **Specified objects:** Select this option to specify the selection criteria. When you select this option, OracleAS Web Cache traverses the contents of the cache to locate the objects to be invalidated. Use this option for invalidation of multiple objects.
 - **All objects:** Select this option to remove all objects from the cache.
 - **The single document that matches this URL:** Select this option to invalidate one object in the cache. Include the complete URL path and file name in the Document to Invalidate field.

Basic Content Invalidation (continued)

3. Specify when to start the invalidation:
 - **Remove objects immediately:** If you select this option, the object is refreshed from the application Web server when the cache receives the next request for it.
 - **Remove objects as soon as possible:** Select this option to mark objects as invalid and then refresh them based on the origin server capacity.
4. Click Preview to view the list of documents to ensure that you are removing only the documents that you want to remove.
5. Click Finish.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Logging Events and Accessing Information

- The OracleAS Web Cache events and errors are stored in an event log.
- The access log contains information about the HTTP requests sent to OracleAS Web Cache.
- You can configure the content of the access log files by defining the fields to appear for each HTTP request event.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Logging Events and Accessing Information

The OracleAS Web Cache events and errors are stored in an event log. The event log can help you determine which documents or objects have been inserted into the cache. It can also identify listening port conflicts or startup and shutdown issues.

The event log has a file name of `event_log` and is stored in the Oracle Home/`webcache/logs` directory.

OracleAS Web Cache generates an access log that contains information about the HTTP requests sent to OracleAS Web Cache. By default, the access log has a file name of `access_log` and is stored in the Oracle_Home/`webcache/logs` directory.

OracleAS Web Cache supports the following log formats:

- Common Log Format (CLF)
- Combined Format
- Web Cache Log Format (WCLF)

Configuring Access Log

Access Logs
Access logs contains information about the HTTP requests sent to Web Cache. You can set up access logging for individual sites, overriding the defaults.

☒ Enable Access Logging
Directory
☒ Buffer in Memory
Takes effect immediately when OK is clicked.

File Name
Time Style
Rollover
Format
☐ Include ESI Fragment Requests

[▶ Show Per-Site Settings](#)

Related Links
[Custom Rollover Schedules](#)
[Custom Access Log Formats](#)
[Rollover Log Files](#)

Cancel

Configuring Access Log

To establish access log configuration settings, perform the following steps:

1. Navigate to the Application Server Control Web Cache Administration page. Click Logging in the Web Cache section. The Logging page appears.
2. Scroll down to the Access Logs region.
3. Set access log settings that are cache specific, as follows:
 - a. Select Enable Access Logging to enable logging, or deselect to disable logging.
 - b. In the Directory field, enter the directory in which to write access logs. The default is \$ORACLE_HOME/webcache/logs.
 - c. Select the Buffer in Memory check box to enable buffered logging or disable buffered logging. With buffered logging, OracleAS Web Cache writes to the access log after the buffer is full. The buffer size is set to 2048 bytes. When the limit is reached, OracleAS Web Cache writes buffered events to the access log file.
 - d. In the File Name field, enter a name for the access log file.
 - e. From the Time Style list, select the time style to use in timestamps for requests in the access log.

Configuring Access Log (continued)

- f. From the Rollover list, select the frequency at which OracleAS Web Cache saves current log information to the access log file and future log information to a new log file.
 - g. From the Format list, select an access log format.
 - h. Select Include ESI Fragment Requests to log the ESI fragment log messages.
4. Click OK.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Configuring Event Log

Logging

Page Refreshed Sep 6, 2005 10:53:37 AM [Cancel] [OK]

Event Log

The event log contains event and error information. It can help you detect port conflicts, find startup or shutdown issues, and determine what objects or objects have been inserted into the cache. Event logging is always enabled.

Directory File Name

Logging Level Time Style

☒ Buffer in Memory Rollover

☒ Include Request Details

Takes effect immediately when OK is clicked.

Takes effect immediately when OK is clicked.

Configuring Event Log

Configuring event log involves two groups of settings: the cache-specific setting and the general information. To configure the event log, perform the following steps:

1. Navigate to the Application Server Control Web Cache Administration page. Click Logging in the Web Cache section. The Logging page appears.
2. Set cache-specific event log settings in the Event Log region:
 - a. In the Directory field, enter the directory in which to write event logs. The default is \$ORACLE_HOME/webcache/logs.
 - b. From the Logging Level list, select the level of detail for the event log. The detail for the event log ranges from minimal to extensive.
 - c. Select Buffer in Memory to enable buffered logging, or deselect to disable buffered logging. With buffered logging, OracleAS Web Cache writes to the event log after the buffer is full. The buffer size is set to 2048 bytes. When the limit is reached, OracleAS Web Cache writes buffered events to the event log file.
 - d. In the File Name field, enter a name for the access log file.
 - e. From the Time Style list, select the time style to use in timestamps for requests in the access log.

Configuring Event Log (continued)

- f. From the Rollover list, select the frequency at which OracleAS Web Cache saves current log information to the access log file, and future log information to a new log file.
 - g. From the Format list, select an access log format.
 - h. Select Include ESI Fragment Requests to log the ESI fragment log messages.
- 3. Click OK.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Event Log with Startup Entries: Example

```
[21/Feb/2005:03:41:19 -0800] [notification 9612] [ecid: -] OracleAS
Web Cache 10g (10.1.2), Build 10.1.2.0.0 041220
[21/Feb/2005:03:41:19 -0800] [notification 9403] [ecid: -] Maximum
number of file/socket descriptors set to 900.
[21/Feb/2005:03:41:19 -0800] [notification 9612] [ecid: -] OracleAS
Web Cache 10g (10.1.2), Build 10.1.2.0.0 041220
[21/Feb/2005:03:41:19 -0800] [notification 9403] [ecid: -] Maximum
number of file/socket descriptors set to 900.
[21/Feb/2005:03:41:19 -0800] [notification 13002] [ecid: -] Maximum
allowed incoming connections are 700
[21/Feb/2005:03:41:19 -0800] [alert 13002] [ecid: -] Failed to
assign port 7777: Address already in use
[21/Feb/2005:03:41:19 -0800] [alert 9607] [ecid: -] Failed to start
the server
[21/Feb/2005:03:41:20 -0800] [alert 12209] [ecid: -] The server
process could not initialize
[21/Feb/2005:03:41:20 -0800] [notification 9604] [ecid: -] The
server is exiting
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Event Log with Startup Entries: Example

The slide shows an excerpt of the event log with port conflict event messages.

During configuration, you configure listening ports from which OracleAS Web Cache receives browser requests. By default, the port is 7778 for HTTP requests. When you start OracleAS Web Cache, a port conflict check is performed. If there is a port conflict, OracleAS Web Cache fails to start and port conflicts are reported to the event log file.

Configuring Rollover Frequency

Rollover frequency enables:

- **Hourly rollover**
- **Rollover at specified times of the day**
- **Users to manually roll over access log file without shutting down OracleAS Web Cache**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Rollover Frequency

The access and event logs can grow and create a contention for disk space over a period of time, depending on the level of activity and the level of details you have specified to be logged. You can configure the frequency of automatic rollover of access logs. This enables you to recycle your access log space at your configured frequency.

You can configure the rollover frequency policy as follows:

1. Select an existing policy, and click Edit Selected to modify an existing rollover policy. Or, click Add to create a new policy. The Edit/Add Access Log Rollover Policy dialog box appears.
2. In the Rollover Policy Name field, enter a unique name for the rollover policy.
3. From the Rollover Frequency list, select how often you want to change the frequency at which OracleAS Web Cache saves current log information to `access_log_file.yyyymmdd`, and writes new log information to the access log file.
4. From the Time Style list, select either LOCAL or GMT to set the time style that you want to associate with a schedule.
5. Select a schedule from the Schedules list. To create a new schedule, select the day, enter the time based on the selected time style, and then click Add Schedule.
6. Click Submit.

Oracle Application Server 10g R2: Administration I 8-51

Manual Rollover of Logs

Rollover Log Files
Select log files to roll over immediately. During the rollover process, Web Page Refreshed Sep 6, 2005 10:57:50 AM
Cache saves current log file to the log_file.yyyymmdd files and writes new log information to the current log.

Event Log
Event Log event_log Rollover Event Log

Access Logs
Rollover Access Log
Select All | Select None
Select Site or Default File Name ▲
☐ Default access_log
Related Link Logging

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Manual Rollover of Logs

In addition to configuring event and access log rollover frequency (as described earlier), you can also use Application Server Control to manually initiate the rollover of event and access logs. During the rollover process, OracleAS Web Cache saves current log file to the log_file.yyyymmdd file and writes new log information to the event_log file.

To immediately roll over log files, perform the following steps:

1. Navigate to the Application Server Control Web Cache Administration page.
2. Click Rollover Event Log Files in the Operations section.
3. To roll over event log files from the Event Logs table, select an individual cache or click Select All to select all the caches. Click Rollover Event Log in the Event Log section.
4. To roll over access log files, from the Access Logs table, select an access log for a configured site, or click Select All to select all the caches. Click Rollover Access Log.
5. Click Submit.

Summary

In this lesson, you should have learned how to:

- Start, stop, and restart OracleAS Web Cache
- Change passwords for administrative users and listener ports
- Specify site-to-server mappings
- Create and configure caching rules
- Set up basic invalidation mechanism
- Set up expiration rules
- Configure access and event logs
- Obtain basic performance statistics

ORACLE

Copyright © 2005, Oracle. All rights reserved.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

9

Configuring and Managing OC4J

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Create OC4J instances**
- **Start and stop OC4J instances**
- **Enable or disable application startup**
- **Configure OC4J instance properties**
- **Configure the Web site and JSP properties**
- **Edit OC4J configuration files**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Java 2 Platform, Enterprise Edition (J2EE): Overview

The Java 2 Platform, Enterprise Edition (J2EE) is a standard for developing and implementing enterprisewide applications:

- **It provides multitier applications support.**
- **It is designed to help improve the process of developing, deploying, and implementing enterprisewide applications.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

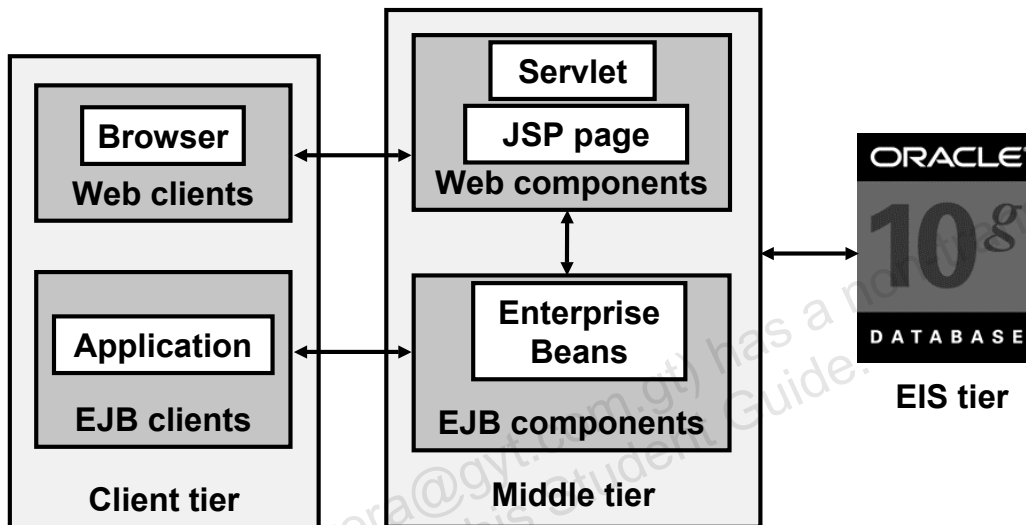
Java 2 Platform, Enterprise Edition (J2EE): Overview

Java 2 Platform, Enterprise Edition (J2EE) is a complete application architecture for enterprise class applications. J2EE was first proposed by Sun Microsystems and has been included in the Java Community Process to make it a part of the open systems movement.

J2EE emphasizes a portable, component-based approach to the creation, deployment, and management of complex applications. J2EE supports components on four tiers: client, Web, business, and the Enterprise Information System (EIS).

J2EE Platform

- Is a multitiered, distributed application model
- Supports component-based J2EE applications



Copyright © 2005, Oracle. All rights reserved.

J2EE Platform

J2EE defines a platform for developing, deploying, and executing applications in a multitiered, distributed application model. That is, the application logic of a J2EE application can be divided into components based on their functions and distributed to the appropriate tier on the multitiered architecture.

The slide shows a standard multitiered J2EE application model where the application logic is distributed into the following tiers:

- Client-tier components, such as a Web browser, run on the client machine.
- Presentation logic is built with Web-tier components such as JavaServer Pages (JSP) and Java servlets that run on the J2EE server.
- Server-side business logic is distributed as business-tier components that run on the J2EE server. Enterprise JavaBeans (EJB) and the J2EE framework Oracle ADF Business components (BC) are examples of business-tier components.
- Data is stored in the Enterprise Information System (EIS) tier that runs on the database server, such as Oracle Database 10g.

Note: This slide shows servlets and EJBs on a single machine. However, the EJB container can also be on a separate machine (process) from the servlet container.

The J2EE application components are developed in the Java programming language. When the J2EE platform is used, the J2EE components are assembled into a J2EE application, verified according to the J2EE specification, and deployed to the J2EE server. The deployed applications are run and managed by the J2EE server.

Client-Tier Components

- **A Web browser:**
 - Is used for a Web-based J2EE application
 - Downloads static or dynamic Web pages from Web-tier components
 - Is a thin client
- **An application client:**
 - Is used for a non-browser-based J2EE application
 - Executes on the client machine
 - Contains a graphical or command-line interface
 - Is a thick client
 - Accesses business-tier components or a servlet on the Web tier

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Client-Tier Components

Web Browser Client

For a Web-based J2EE application, the user's Web browser is the client component. The Web browser downloads static or dynamic Web pages from the Web tier to a client machine.

Dynamic Web pages are generated by servlets and JSPs from the Web tier. JSPs do not require Java plug-ins or security policy files to be downloaded to the client machine.

A downloaded Web page can contain an embedded applet that executes in the Java Virtual Machine (JVM) in the Web browser. An applet may require a Java plug-in and a security file to successfully execute in the user's Web browser.

Web-based clients are typically thin clients, which do not perform operations such as executing complex business rules, connecting to the database, and querying the database. These operations are generally performed by business-tier components.

Application Client

An application client executes on a client machine for a non-browser-based J2EE application. It contains a graphical user interface (GUI) created from Swing, Abstract Window Toolkit (AWT) APIs, or a command-line interface. Application clients can directly access business-tier components. They can access a servlet that is running in the Web tier through an HTTP connection. Application client can also access servlets and EJBs on the server tier.

Oracle Application Server 10g R2: Administration I 9-5

Web-Tier Components

- **A Web tier may consist of:**
 - Java servlets
 - JSPs
- **Servlets and JSPs:**
 - Work on a request-response model
 - Generate HTML dynamically
 - Access the database through JDBC
 - Access the business-tier components
 - Handle user-centric events, such as an HREF link or form submission
 - Usually generate visual interfaces, such as a Web page

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Web-Tier Components

J2EE Web-tier components can be either servlets or JSPs that can statically and dynamically generate HTML, Wireless Markup Language (WML), or Extensible Markup Language (XML) pages. Java servlets provide a simple yet powerful API for generating Web pages by dynamically processing the client requests and constructing responses. JSPs simplify the process of dynamically generating the content, by allowing Java as the scripting language inside HTML pages. JSPs are translated and compiled into Java servlets, which are then run in a Web server like any other servlet.

Web components can access the business-tier components that access the database. Web components can handle requests from the client, such as form submission. Some advantages of using Web components are listed below:

- The HTML interface does not require prior installation on the client machines, whereas the conventional clients require redeployment of the applications on the client machines.
- The HTTP protocol, over which the client's request for Web pages are served, can pass through most firewalls.

Thus, a combination of Web components and EJBs enables:

- Presentation logic for generating Web pages through Web components
- Transactional business logic through EJBs

Business-Tier Components

Business-tier components:

- **Are EJBs**
- **Handle business logic**
- **Receive data from client programs**
- **Retrieve data from database storage**
- **Process the data and communicate with the database and the client program**
- **Can be invoked by the Web-tier components**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Business-Tier Components

EJBs are the components that run in the business tier. These distributed components contain business logic that meets the needs of a particular business domain, such as banking, order entry system, or human resources management. The business-tier components can receive data from client programs, process the data, and send the processed data to the database server for storage. The business-tier components can also retrieve data from the database, process it, and send it back to the client program.

The Web-tier components may invoke the business-tier components where the business logic is handled. For example, a servlet may invoke an enterprise bean to insert new customer details and return any processed data back to the client.

Enterprise JavaBeans (EJB)

EJBs:

- **Are server-side components written in Java**
- **Contain the business logic of an enterprise application**
- **Are hosted in EJB containers**
- **Are platform independent**
- **Provide remote services for clients**
- **Can be exposed as Web services**
- **Use JDBC to connect to a database**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Enterprise JavaBeans (EJB)

Enterprise JavaBeans (EJB) is an architecture for developing transactional applications as distributed components in Java. EJB is a powerful development methodology for distributed application development. When applications are developed with enterprise beans, neither the bean developer nor the client application programmer needs to be concerned with details such as transaction support, security, remote object access, and many other complicated and error-prone issues. These are provided transparently for the developer by the EJB container.

EJBs offer portability. A bean that is developed on one EJB container runs on other EJB containers that meet the EJB specification. Oracle Application Server 10g implements the EJB specification by providing a server and a container that hosts the Enterprise JavaBeans.

EJBs are accessed using Java's RMI framework, which can be implemented with different network protocols. Oracle Application Server 10g provides the RMI/Internet Inter-ORB Protocol (IIOP) as well as a platform-specific, optimized protocol called Oracle Remote Method Invocation (ORMI).

Additionally, a stateless session EJB can be exposed as a Web service. Web services are a standard for building interoperable, distributed applications that are platform- and programming-language independent.

Comparing JAR, WAR, and EAR Files

You can package components in the J2EE application into:

- **Java Application Archives (JAR):** JavaBeans and Enterprise JavaBeans
- **Web Archives (WAR):** HTML documents, servlets, JSPs, applet class files
- **Enterprise Archives (EAR):** HTML files, Java class files, XML files

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Comparing JAR, WAR, and EAR Files

To deploy EJBs and other components in the J2EE application, you must package all the components together. This also includes JSP files, images, utility classes, and other files that are part of this package.

JAR Files

Java provides a utility for creating archives, called Java Application Archives (JAR). In addition to using JAR files for archiving and distribution, you can also use them for deployment and encapsulation of libraries, components, plug-ins, and other files (such as image files). The JAR file maintains the file subdirectories, and special files in the JAR, such as manifests and deployment descriptors, instruct how the JAR is to be treated. You can package standard JavaBeans and Enterprise JavaBeans or an entire application into JAR files that can be executed by the JVM.

WAR Files

You add Web components to a J2EE application in a package called a Web Archive (WAR) file. WAR files are similar to JAR files but contain a .war extension. You can include HTML documents, servlets, JSPs, and applet class files into WAR files. A WAR file has a specific hierarchical directory structure. The top-level directory of a WAR is the document-root directory of the application. JSP pages, client-side classes and archives, and static Web resources are stored in the document-root directory.

Comparing JAR, WAR, and EAR Files (continued)

WAR Files (continued)

The document-root directory contains a subdirectory called WEB-INF, which contains the following files and directories:

- Tag library descriptor files
- classes: Directory that contains server-side classes: servlet, utility classes, and JavaBeans components
- web.xml: The Web application deployment descriptor
- lib: Directory that contains JAR archives of libraries

EAR Files

An Enterprise Archive file or EAR is a JAR file that contains Web modules of a J2EE application. A Web module is an entity consisting of one or more resources such as HTML files, Java class files, and XML files. In other words, an EAR file is a JAR file that can contain JAR and WAR files in addition to other files, and ends with the .ear extension. An EAR file also contains an application descriptor called application.xml that describes its contents.

Oracle Application Server Containers for J2EE (OC4J)

- **OC4J is the J2EE server implementation in Oracle Application Server 10g.**
- **Key features:**
 - **Implements J2EE 1.3 Specification**
 - **Runs on standard JVM**
 - **Provides high performance and scalability**
 - **Is productive for developers to use**
 - **Is simple to manage and deploy**
 - **Provides clustering for high availability and failover**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Containers for J2EE (OC4J)

Oracle Application Server 10g provides a complete J2EE server, called Oracle Application Server Containers for J2EE (OC4J). It is written in Java and executes on the standard Java Virtual Machine (JVM). OC4J is installed in three ways:

- It is included in Oracle Application Server 10g.
- It is included in Oracle JDeveloper 10g.
- It is available as a stand-alone version (that is, without any other software).

Managing OC4J

You can manage OC4J using:

- **Application Server Control:**
 - Recommended management tool for any Oracle Application Server installation
 - Graphical interface to manage OC4J components and clusters, and to deploy applications
- **Command-line utilities:**
 - `opmnctl`: Starts and stops an OC4J instance
 - `dcmctl`: Creates an OC4J instance and deploys applications

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing OC4J

As with all components of Oracle Application Server, OC4J can also be managed in two distinct ways:

- Using the graphical user interface of Application Server Control, you can interactively manage the OC4J instances. This is particularly useful for monitoring and troubleshooting OC4J instances.
- Using command-line interfaces, such as DCMCTL and OPMNCTL, you can manage the OC4J instances in a noninteractive environment, that is, when performing batch processing or scheduled/automated maintenance operations.

Creating an OC4J Instance

System Components

Start Stop Restart Delete OC4J Instance Enable/Disable Components Configure Component Create OC4J Instance

Select All Select None

Select Name	Status	Start Time	CPU Usage (%)	Memory Usage (MB)
<input type="checkbox"/> home	↑	Sep 6, 2005 4:41:08 AM	0.00	25.62
<input type="checkbox"/> HTTP Server	↑	Sep 6, 2005 4:41:05 AM	0.78	96.60
<input type="checkbox"/> OC4J Portal	↑	Sep 6, 2005 4:41:08 AM	0.00	48.07
<input type="checkbox"/> Portal:portal	↑	N/A	N/A	N/A
<input type="checkbox"/> Web Cache	↑	Sep 6, 2005 4:41:04 AM	0.00	13.02
<input type="checkbox"/> Management	↑	Sep 6, 2005 4:43:10 AM	0.60	90.05

Farm > Application Server: portal.edrsr16p1 >

Create OC4J Instance

Cancel Create

Enter the name of the OC4J instance you wish to create.

* OC4J Instance Name OC4J_Temp

Cancel Create

Confirmation

OC4J instance "OC4J_Temp" was created.

OK

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Creating an OC4J Instance

A default OC4J instance is installed with the name home. You can create additional instances, each with a unique name within an application server instance:

1. In Application Server Control, navigate to the application server instance where you want to create the new OC4J instance.
2. Click Create OC4J Instance. This displays a page that requests a name for the new instance.
3. Provide a name in the field.
4. Click Create.
5. A Confirmation page appears indicating that the OC4J instance was created.

A new OC4J instance is created with the name you provided. This OC4J instance shows up on the application server instance page in the System Components section. The newly created OC4J instance is not started up; you should start it before you can deploy applications to it.

When you create a new OC4J instance, you also create a directory of the same name in the \$ORACLE_HOME/j2ee/ directory. In the example in the slide, the OC4J_Temp directory is created in the /home/oracle/portal/j2ee/ directory.

Creating an OC4J Instance Using dcmctl

You may also need to create an OC4J instance in a noninteractive way. In such cases, you can use the `dcmctl` command-line utility to create an OC4J instance. You can use the following command to create an OC4J instance with the name `OC4J_Temp`:

```
dcmctl createComponent -ct oc4j -co OC4J_Temp
```

You can verify whether the OC4J instance is created by using the `dcmctl listcomponents` command:

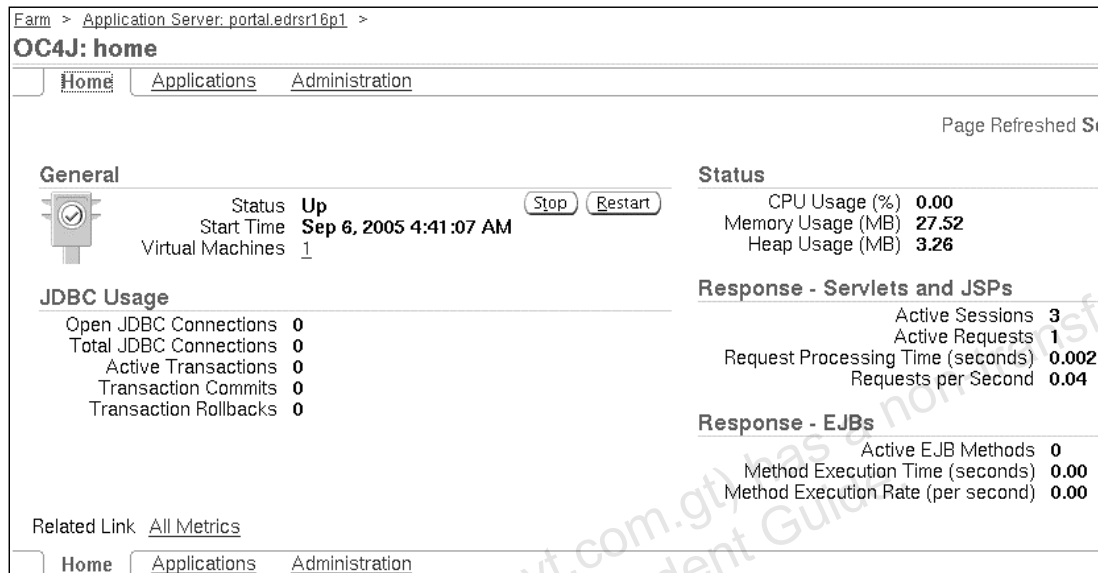
```
$ dcmctl listcomponents
1
Component Name: home
Component Type: OC4J
Instance:      portal.edrsr16p1.us.oracle.com

2
Component Name: HTTP_Server
Component Type: HTTP_Server
Instance:      portal.edrsr16p1.us.oracle.com
...
...
...
$
```

You can also check whether the `OC4J_Temp` directory has been created under the `.../j2ee` directory:

```
$ ls ../../j2ee
deploy.ini      home          j2eetargets.xml
OC4J_Demos     oc4j_opmn.xml OC4J_Portal    OC4J_Temp
properties
$
```


Application Server Control: OC4J Home Page



Application Server Control: OC4J Home Page

The OC4J home page provides a single view of the instance and provides for administration of the various elements in the J2EE application environment. Use the OC4J home page of Application Server Control to:

- Configure the OC4J instance
- Administer services and resources, such as data sources and security
- Monitor the availability, usage, and performance of the server and applications

The OC4J home page has the following sections:

- **General:** This section provides a snapshot of the current status of the OC4J server, and enables you to stop, start, or restart the OC4J server.
- **Status:** The Status section provides a quick view of the performance of the server, CPU Usage, Memory Usage, and Heap Usage.
- **JDBC Usage:** This section provides the number of open JDBC connections, active transactions, number of commits, and rollbacks.
- **Response - Servlets and JSPs:** This section provides details about the active sessions, active requests, average time for processing requests, and the number of requests processed per second.
- **Response - EJBs:** The Response - EJBs section provides transactional details about Enterprise JavaBeans.


Starting and Stopping OC4J Instance

Farm > Application Server: portal.edrsr16p1 >

OC4J: OC4J_Temp

Home Applications Administration

Page Refreshed Sep 8, 2005 3:58:15 AM

General		Status
	Status Down	Start
	Start Time Unavailable	CPU Usage (%) Unavailable
	Virtual Machines Unavailable	Memory Usage (MB) Unavailable
		Heap Usage (MB) Unavailable

 **Confirmation**

"OC4J_Temp" has been started.


OK

Farm > Application Server: portal.edrsr16p1 >

OC4J: OC4J_Temp

Home Applications Administration

Page Refreshed Sep 8, 2005 4:03:43 AM

General		Status
	Status Up	Stop Restart
	Start Time Sep 8, 2005 4:03:43 AM	CPU Usage (%) 0.00
	Virtual Machines 1	Memory Usage (MB) 56.48
		Heap Usage (MB) 5.06

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting and Stopping OC4J Instance

You can start, stop, and restart the OC4J instance by using the OracleAS Instance home page or the OC4J instance home page.

Using the OracleAS instance home page, select the OC4J instance (check box) in the components table, and click Start (at the top of the table).

You can also use the OC4J home page to start OC4J instances:

1. Click Start in the General Information section on this page.
2. Click OK on the Confirmation page.
3. The OC4J instance is displayed with the status Up.

Starting and Stopping OC4J Instances Using OPMN

- You can use the `opmnctl` utility to start and stop all configured OC4J instances from the command line.
- To start and stop the `OC4J_Temp` instance:

```
$> opmnctl startproc process-type=OC4J_Temp  
$> opmnctl stopproc process-type=OC4J_Temp
```

- To start and stop all OC4J instances:

```
$> opmnctl startproc ias-component=OC4J  
$> opmnctl stopproc ias-component=OC4J
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Stopping and Starting OC4J Instances Using OPMN

To stop a specific OC4J instance, for example, `OC4J_Temp`, use the command:

```
$> opmnctl stopproc process-type=OC4J_Temp
```

To start a specific OC4J instance, for example, `OC4J_Temp`, use the command:

```
$> opmnctl startproc process-type=OC4J_Temp
```

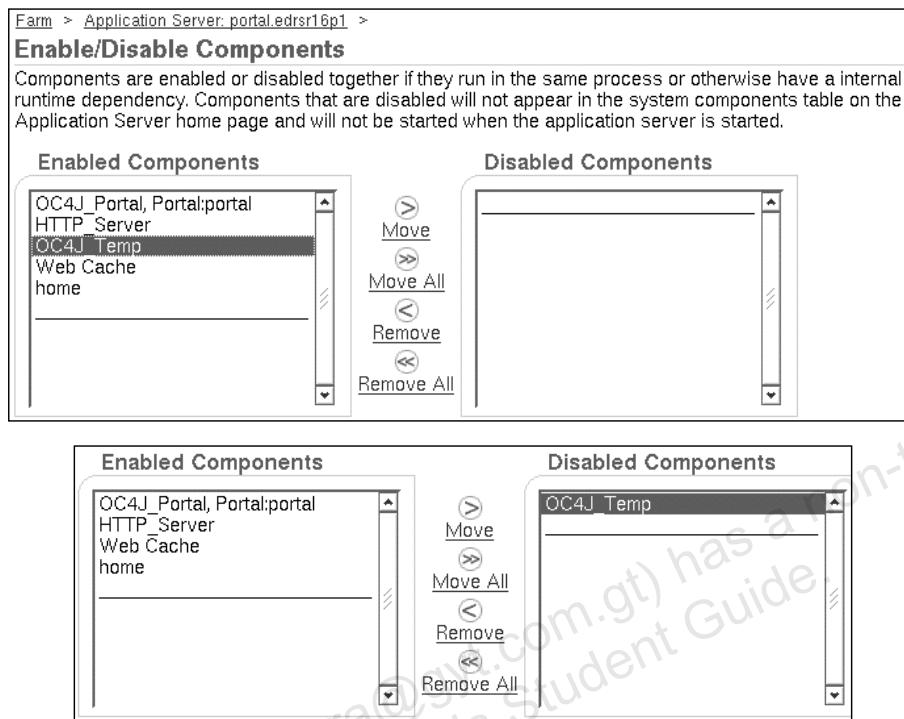
To stop all OC4J instances that are configured in the OracleAS Instance, use the command:

```
$> opmnctl stopproc ias-component=OC4J
```

To start all OC4J instances belonging to the OracleAS Instance, use the command:

```
$> opmnctl startproc ias-component=OC4J
```

Disabling OC4J Instances



Copyright © 2005, Oracle. All rights reserved.

Disabling OC4J Instances

You can disable or enable installed components using Application Server Control. For example, on a production system, you may not want some components, such as OC4J_Temp, to be running. You can disable unwanted components. Such components, even though they are installed, cannot be started with the OracleAS instance. This can save your system resources.

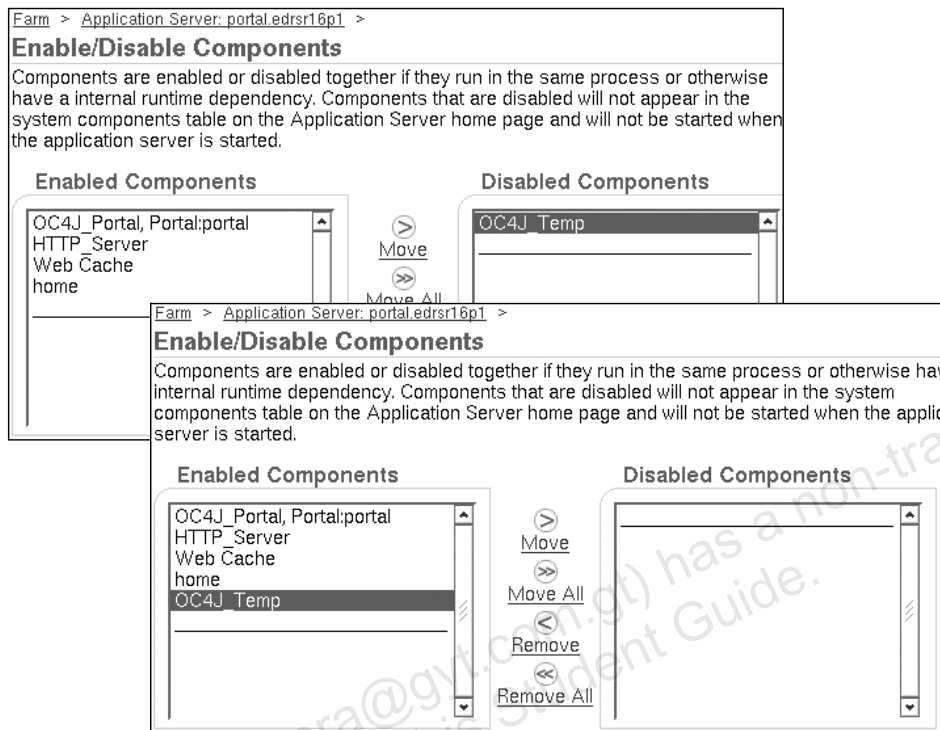
To disable a component, click the Enable/Disable Components link on the OracleAS instance page. The Enable/Disable Components page is displayed.

1. Select the component that you want to disable from the Enabled Components list.
2. Click Move (>). The selected component appears under Disabled Components.
3. Click OK. A confirmation page is displayed, on which you confirm disabling the component.

The disabled component no longer appears in the System Components table on the OracleAS instance page.

All the subcomponents (applications deployed to the OC4J instance) of the disabled component are stopped and disabled.

Enabling OC4J Instances



Copyright © 2005, Oracle. All rights reserved.

Enabling OC4J Instances

You can enable a previously disabled component from the OracleAS Instance home page. Click the Enable/Disable Components link to invoke the Enable/Disable Components page, and perform the following steps:

1. Select the component you want to enable from the Disabled Components list.
2. Click Remove. The selected component appears on the list of Enabled Components.
3. Click OK.

The enabled component appears in the System Components table of the OracleAS Instance home page. The component is not started. You have to start the component.

OC4J Configuration Basics

OC4J has three groups of configuration files:

- **The `mod_oc4j` configuration files are:**
 - Used to administer the `mod_oc4j` module of Oracle HTTP Server
 - In `$ORACLE_HOME/Apache/Apache/conf`
- **OC4J server configuration files are:**
 - Used to administer the OC4J server
 - In `$ORACLE_HOME/j2ee/<instance>/config`
- **Two types of OC4J application configuration files:**
 - **J2EE standard:** Stored in `/applications/<app-name>`
 - **OC4J specific:** Stored in `/application-deployments/<app-name>`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OC4J Configuration Basics

Each OC4J instance can contain multiple J2EE applications. The access to Web applications for HTTP clients is provided using `mod_oc4j` between Oracle HTTP Server and OC4J.

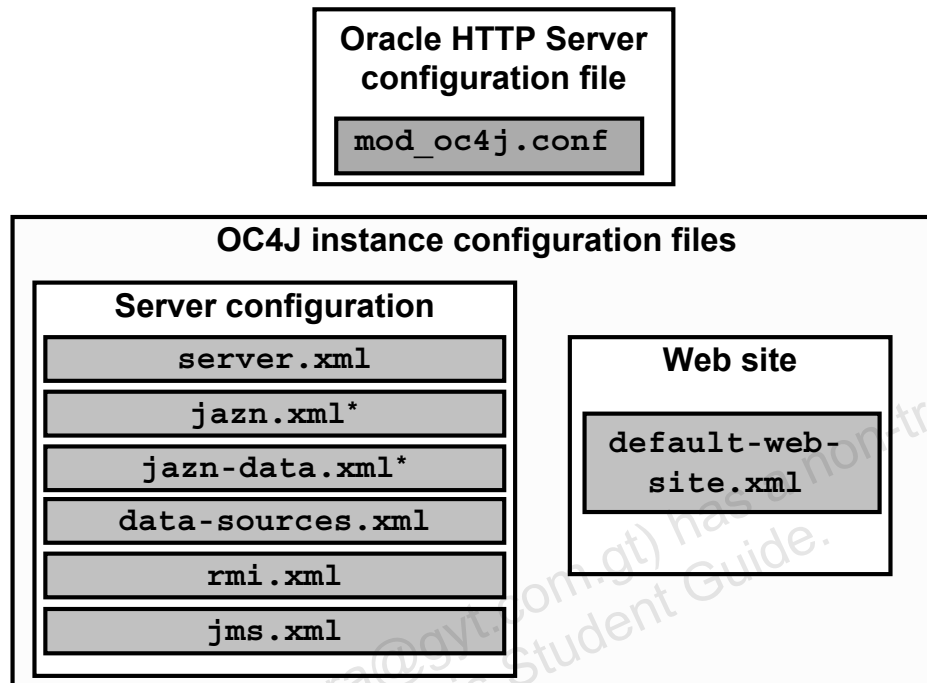
The relationship of each of these components is described within the OC4J configuration files.

The application configuration files describe the application deployment and client configuration.

- OC4J server configuration files are OC4J specific and configure the OC4J server and point to the location of key J2EE configuration files.
- J2EE application configuration files for J2EE applications and clients are J2EE application specific and are used for the deployment of J2EE applications.

Each application is a standard J2EE application defined in an EAR file. An application can have both Web application components, such as servlets and JSP pages, and EJB applications. J2EE applications with a Web application are made accessible to Web clients by binding them to a URL. Applications that contain only EJBs are not bound to a URL in a Web site, but are accessible in the server through Remote Method Invocation (RMI) or locally using the same server-level JVM.

OC4J Instance Configuration Files



ORACLE

Copyright © 2005, Oracle. All rights reserved.

OC4J Instance Configuration Files

The OC4J instance is configured by using a few configuration files. These files are standard for J2EE servers and provide a way of integrating components with the OC4J framework. There is no need to modify the configuration files that are contained in JARs, WARs, and EARs while they are being deployed.

There is an implied hierarchy to these configuration files:

```
server.xml
|----->rmi.xml
|----->jms.xml
|----->application.xml
|           |----->principals.xml
|           `----->data-sources.xml
|----->global-web-application.xml
`----->default-web-site.xml
           |----->default-web-app
           `----->web-app
```

The `jazn.xml` and `jazn-data.xml` files describe the security configuration by using the Java Authentication and Authorization Service (JAAS). If JAAS is not used, these files need not be configured.

Relationship of Configuration Files

When an application is deployed, an entry is made in the `/config/server.xml` file:

```
<application name="app01"
  path="../../../applications/app01.ear" />
```

The context root for this entry is defined in `/config/default-web-site.xml`:

```
<web-app application="app01" name="app01"
  root="/app01"/>
```

The modules of the application are defined in `/applications/app01/META-INF/application.xml`:

```
<web><web-uri>webapp1.war</web-uri></web>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Relationship of Configuration Files

The `server.xml`, `default-web-site.xml`, and `application.xml` files work together to define the configuration for an application. If an application named “app01” is deployed by using an `app01.ear` file (and `app01.ear` contains `webapp1.war`), then the entries in the corresponding files (as shown in the slide above) are created during deployment.

In the `server.xml` file, each existing application contains a line that provides details about the application, such as name and path.

In the `default-web-site.xml` file, a `<web-app>` entry exists for each Web application that is bound to the Web site upon OC4J startup. The application attribute is the name provided in `server.xml` as the application name. The name attribute is the name of the WAR file, without the `.war` extension. The root attribute defines the root context for the application.

`application.xml` contains the standard J2EE application descriptor for the application.

Sample server.xml File

```
<application-server
  localhostIsAdmin="true"
  application-directory="../applications"
  deployment-directory="../application-deployments"
  connector-directory="../connectors">

  <rmi-config path="../rmi.xml" />
  <jms-config path="../jms.xml" />
  <log><file path="../log/server.log" /></log>

  <global-application name="default"
    path="application.xml" />

  <global-web-app-config
    path="global-web-application.xml" />
  <web-site path="../default-web-site.xml" />

  <application name="app01"
    path="../applications/app01.ear" />
  ...
</application-server>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Sample server.xml File

This is an example of what the server.xml file looks like. Note the following aspects of this file:

- All directories are relative to the config directory, which makes maintenance much easier.
- The application-directory attribute specifies a directory to store applications (EAR files). If none is specified (the default), OC4J stores the information in j2ee/home/applications.
- The deployment-directory attribute identifies where the OC4J-specific generated files will be persistently stored. Each deployed application has a corresponding deployment-directory attribute. A directory is created for each deployed application in which the generated files will be stored.
- The application-auto-directory attribute specifies that files placed in this directory are automatically deployed without any further action necessary from an administrator. This is mainly a developer-oriented functionality and is disabled in Oracle Application Server.
- The application tag defines the name and the path to the application archive; in this case, app01 and ../applications/app01.ear.

Sample default-web-site.xml File

```
<web-site port="3301" protocol="ajp13"
  display-name="Default OC4J Web Site">
  <default-web-app application="default"
    name="defaultWebApp" root="/j2ee"/>
  <web-app application="default" name="dms"
    root="/dmsoc4j"/>
  <web-app application="app01" name="app01"
    root="/app01"/>
  <web-app application="BC4J" name="webapp"
    root="/webapp"/>
  <access-log
    path="../../log/default-web-access.log"/>

</web-site>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Sample default-web-site.xml File

The default-web-site.xml file contains the configuration for the default Web site. The <web-site ...> tag contains the configuration for a Web site. You can specify the following:

- **port:** The port of this instance; default is 3000
- **protocol:** The protocol to use for communication between this OC4J instance and mod_oc4j
- **display-name:** The user-friendly name shown when administrating this site
- **default-web-app:** The tag that identifies which application is displayed if no application name is requested. This is the application that is bound to the root of the site.
- **web-app:** Binds a Web module from a J2EE application to a virtual path
- **access-log:** Specifies the path to the access log file

For additional information about OC4J configuration files, refer to the *Oracle Application Server Containers for J2EE User's Guide 10g Release 2 (10.1.2)*.

Configuring OC4J Using Application Server Control

Click the **Administration** link on the OC4J home page to access the OC4J Administration page.



Copyright © 2005, Oracle. All rights reserved.

OC4J Administration Page

The OC4J Administration page provides access to basic and advanced configuration functions, and provides links to other pages for more detailed operations, such as setting the directories of the default application and configuration file paths.

The OC4J server configuration is defined in a set of XML files that specify the properties of the server and other entities such as data sources. The OC4J home page contains administration links to property pages for editing these configuration files and adding new services.

For example, you can use the Server Properties page to edit settings and properties, such as default application settings, and RMI and JMS configuration file paths.

You can also edit the server configuration file directly from the interface using the Advanced Properties page for setting properties not presented through the interface. The ports such as RMI and JMS are controlled by OPMN.

Server Properties Page: General Section

Click the **Server Properties** link on the **OC4J Administration** page to access **Server Properties**.

Farm > Application Server: portal.edrsr16p1 > OC4J: home >

Server Properties

Page Refreshed Sep 8, 2005 4:31:02 AM

General	
Name	home
Server Root	/home/oracle/portal/j2ee/home/config
Configuration File	/home/oracle/portal/j2ee/home/config/server.xml
Default Application Name	default
Default Application Path	application.xml
Default Web Module Properties	global-web-application.xml
Application Directory	./applications
Deployment Directory	./application-deployments

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Server Properties Page: General Section

The OC4J Administration page of Application Server Control is the interface for configuring OC4J instances.

Use the Server Properties page to view or edit the properties for the current OC4J container. The Server Properties page contains two sections: General and Multiple VM.

The General section contains the following fields:

- **Name, Server Root, Configuration File, Default Application Name, and Default Application Path:** These fields are set at the time of creation of the OC4J instance and cannot be changed.
- **Default Web Module Properties:** This field specifies the location of a file that defines the properties that are applicable to all Web modules. By default, this points to the `global-web-applications.xml` file.
- **Application Directory:** The default directory to place the master EAR file of the deployed application is the `/applications` directory. You can change this location of the default directory in this field. The directory is relative to `j2ee/home/config`.
- **Deployment Directory:** The default directory to place the modified module deployment descriptors with added defaults is the `/application-deployments` directory. You can change this location of the default directory in the Deployment Directory field. The directory is relative to `j2ee/home/config`.

Web Site Properties

Farm > Application Server: portal.edrsr16p1 > OC4J: home >

Website Properties

Page Refreshed Sep 8, 2005 4:32:05 AM

Default Web Module

Name **defaultWebApp**
Application **default**
Load on startup **true**

URL Mappings for Web Modules

Name	Application	URL Mapping	Load on startup
dms	default	/dmsoc4j	<input type="checkbox"/>
IsWebCacheWorkingWeb	IsWebCacheWorking	/IsWebCacheWorking	<input checked="" type="checkbox"/>
webapp	BC4J	/webapp	<input checked="" type="checkbox"/>
wsrp-samples	portletapp	/portletapp	<input checked="" type="checkbox"/>

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Web Site Properties

Use this page to change the default Web application and its parent. You can specify whether one or all Web applications are to be loaded upon startup. These parameters are stored in the `default-web-site.xml` file. The page contains two major sections: Default Web Module and URL Mappings for Web Modules.

The Default Web Module section contains the following nonconfigurable fields:

- **Name:** Displays the desired Web application name
- **Parent Application:** The name of the J2EE application, as specified by the application attribute of an `<application>` element in the `server.xml` file
- **Load on startup:** Specifies whether this Web application is loaded when the OC4J instance is started

The URL Mappings for Web Modules table lists all current Web modules contained within the OC4J container, and lists the following information about each module:

- **Application:** The name of the application to which the Web module belongs
- **URL Mapping:** URL to which this Web module is bound
- **Load on startup:** Specifies whether this Web module is loaded when the OC4J instance is started

JSP Properties

Farm > Application Server: portal.edrsr16p1 > OC4J: home > Web Module: Global Web Module >

JSP Properties: jsp

Page Refreshed Sep 8, 2005 4:33:55 AM

Oracle JSP Container Properties

The following properties may be used to configure the Oracle JSP Container.

Debug Mode	No	Emit Debug Info	No
External Resource for Static Content	No	When a JSP Changes	Recompile JSP
Generate Static Text as Bytes	No	Precompile Check	No
Tags Reuse Default	No	Validate XML	No
Reduce Code Size for Custom Tags	No		

SQLJ Command

Alternate Java Compiler

Revert Apply

JSP Properties

Use the JSP Container Properties page to configure all JSPs deployed in the current OC4J instance. These properties can be included in the `global-web-application.xml` file within the `<servlet>` element:

- **Debug Mode:** Set Debug Mode to True to print the stack trace when a run-time exception occurs. The default is True.
- **External Resource for Static Content:** Set this field to True to place all static content of the page into a separate Java resource file during translation. The default is False.
- **Generate Static Text as Bytes:** Set this field to True to instruct the JSP translator to generate static text in JSP pages as characters instead of bytes. The default is False.
- **Tags Reuse Default:** This specifies a default setting for JSP tag handler pooling (True to enable by default; False to disable by default). You can override this default setting for any particular JSP page. The default is True.
- **Reduce Code Size for Custom Tags:** Set this field to True for further reduction in the size of generated code for custom tag usage. The default is False.
- **Emit Debug Info:** Set this field to True in a development environment to generate a line map to the original `.jsp` file for debugging. The default is False.

JSP Properties (continued)

- **When a JSP Changes:** This determines whether classes are automatically reloaded or JSP pages are automatically recompiled, in case of changes. Possible settings are Do Nothing, Reload Classes, and Recompile JSP. The default is Recompile JSP.
- **Precompile Check:** Set this Boolean to True to check the HTTP request for a standard `jsp_precompile` setting. The default is False.
- **Validate XML:** This specifies whether XML validation is performed on the `web.xml` file and TLD files. The default is False.
- **SQLJ Command:** Use this if you want to specify a SQLJ command line, or if you want to specify an alternative SQLJ translator, optionally with command-line settings (for development). If you specify an alternative translator, it will be spawned in a separate JVM. A null setting means that you use the Oracle SQLJ version provided with Oracle Application Server, with its default option settings.
- **Alternate Java Compiler:** Use this if you want to specify a javac command line, or if you want to specify an alternative Java compiler, optionally with command-line settings (for development). If you specify an alternative compiler, it will be spawned in a separate JVM (javac runs in the same JVM). A null setting means that you should use the JDK javac with default settings.

Advanced Properties

Farm > Application Server: portal.edrsr16p1 > OC4J: home >

Advanced Server Properties

Page Refreshed Sep 8, 2005 4:35:47 AM

Configuration Files

File Name	Location
jms.xml	/home/oracle/portal/j2ee/home/config
server.xml	
rmi.xml	
global-web-application.xml	
default-web-site.xml	

Edit server.xml

This configuration file is located at /home/oracle/portal/j2ee/home/config/server.xml

```
<?xml version="1.0"?>
<!DOCTYPE application-server PUBLIC "-//Oracle//DTD OC4J Application-server 9.04//EN"
"http://xmlns.oracle.com/ias/dtds/application-server-9_04.dtd">

<application-server localhostAdmin="true"
application-directory="./applications"
deployment-directory="./application-deployments"
connector-directory="./connectors"
>
  <rmi-config path="/rmi.xml" />
  <sep-config path="/internal-settings.xml" />
  <jms-config path="/jms.xml" />
  <javacache-config path="./../javacache/admin/javacache.xml" />
  <j2ee-logging-config path="/j2ee-logging.xml" />
</log>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Advanced Server Properties

Use the Advanced Server Properties page to view a list of configuration files for the current OC4J server. You can edit any of the files in the list by clicking the file name in the table.

The Advanced Server Properties page lists a table of all configuration files and includes the file name and location of each of the files. Within this section, you can configure the XML files for the current OC4J instance for `server.xml`, `global-web-application.xml`, `rmi.xml`, `jms.xml`, and `default-web-site.xml`.

When you access the Advanced Properties page from an application home page, you can edit the following OC4J Configuration files:

- For the default application, `application.xml`, `oc4j-connectors.xml`, `principals.xml`, and `data-sources.xml` can be edited.
- For a deployed application, `principals.xml` and `orion-application.xml` can be edited.

You can modify `data-sources.xml` and `principals.xml` at both the global and local levels. To modify the global definitions, modify them under the default application. To modify them locally in an application, modify them under the designated application.

Application Deployment

- **Deploying applications to Oracle Application Server is simple.**
- **The deployer configures the OC4J instance with applications:**
 - **Web, EJB, and J2EE applications**
 - **Uses manual or automatic deployment methods**
- **The OC4J instance verifies and deploys the applications:**
 - **It automatically deploys and redeploys new applications.**
 - **It generates the required OC4J-specific application files.**
 - **The OC4J instance should be restarted.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Application Deployment

The deployment of a J2EE application can be performed by using Application Server Control or by using the `dcmctl` command-line utility. You can also deploy applications manually by:

- Modifying the configuration files
- Unpacking the deployment archive file in the deployment directory

Using Application Server Control to deploy a J2EE application, you are expertly guided through deployment via a Deploy Application Wizard. You can access this wizard by navigating from the OC4J home page to the Applications property page and by using the Deploy EAR or WAR file buttons.

OC4J Applications Page

Farm > Application Server: portal.edrsr18p1 >

OC4J: home

Home Applications Administration

Page Refreshed Sep 8, 2005 4:38:04 AM

Default Application Name default
Default Application Path application.xml

Deployed Applications

Deploy EAR file Deploy WAR file

Edit Undeploy Redeploy

Select	Name	Path	Parent Application	Active Requests	Request Processing Time (seconds)	Active EJB Methods
<input type="checkbox"/>	ADFBCManager	../applications/ADFBCManager.ear	default	0	0.00	0
<input type="checkbox"/>	BC4J	../applications/BC4J.ear	default	0	0.00	0
<input type="checkbox"/>	IsWebCacheWorking	../applications/IsWebCacheWorking.ear	default	0	0.00	0
<input type="checkbox"/>	portletapp	../applications/portletapp.ear	default	0	0.00	0

Home Applications Administration

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OC4J Applications Page

You can click the Applications tab of the OC4J home page to display the Applications page. This page contains a table of deployed applications. You can use this page to deploy and manage applications.

Note that the parent application for all the applications in the instance is default.

Maintaining Applications

Farm > Application Server: portal.edrsr16p1 > OC4J: home >
Application: BC4J Page Refreshed Sep 8, 2005 4:39:55 AM

General
[Redeploy](#) [Undeploy](#)
Status **Loaded**
Auto Start **true**
Parent Application default

Response - Servlets and JSPs
Active Sessions **3**
Active Requests **0**
Request Processing Time (seconds) **0.00**
Requests per Second **0.00**

Response - EJBs
Active EJB Methods **0**
Method Execution Time (seconds) **0.00**
Method Execution Rate (per second) **0.00**

Web Modules

Name	Path	Active Requests	Request Processing Time (seconds)	Active Sessions
webapp	webapp.war	0	0.00	3

EJB Modules

Name	Path	Active EJB Methods	Method Execution Time (seconds)
No EJB modules found			

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Maintaining Applications

You can monitor and administer the configuration of applications deployed to OC4J instances. The OC4J home page for Application Server Control provides a list of deployed applications and support for such common operations.

You can get a consolidated view of application performance and a list of the Web and EJB modules that are deployed for the application from the Application home page. You can get the aggregate application performance metrics, such as usage volume and responsiveness.

You can perform administration tasks, such as changing an application's user manager, adding data sources and security groups, and modifying application settings using the property pages linked with the application's home page. For example, if an EJB application uses container-managed persistence (CMP), the administrator can use the application property pages to control the process of creating and deleting the database tables used to track a CMP bean's session state.

Note: A CMP is a Java application that stores all state information in a container.

Maintaining Web Modules

Farm > Application Server: portal.edrsr16p1 > OC4J: home > Application: BC4J >

Web Module: webapp

Page Refreshed Sep 8, 2005 4:47:15 AM

General

Status

Loaded

URL Mapping

/webapp

Referenced EJBs

0

Response and Load

Active Sessions

3

Active Requests

0

Request Client Time (seconds)

0.00

Request Load Time (seconds)

0.00

Requests per Second

0.00

Requests Processed

327

Servlets/JSPs

Name ▲	Status	Type	Source	Active Requests	Request Client Time (seconds)	Requests per Second	Startup Priority
EMDServlet	Loaded	Servlet		0	0	0	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Maintaining Web Modules

You can drill down to an application module by using the link on the Application home page and further to a Web module home page.

The Web module home page contains a list of the deployed servlets, JSPs, or EJBs for the module. The list includes the status and performance of each object, such as response time and the volume of requests processed.

Web module properties can be viewed and edited from the Web Module home page. For example, you can change the URL mappings or add or remove chaining for servlet filters.

Summary

In this lesson, you should have learned how to:

- **Create OC4J instances**
- **Start and stop OC4J instances**
- **Enable or disable application startup**
- **Configure OC4J instance properties**
- **Configure Web site and JSP properties**
- **Edit the OC4J configuration files**
- **Configure deployed applications and Web modules**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

10

Deploying Java 2, Enterprise Edition (J2EE) Applications

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

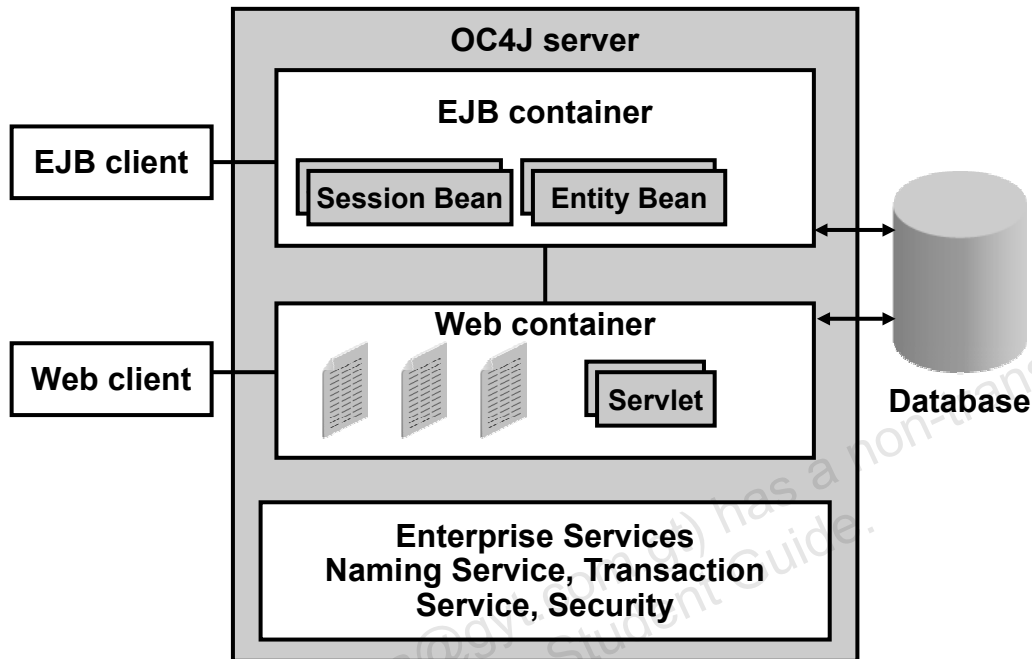
After completing this lesson, you should be able to do the following:

- **Deploy Web applications to Oracle Application Server**
- **Configure data sources to be used with OC4J**
- **Provide necessary mappings for an Oracle database**
- **Deploy J2EE applications**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

J2EE Architecture



ORACLE

Copyright © 2005, Oracle. All rights reserved.

J2EE Architecture

The diagram in the slide explains the major components in the J2EE logical architecture. It shows the set of classes, interfaces, and services that make up EJB architecture as well as a Web application.

Databases and J2EE

- **Many J2EE applications use a database.**
- **J2EE applications are designed to be portable across application servers; that is, they not dependent on operational details.**
- **Operational details are supplied by the deployer, which provides logical-to-physical mappings.**
- **Data sources provide logical definitions of databases.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Databases and J2EE

Today, most J2EE applications use a database to have a persistent storage for data. Servlets, as well as JSPs and EJBs, need to communicate with the database.

Connection details should not be stored in the application code itself. Application developers use a logical representation of a database in their codes. The deployer maps this logical representation to physical data sources. These data sources are published in the Java Naming and Directory Interface (JNDI) tree.

J2EE applications retrieve connections to the database through the `java.sql.DataSource` objects. The `DataSource` objects are looked up through JNDI using the published name. The connection methods are then used to connect to the specified database. There are different types of `DataSource` objects available to be configured.

Enterprise JavaBeans

- **Enterprise JavaBeans (EJB) is the server-side component architecture for the J2EE platform.**
- **EJB enables rapid and simplified development of distributed, transactional, secure, and portable Java applications.**
- **EJB applications can be ported across platforms without much difficulty.**
- **EJB applications are object oriented and allow reuse of code.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Enterprise JavaBeans

Enterprise JavaBeans is the server-side component architecture for the J2EE platform.

EJB applications allow developers to focus on the actual business architecture of the model, rather than worry about the programming and coding needed to connect all the working parts. This task is left to EJB server vendors. Developers just design (or purchase) the needed EJB components and arrange them on the server.

EJB applications can be ported on different platforms without much difficulty. Because they are object oriented, they can be implemented into existing systems with little or no recompiling and configuring.

EJB Structure

- The EJB modules are packaged as an EJB Java Archive (JAR) file.
- The EJB deployment tools use a standard format for packaging enterprise beans with their declarative information.
- The `ejb-jar.xml` file contains:
 - The deployment descriptor as specified by J2EE
 - The run-time attributes of the bean
- Using the `ejb-jar.xml` file, you can specify the run-time behavior of enterprise beans.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

EJB Structure

A deployment descriptor specifies the run-time attributes of the bean for deployment and loading into the database. Deployment descriptors are like property sheets for EJB components and are used for customizing and specifying the dependencies of the beans on the environment. For example, a deployment descriptor declares the transactional properties of the bean.

After the deployment descriptor is saved in a file called `ejb-jar.xml`, the enterprise beans, along with all the supporting classes and their deployment descriptors, are packaged and shipped in a `.jar` file.

At the time of deployment, the deployment descriptor is the first file to be read by the deployment tool. It notifies the deployment tool, the type of beans packed in the JAR file, their transaction management attributes, and the access permissions on the beans. This file is also a contract between the bean provider and the application assembler, and between the application assembler and deployer.

A metafile contained within the `ejb-jar` file describes the contents of the JAR file. It is useful for the deployment tools.

EJB and OC4J

- **EJB modules can be deployed to any J2EE-compliant server.**
- **Although the EJB module does not need to be modified, the module needs to be mapped to its server environment.**
- **The `orion-ejb-jar.xml` file provides the mapping for an EJB module to OC4J.**
- **The EJB modules should be packaged into an EAR before being deployed to Oracle Application Server.**

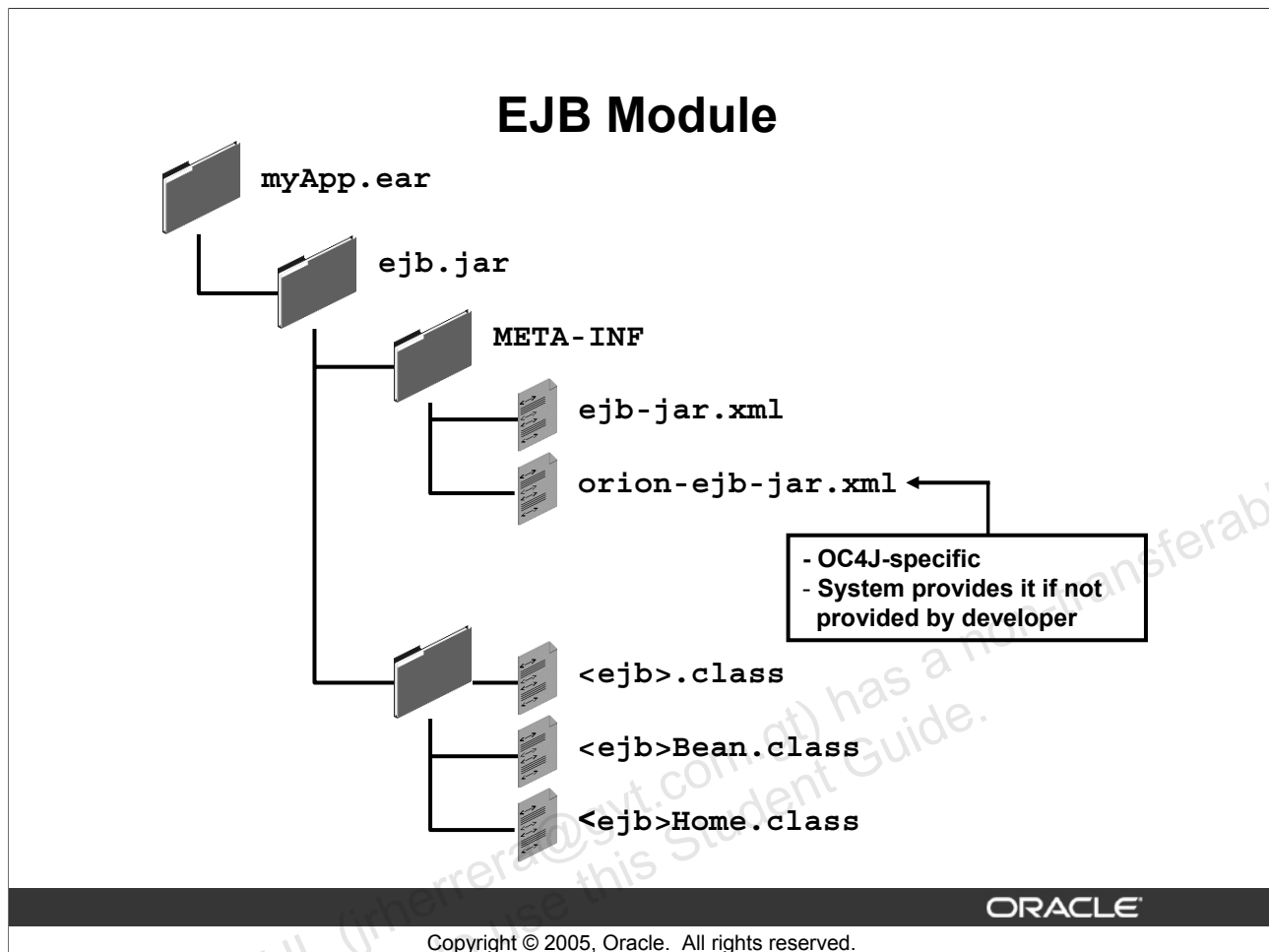
ORACLE

Copyright © 2005, Oracle. All rights reserved.

EJB and OC4J

The `orion-ejb-jar.xml` file contains the deployment time information for an EJB. It is located in the `$ORACLE_HOME/j2ee/home/application-deployments` directory. You can find the file under `/deploymentName/jarname(.jar)/orion-ejb-jar.xml` after deployment and under `META-INF/orion-ejb-jar.xml` below `ejb-jar` root if the EJB is bundled with the application or if no deployment directory is specified in the `server.xml` file.

If you use the `deployment-directory` attribute, which is the default, the bundled version is copied to the deployment location if and only if no file exists at that location. The `orion-ejb-jar.xml` file is used to specify the initial (first-time) deployment properties. After each deployment, the deployment file is reformatted, augmented, and altered by the server to add any new or missing information to it.



EJB Module

Simplified Configuration Customizing

Any Oracle-specific configuration information can be customized by manually editing a set of XML configuration files, which capture deployment and configuration information that is specific to Oracle Application Server. This information includes settings for the following:

- Automatically creating and deleting tables for CMP
- Security role mappings
- JNDI namespace access
- Session persistence and time-out
- Transaction-retry
- CMP and object-relational mappings
- Buffering
- Character sets
- Locales
- Virtual directories
- Cluster configuration
- Session-tracking
- Development and debugging mode

Deploying Web Application Modules Using Application Server Control

Deployed Applications						
			Deploy EAR file		Deploy WAR file	
Edit Undeploy Redeploy						
Select	Name	Path	Parent Application	Active Requests	Request Processing Time (seconds)	Active EJB Methods
	jpdsk	./applications/jpdsk.ear	default	0	0.00	0
	orauddi	./applications/orauddi.ear	default	0	0.00	0
	orauddi	./applications/orauddi.ear	default	0	0.00	0
	orauddi	./applications/orauddi.ear	default	0	0.00	0

Select the Web Application (.war file) you wish to deploy. This web application will be wrapped into a J2EE application (.ear file) before deployment.

Web Application

Specify the name you would like this application to be called and the URL to map this web application to.

Application Name

Map to URL

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Deploying Web Application Modules Using Application Server Control

To deploy applications to Oracle Application Server, the developer must package the components of the application into archives. You should, therefore, receive the application properly packaged from the developers (for example, myapp provided as myapp.war). Deploying an application to Oracle Application Server requires that a number of support files be updated. Although other options are available, it is recommended to use Application Server Control to deploy components to the application server. The Web application archive (WAR) file to be deployed should be copied to the client system from which you invoke Application Server Control. To deploy a WAR file to the application server, perform the following steps:

1. Navigate to the OC4J home page of your instance. Scroll down and click Applications.
2. In the Deployed Applications region, click Deploy WAR file. The first page of the Deployment Wizard appears.
3. On the Deploy Web Application page, to deploy your Web Application, perform the following steps:
 - a. Enter the path to your WAR file in the Web Application field, or click the Browse button.
 - b. The Application Name field refers to the unique name of the application. Enter the application name myapp.
 - c. The mapping of a URL to your Web application is specified in the Map to URL field. Enter /myapp in this field.

Oracle Application Server 10g R2: Administration I 10-9

Deploying Web Application Modules Using Application Server Control (continued)

4. Click the Deploy button to deploy your application. Oracle Enterprise Manager displays a confirmation page confirming that your application has been successfully deployed.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Deploying Web Application Modules Using `dcmctl`

The `dcmctl` utility enables you to deploy, redeploy, or undeploy a WAR file manually:

```
dcmctl deployApplication -file  
/private/myapp.war -a myapp -co home -rc  
myapp  
  
dcmctl redeployApplication -file  
/private/myapp.war -a myapp -co home -rc  
myapp  
  
dcmctl undeployApplication -a myapp -co home
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Deploying Web Application Modules Using `dcmctl`

The `dcmctl deployApplication` command deploys an application to the home instance determined by the `-co` option. The WAR or EAR file is supplied with the `-file` option. The `deployApplication` command copies the WAR or EAR file from the specified location to the server.

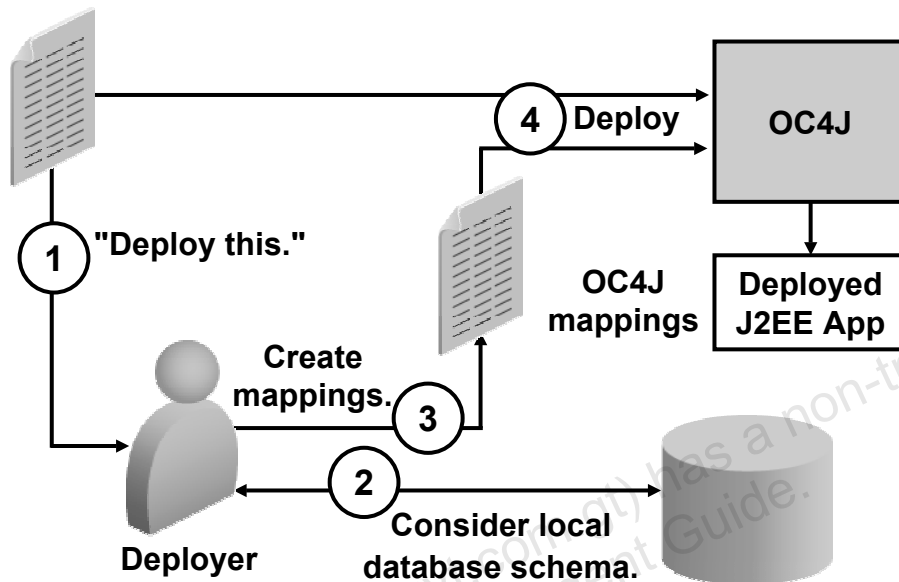
The application name denoted by `-a` is assigned to the application for administrative purposes. The name used to access the application from the Web is the name supplied with `-rc`.

The `-rc` option is required if the application is a WAR file. You should not use the `-rc` option when deploying an EAR file.

Using `dcmctl redeployApplication`, you can redeploy the specified application `myapp`. The `-file` option specifies the WAR or the EAR file, the `-co` option specifies the OC4J instance, and the `-rc` option specifies the name of the application used for administrative purposes.

To undeploy the indicated application, use the `dcmctl undeployApplication` command.

Data Sources and the Deployer Role



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Data Sources and the Deployer Role

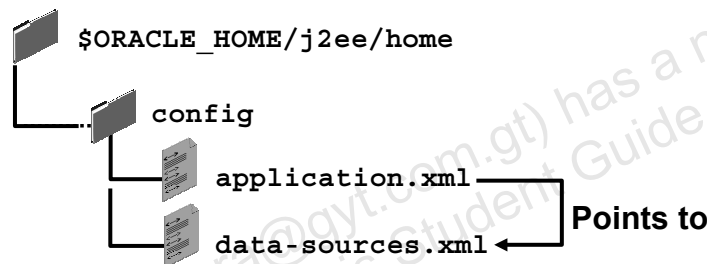
A data source is a Java object that has the properties and methods specified by the `javax.sql.DataSource` interface. Data sources are factory classes that return JDBC connections. J2EE applications use JNDI to look up `DataSource` objects. Every JDBC 2.0 driver has its own implementation of `DataSource` objects that can be bound into an external JNDI namespace. Using data sources is the recommended way for a J2EE application to get a connection. Data sources are preferred to the earlier JDBC `DriverManager` class. Data sources have logical names, which makes applications that use them much more portable.

The deployer is responsible for the mapping between logical `DataSource` object and the physical database connection.

1. After the developer has provided the deployer with the J2EE application archive, which is the EAR file, the deployer has to locate logical `DataSource` references in the deployment descriptor. The deployment descriptor specifies the run-time attributes of the bean for deployment and loading into the database.
2. The deployer considers the local database schema that matches the requirement.
3. The deployer creates the necessary mapping between the database and OC4J.
4. The deployer can deploy the files either manually or automatically.

Specifying Data Sources

- The global data sources for an OracleAS instance are specified in the `data-sources.xml` file:
 - Each data source is specified using an XML tag.
 - Attributes specify values for the data source.
- The application-specific data source files use the `<data-sources>` tag in `application.xml`.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specifying the Data Source

The `data-sources.xml` file in the `$ORACLE_HOME/j2ee/home/config` directory establishes data sources at the OC4J instance level.

Your application can directly use the data sources that are defined in the `data-sources.xml` file. Set the reference to the `data-sources.xml` file by using the `path` attribute in the `<data-sources>` tag of your application's `application.xml` configuration file.

OC4J parses the `data-sources.xml` file when it starts, instantiates `DataSource` objects, and binds them into the JNDI namespace of the OC4J server. Therefore, if you add a new data source specification to this file, you must restart the OC4J server instance to make the new data source available for lookup.

Each application also has a separate JNDI namespace. The `web.xml` and `orion-web.xml` files contain entries that can be used in mapping application JNDI names to data sources.

The XML definition for each data source includes a JDBC connection string and, optionally, a database account. After deploying, the application accesses the data sources at run time through a JNDI lookup.

Obtaining Data Source Information

Farm > Application Server: portal.edrsr16p1 >
OC4J: home

Home Applications Administration

Page Refreshed Sep 8, 2005 6:56:34 AM

Instance Properties
[Server Properties](#)
[Website Properties](#)
[JSP Container Properties](#)
[Replication Properties](#)
[Advanced Properties](#)

Application Defaults
[Data Sources](#)
[Security](#)
[JMS Providers](#)

OC4J Inheritance
OC4J applications have a hierarchical parent-child relationship to facilitate administration through inheritance. A child application inherits certain attributes from its parent application such as principals and JNDI objects including data sources, JMS providers and EJBs. When an OC4J application is deployed, you specify the parent application. The Default Application is the top of the parent hierarchy.

Data Sources

Page Refreshed Sep 8, 2005 6:57:44 AM

This table contains all the data sources configured for this application. Each data source is bound to the specified JNDI location.

Select a Data Source and... [Edit](#) [Create Like](#) [Delete](#) [Create](#)

Select	Name	JNDI Location	Class	JDBC Driver	Monitor Performance
<input checked="" type="radio"/>	OracleDS	jdbc/OracleCoreDS	com.evermind.sql.DriverManagerDataSource	oracle.jdbc.driver.OracleDriver	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Obtaining Data Source Information

You can use the OC4J home page of Application Server Control to get the data source information:

1. Click the Administration tab on the OC4J home page.
2. Click the Data Source link in the Application Defaults section. The Data Sources page is displayed. These are global data sources that can be used by all applications deployed in this OC4J instance.
3. To obtain information about data sources that are local to a particular application, drill down to that application page, and then select Data Source in the Administration region.

A preinstalled, default data source is an emulated data source. These data sources are wrappers around Oracle data sources. For example, you can use data sources that emulate the XA protocol for JTA transactions. Emulated data sources offer OC4J caching, pooling, and Oracle JDBC extensions for Oracle data sources. Nonemulated data sources provide full (nonemulated) JTA services, including two-phase commit capabilities for global transactions; however, they are not global (across databases) in nature.

The OracleDS default data source is an emulated data source. You can use this data source for applications that access and update only a single data server.

Obtaining Data Source Information (continued)

If you need to update more than one database and want these updates to be included in a Java Transaction API (JTA) transaction, then you must use a nonemulated data source.

The default emulated data source is very fast and efficient, because it does not enable two-phase commit operations. This is necessary if you were to manage more than a single database.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Sample data-sources.xml File

Configuring a JDBC thin connection data source:

```
<data-source
  class="com.evermind.sql.DriverManagerDataSource"
  name="OracleDS"
  location="jdbc/OracleCoreDS"
  xa-location="jdbc/xa/OracleXADS"
  ejb-location="jdbc/OracleDS"
  connection-driver="oracle.jdbc.driver.OracleDriver"
  username="scott"
  password="tiger"
  url="jdbc:oracle:thin:@localhost:1521:oracle"
  inactivity-timeout="30"
/>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Sample data-sources.xml File

In this example, the attributes of the data-source tag are defined as follows:

- `class` defines the type of data source or which `DataSource` class you want to use.
- `name` identifies this data source. It defaults to the value of `location`.
- `location`, `xa-location`, and `ejb-location` attributes are JNDI names that this data source is bound to within the JNDI namespace. You should use only the `ejb-location` JNDI name in the JNDI lookup for retrieving this data source.
- `connection-driver` defines the type of connection you expect to be returned to you from the data source.
- `username` identifies the database username.
- `password` specifies the password for the database user.
- `url` is the JDBC connection URL.
- `inactivity-timeout` defines the time (in seconds) to cache unused connections before closing them. The default is 30 seconds.

Creating a Data Source: General

The Create Data Source page includes the following regions: General, Datasource Username and Password, JNDI Locations, Connection Attributes, and Properties.

General	
* Name	OracleDS
Description	
* Data Source Class	com.evermind.sql.DriverManageDataSource
JDBC URL	jdbc:oracle:thin:@//localhost:1512/oracle.regress.rdbms.c
JDBC Driver	oracle.jdbc.driver.OracleDriver <small>This field is required if you are using a generic Orion Data Source Class.</small>
Schema	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

General Section

You can use the General region to define the following aspects about a data source:

- Name is a user-defined name to identify the data source.
- Description can be used for a user-defined description of the data source.
- Data Source Class name specifies the class that the data source is instantiated as, for example, `com.evermind.sql.ConnectionDataSource`.
- JDBC URL specifies the URL to the database represented by this data source. For example, if using an Oracle Thin driver, the URL could be the following:
`jdbc:oracle:thin:@my-lap:1521:ORCL`
- JDBC Driver is the JDBC driver to use. An example of a JDBC driver is `oracle.jdbc.driver.OracleDriver`.
- Schema is an optional parameter. By using the data source schema attribute, you can associate a data source with a `database-schema.xml` file that you can customize for its particular database. Associating a data source with a `database-schema.xml` file allows you to influence what SQL is ultimately generated by the container. This can help you solve problems such as accommodating additional data types supported in your application (for example, `java.math.BigDecimal`), but not in your database.

Creating a Data Source: Username and Password

Datasource Username and Password	
Cleartext passwords may pose a security risk, especially if the permissions on the data-sources.xml configuration file allows it to be read by any user. You can specify an indirect password to avoid this risk. An indirect password is used to do a look up in the User Manager to get the password.	
Username	<input type="text" value="scott"/>
<input checked="" type="radio"/> Use Cleartext Password	<input type="text" value="Password"/>
<input checked="" type="radio"/> Use Indirect Password	<input type="text" value="Indirect Password"/>
example: Scott, customers/Scott	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Username and Password Section

You can use this section to enter the username and password for accessing the data source. You can specify clear text (not encrypted) password or use a more secure indirect password.

An indirect password is made up of a special indirection symbol (->) and a username (or username and realm). When OC4J encounters an indirect password, it uses its privileged access to retrieve the password associated with the specified user from the security store provided by a user manager.

If the Indirect Password field contains `scott`, then a user named `scott` with the password should have been created in the User Manager. The corresponding verification will be taken care of by the User Manager.

Note: The User Manager is a Web-based tool that is used to perform user-support tasks, such as creating a new user, resetting the PIN and password for a user, assigning a special role to a user, or troubleshooting any issues that occur while using mobile applications.

Creating a Data Source: JNDI Locations

JNDI Locations	
For an emulated Data Source, please specify all three location attributes. It is recommended that you reference the EJB Location attribute in your code to look up this Data Source. For a non-emulated Data Source, the location attribute is all that is needed.	
* Location	<input type="text" value="jdbc/OracleCoreDS"/>
Transactional(XA) Location	<input type="text" value="jdbc/xa/OracleXADS"/>
EJB Location	<input type="text" value="jdbc/OracleDS"/>
For emulated data sources, retrieve the data source using the JNDI value in this field.	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Creating a Data Source: JNDI Locations

You can use this region to define the JNDI location string that the data source is bound with. Previous releases supported the `location` and `xa-location` attributes for retrieving data source objects. The JNDI location string is used within JNDI lookups for retrieving the data source. Applications, EJBs, servlets, and JSPs should use only the JNDI-named `ejb-location` in an emulated data source.

All the three values must be specified for emulated data sources, but only `ejb-location` is actually used.

If you want EJBs to use this data source for container-managed persistence (CMP), then you need to specify values for both the `Transactional (XA)` and `EJB Location` attributes. However, `Location` and `XA Location` should not be used for an emulated data source.

Note: Container-managed persistence (CMP) occurs when the entity object delegates persistence services. With CMP, the EJB container transparently and implicitly manages the persistent state. The enterprise bean developer does not need to code any database access functions within the enterprise bean class methods.

Creating a Data Source: Connection Attributes and Properties

Connection Attributes	
Connection Retry Interval (seconds)	1
Max Connection Attempts	
Cached Connection Inactivity Timeout (seconds)	30
The following attributes only apply if you are using pooled data sources	
Maximum Open Connections	
Minimum Open Connections	
Wait For Free Connection Timeout (seconds)	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Connection Attributes and Properties

Using the Connection Attributes region, you can modify connection tuning parameters, including the retry interval, pooling parameters, timeout parameters, and maximum attempt parameter. You can use:

- The Connection Retry Interval (seconds) field to define the interval to wait (in seconds) before retrying a failed connection attempt. The default value is 1 second.
- The Max Connection Attempts field to specify the number of times to retry making a connection. This is useful when the network is not stable or the environment is unstable for any other reason that sometimes make connection attempts fail.

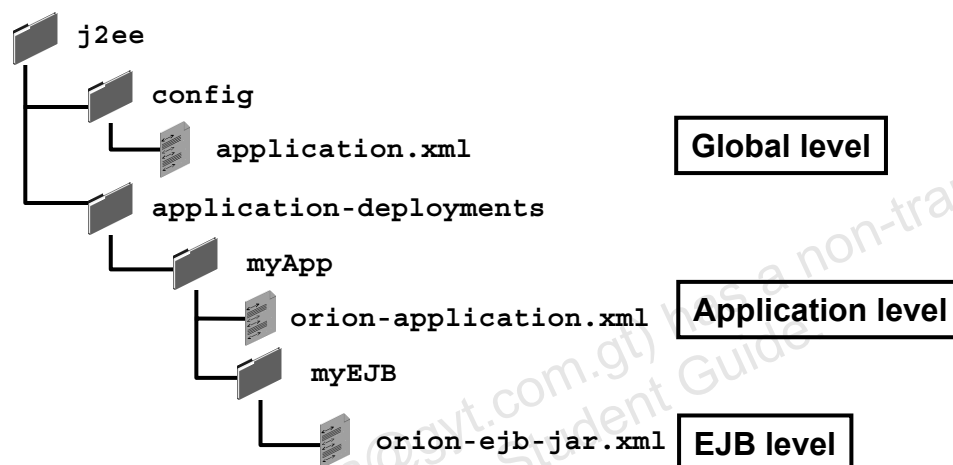
For pooled data sources, you can use:

- The Maximum Open Connections and Minimum Open Connections fields to determine the maximum and minimum number of open connections, respectively, for a pooled data source. There is no default provided.
- The Wait For Free Connection Timeout (seconds) field to set the number of seconds to wait for a free connection if the pool is used up. The default is 60 seconds.

Entries to the Properties region may be necessary when configuring a custom or third-party data source.

Specifying CMP Data Source

The files to configure the data source details for an application are provided by the developers.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Specifying CMP Data Source

Container-managed persistence (CMP) beans need a data source to be defined. You specify a default data source in the `orion-application.xml` file, which can be overridden for a given entity bean within the `entity-deployment` tag in `orion-ejb-jar.xml`. These settings override the global OC4J default data source specified in `application.xml`.

Note: When no data source is specified, OC4J uses the first data source in `data-sources.xml` as the default data source.

OC4J automatically creates tables for entity bean persistence so that fields are mapped correctly to database types. This behavior is controlled by the following attributes in `orion-application.xml`:

- **autocreate-tables:** Whether to automatically create database tables for CMP beans in this application. The default is `True`.
- **autodelete-tables:** Whether to automatically delete old database tables for CMP beans when redeploying in this application. The default is `False`.
- **default-data-source:** The default data source to use if other than the server default. If specified, this must point to a valid CMP (`ejb-location`) data source for this application.

Binding EJBs to Existing Tables

1. Set `autocreate-tables` to `False`.
2. Deploy `yourEjb` in `yourApp.ear`.
3. Get the generated `orion-ejb-jar.xml` file and reconfigure it to target existing tables.

```
<orion-ejb-jar>
  <enterprise-beans>
    <entity-deployment table="yourTable">
      ...
    </entity-deployment>
  </enterprise-beans>
</orion-ejb-jar>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Binding EJBs to Existing Tables

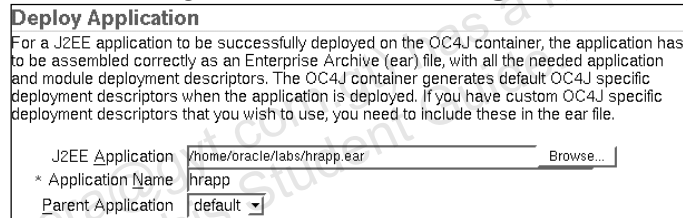
If you want to bind your EJB to already existing tables, you must modify `orion-application.xml` and set the `autocreate-tables` attribute to `False`. Then wrap your EJB, named `yourEjb` in the example, in an enterprise archive and deploy it. This generates the `orion-ejb-jar.xml` file that can be configured appropriately to your needs.

Deploying J2EE Applications Using Application Server Control

1. Navigate to the Applications properties page and click Deploy EAR file.



2. To select the J2EE application that you want to deploy, use the Deploy Application page.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Deploying J2EE Applications

To deploy a new EJB application, perform the following steps:

1. Navigate to the OC4J Applications Properties page of Application Server Control. Click the Deploy EAR file button.
The Deploy Application Wizard is displayed. The Deploy Application Wizard leads you through the process of creating a J2EE application, including URL mappings for Web modules, resource reference mappings, and security role mappings. As you navigate through the wizard, a navigation trail appears at the top of the page to indicate your position in the wizard.
2. To select the J2EE application that you want to deploy, use the Deploy Application page. The application file is contained in an Enterprise Archive (EAR) file with all required application and module deployment descriptors.
 - a. Enter the location of the EAR file.
 - b. Enter the name of the application you are deploying in the Application Name field.
 - c. Select the parent application. The parent application helps you organize your application library.

Deploying J2EE Applications Using Application Server Control

3. To map a Web module to a URL pattern in a Web site, use the URL Mappings for Web Module page.

URL Mappings for Web Modules User Manager Review

Deploy Application: URL Mapping for Web Modules

A web module needs to be mapped to an URL pattern in the default web site before it can be accessed. The following table lists all the web modules found in your application. Specify the URL mapping for each of these modules.

Name	URL Mapping
	/hrapp

Cancel Step 1 of 3 Next Finish

Deploying J2EE Applications (continued)

3. To map a Web module to a URL pattern in a Web site, use the URL Mappings for Web Module page. Click Next to move to the User Manager page.

Deploying J2EE Applications Using Application Server Control

4. To configure which User Manager to use for security, use the User Manager page.

URL Mappings for Web Modules **User Manager** Review

Deploy Application: User Manager

Specify a user manager to be associated with the application. Note that all web modules in your application will be automatically SSO enabled, when use JAZN LDAP as your user manager.

☒ Use JAZN XML User Manager
Default Realm
XML Data File

☐ Use XML User Manager
Path to principals file

☐ Use JAZN LDAP User Manager
LDAP Location
Default Realm

☐ Use Custom User Manager
Name
Class Name
Description

Initialization Parameters for Class	
Select Name	Value
No initialization parameters	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Deploying J2EE Applications (continued)

4. To configure which User Manager to use for security, use the User Manager page. For complete security, you should choose the JAZN XML User Manager. You must already have your User Manager set up and configured. Most of the entries on this page require an XML file that designates the security roles, users, and groups for your security mappings.

If you select XML User Manager, you will find that this is not the most secure option. It requires a `principal.xml` file to be configured.

To use JAZN XML User Manager as the recommended User Manager, you must set up a default realm and a `jazzn-data.xml` file. If you choose JAZN LDAP User Manager, this requires a default realm and an LDAP location.

Before you can use Custom User Manager, this User Manager must be programmed; enter the class name in this field.

Provide the necessary information for the User Manager of your choice, and click Next to move to the Summary page.

Deploying J2EE Applications Using Application Server Control

5. To review your selections and statistics for the application you are deploying, use the Summary page.

URL Mappings for Web Modules User Manager **Review**

Deploy Application: Review

Ear File to Deploy **transtrace.ear**
Deployment Destination **Instance OC4J_Temp**
URLs Mapped to Application **hrapp**

☒ **TIP** The HTTP listener will be restarted after deployment, to pick up the new web module mappings.

Deploying J2EE Applications (continued)

5. To review your selections and statistics for the application you are deploying, use the Summary page. Click Deploy on the Summary page to deploy the application.

Now, you are ready to test the application. Use Application Server Control to determine the Oracle HTTP Server port. Enter the URL to run the application from the browser:

`http://<host name>.<domain>:<Oracle HTTP Server port>/hrapp`. You should be able to see the application's welcome page.

Monitoring J2EE Applications

Farm >	
Application Server: portal.edrsr16p1	
Home J2EE Applications Ports Infrastructure Backup/Recovery	
Name ▲	OC4J Instance
ADFBCManager	home
BC4J	home
default	OC4J Portal

Farm > Application Server: portal.edrsr16p1 > OC4J: OC4J Portal >	
Application: default	
Page Refreshed Sep 8, 2005 8:38:49 AM	
General	Response - Servlets and JSPs Active Sessions 0 Active Requests 1 Request Processing Time (seconds) 0.004 Requests per Second 0.08
Status Active	Response - EJBs Active EJB Methods 0 Method Execution Time (seconds) 0.00 Method Execution Rate (per second) 0.00
Web Modules	
Name	Path
defaultWebApp	../home/default-web-app
dms	../home/applications/dms.war

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Monitoring J2EE Applications

You can review and monitor the performance of your J2EE applications that are deployed and maintained with Oracle Application Server. Click J2EE Applications on the Application Server home page to display a list of applications deployed from this application server instance. From the list of J2EE applications, you can navigate to the OC4J instance or application page for information about the performance and availability of each application that you have deployed.

- Verify that the OC4J instances that contain your J2EE applications are running.
- Check the status for your applications.
- If your J2EE applications are not loaded, then deploy them and access the application to verify that they are working properly.

Click an application name to navigate to the application home page. From the application home page, you can monitor and administer the application's status, response, and the Web and EJB modules.

The General section shows the current status of the OC4J application. You can also redeploy or undeploy an OC4J application from here.

Monitoring J2EE Applications (continued)

You can monitor:

- Transactional details about the server from the Response-Servlets and JSPs section
- Transactional details about the Enterprise Java Beans from the Response-EJBs section
- The list of all Web modules used in the OC4J application from the Web Modules section
- The list of all EJB modules used in the OC4J application from the EJB Modules section

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Deploying J2EE Applications Using `dcmctl`

- The OC4J instance must be running.
- The application can be deployed locally or to a remote OC4J instance.
- Examples of deploying applications using `dcmctl`:

```
To the current OracleAS instance
$> dcmctl deployApplication \
> -file /export/users/myEAR.ear \
> -a myEAR -co home
```

```
To a specific (j2ee01) OracleAS instance
$> dcmctl deployApplication -i j2ee01 \
> - file /export/users/myEAR.ear \
> - a myEAR
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Deploying J2EE Applications Using `dcmctl`

The `dcmctl deployApplication` command deploys an application to the current instance using the EAR file supplied with the `-file` option. The application name is assigned to the application for administrative purposes. The name used to access the application from the Web is still the name supplied in the EAR file.

On hosts with multiple OC4J instances, `dcmctl` determines the target OC4J instance as follows:

- If an OC4J instance is specified with the `-co` target option, the operation applies to that OC4J instance within the associated application server instance. The application server instance is determined first by the `-oraclehome` option, and then by the Oracle home directory in which the `dcmctl` executable resides. If the application server instance is part of a cluster, apply the operation to all OC4J instances with the specified name within the cluster.
- If you do not supply the `-co` target option, the deployment applies to all OC4J instances within the associated application server instance. The application server instance is determined first by the `-oraclehome` option, and then by the Oracle home directory in which the `dcmctl` executable resides.

Summary

In this lesson, you should have learned how to:

- **Deploy Web applications to Oracle Application Server**
- **Identify the configuration file that stores data sources**
- **Configure data sources to be used with OC4J**
- **Provide necessary mappings for an Oracle database**
- **Deploy J2EE applications**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

11

Oracle Application Server Security Services

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **List the risks in an Internet environment**
- **Describe the available security services**
- **Describe the Oracle Application Server security architecture**
- **Explain the role of individual Oracle Application Server components in the security architecture**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Security Risks in an Internet Environment

- **Data tampering and fraud**
- **Eavesdropping and data theft**
- **Falsifying user identities**
- **Password-related threats**
- **Unauthorized access to data**
- **Lack of accountability**
- **Hacking**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Security Risks in an Internet Environment

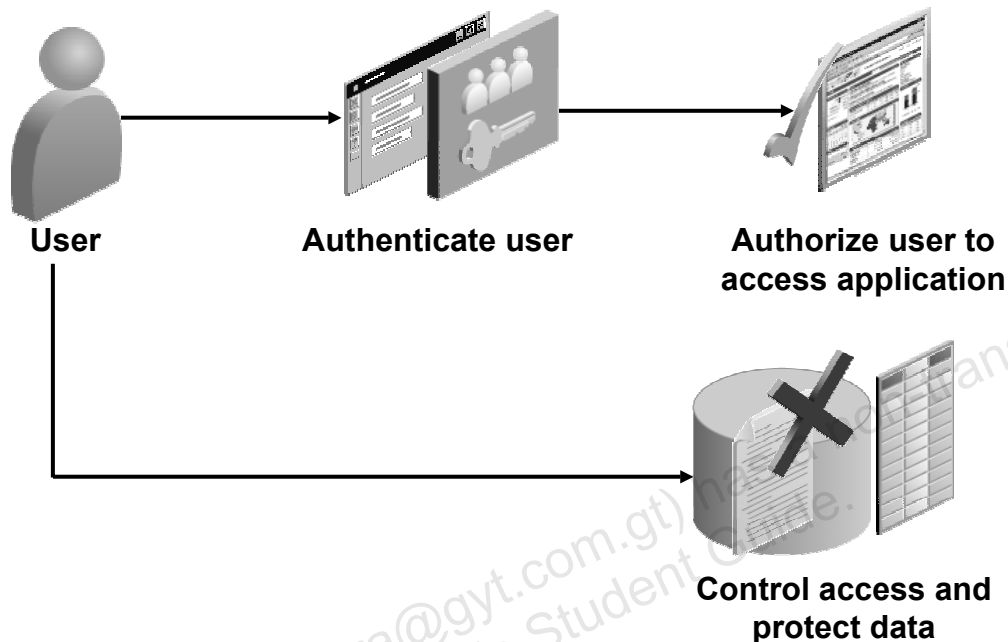
The threats that exist in a Web environment are outlined as follows:

- **Data tampering and fraud:** Privacy of communications is essential to ensure that data cannot be modified or viewed in transit. Distributed environments bring with them the possibility that a malicious third party can perpetrate a computer crime by tampering with data as it moves between sites. An example of this is changing the dollar amount of a banking transaction from \$100 to \$10,000.
- **Eavesdropping and data theft:** Data must be stored and transmitted securely, so that information, such as credit card numbers, cannot be stolen. Network watching programs can be easily installed to eavesdrop on network traffic. Packet observing programs can be designed to find and steal usernames and passwords.
- **Falsifying user identities:** Identity theft is becoming one of the greatest threats to individuals in the Internet environment. Criminals steal personal data, such as checking account numbers and driver's license numbers, and set up fake credit accounts in someone else's name.

Security Risks in an Internet Environment (continued)

- **Password-related threats:** In large systems, users must remember multiple passwords for the different applications and services that they use. As a result, there are several things they may do which compromise password security:
 - They may select easily guessed passwords, such as a name, fictional character, or word found in a dictionary. All of these passwords are vulnerable to dictionary attacks.
 - They may standardize passwords, so that they are the same on all machines or Web sites. This results in a potentially large exposure in the event of a compromised password. Or, they may use passwords with slight variations that can be easily derived from known passwords.
 - Users with complex passwords may write them down where an attacker can easily find them, or they may just forget them, requiring costly administration and support efforts.
- **Unauthorized access to data:** The role of an application server is to mediate access to the back-end database. When the application server has performed its function, the database's native access controls take over. The user's database privileges must determine the specific data (that is, the tables, columns, and rows) that is accessible to him or her.
- **Lack of accountability:** If the system administrator is unable to track users' activities, then users cannot be held responsible for their actions. There must be some reliable way to monitor who is performing which operations on the data. Therefore, a logging service that can be used to audit security-related events is necessary. Database auditing capabilities can also be used.
- **Hacking:** Hackers may try to corrupt your Web site. Or, they may try to steal a Web connection and redirect the user to a different site, fooling the client or server into believing that the site is something which it is not. To prevent corruption, you can control access to administrative functions that govern the content of the site. You can employ user authorization and encryption to help protect against stolen Web connections.
- **Denial-of-service attacks:** Data and Web security also involve the accessibility of information to authorized users, as needed. Although system security by itself does not ensure availability, availability may not be possible without security. Availability is often thought of as continuity of service, ensuring that a Web site, application, or database is available at all times. Although most aspects of this issue are not directly related to security, the fact is that security vulnerabilities can seriously compromise system availability. System availability can be protected by secure configuration and appropriate allocation of resources. The system should be set up in such a way as to avoid exposing any vulnerabilities, which could be exploited by a malicious intruder.
- **Complex user-management requirements:** Systems must often support thousands, or hundreds of thousands, of users; therefore, they must be scalable. In such large-scale environments, the burden of managing user accounts and passwords make your system vulnerable to error and attack. You need to know who the user really is, across all tiers of the application, to have reliable security.

Security Services in an Internet Environment



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Security Services in an Internet Environment

Fundamental security services required in a multiuser, networked environment include the following:

- **Authentication:** When users or systems request access to services or data, the authentication process enables a system to verify the identity of the user or the system.
- **Authorization:** Authorization is required for effective access control. The authorization process determines the privileges that users and other systems have for accessing resources.
- **Access Control:** On the basis of the authenticated identity and authorization privileges of the user, a system grants resources in ways that are consistent with security policies defined for those resources.
- **Data Protection:** Prevent unauthorized users from accessing sensitive data, such as passwords, by encryption mechanisms.

Addressing the Security Challenges

- **Deep data protection**
- **Internet-scale security**
- **Secure hosting and data exchange**

ORACLE

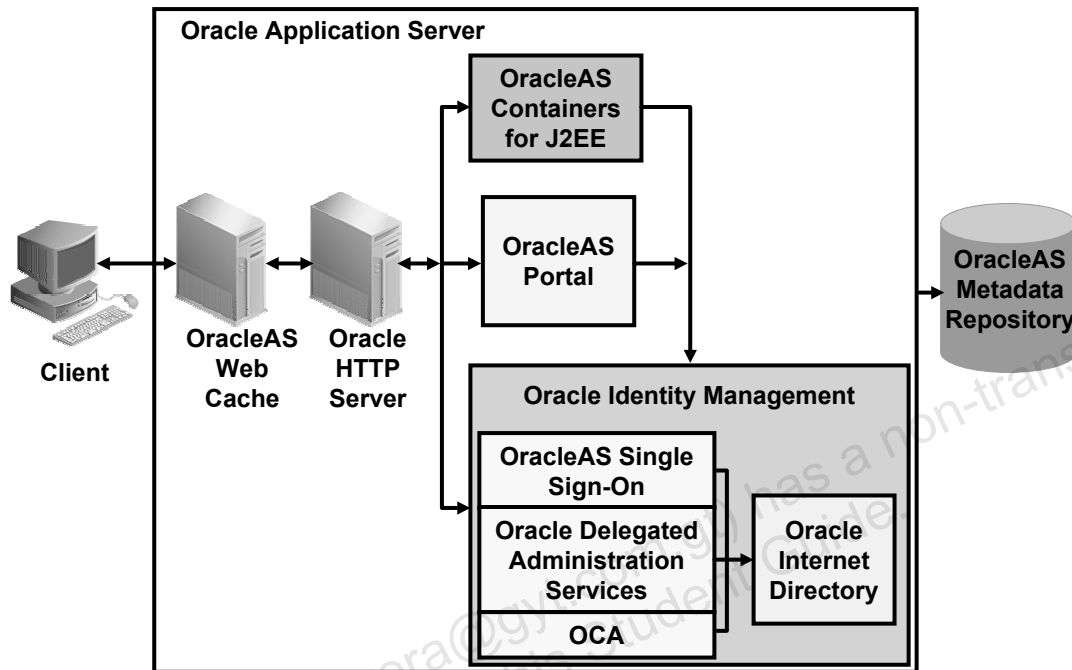
Copyright © 2005, Oracle. All rights reserved.

Addressing the Security Challenges

Oracle Application Server addresses e-business security challenges through deep data protection, Internet-scale security, and secure hosting and data exchange:

- **Deep data protection:** Among the best ways to reduce security risk on the Internet is to provide multiple layers of security mechanisms, so that the failure of a single mechanism does not result in the compromise of critical information. This concept is referred to as *deep data protection*. Oracle Application Server provides it through data encryption, extensive auditing, and access control.
- **Internet-scale security:** Security mechanisms must be scaled to Internet size, supporting many thousands or millions of users, and still be practical to administer. Oracle Application Server provides a number of security features that are customized to building Internet-scale applications, including proxy authentication, support for Internet standards (such as secure sockets layer [SSL]) and relevant public key infrastructure (PKI) standards, Java security, and enterprise user security features (such as directory-based privilege management).
- **Secure hosting and data exchange:** These enable economical, secure partitioning of data access by customer or by user, while supporting secure data sharing among communities of interest. Oracle Application Server makes this possible through support for a public key infrastructure and enterprise user security.

Oracle Application Server Security Architecture



Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Security Architecture

Oracle Application Server 10g implements security features in each component from the Web Cache to the back-end database. Each individual application server component integrates its security features into the security of the application server as a whole.

- OracleAS Web Cache, which can be configured to support HTTPS, is positioned in front, where it caches frequently accessed Web pages or partial pages.
- Oracle HTTP Server supplies Web listener services for both HTTP and HTTPS and, through plug-ins, routes requests for authentication and authorization.
- Oracle Application Server Containers for J2EE (OC4J) provides Java Runtime Environment for Oracle Application Server components. Java Authentication and Authorization Service (JAAS) provider ensures secure access to and execution of Java applications along with integration of Java-based applications with Oracle Application Server Single Sign-On.
- OracleAS Portal provides the infrastructure to create and manage Web pages. It enables you to display multiple Web pages on each portal page, with links to contents through Java applications. The portal uses Oracle Application Server Single Sign-On to provide single sign-on capabilities for secure access to contents and applications.

Oracle Application Server Security Architecture (continued)

- Oracle Identity Management provides an integrated infrastructure of directory, security, and user-management functionality on which Oracle products rely. Oracle Identity Management includes:
 - Oracle Internet Directory
 - Oracle Delegated Administration Service
 - OracleAS Single Sign-On
 - OracleAS Certificate Authority

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

What Is SSL?

- **Secure sockets layer (SSL) is an industry-standard protocol for securing network connections.**
- **SSL involves three mechanisms:**
 - **Encryption**
 - **Authentication**
 - **Data Integrity**
- **Oracle Application Server supports SSL, versions 2 and 3, and Transport Layer Security (TLS), version 1.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is SSL?

Using simple authentication can pose a serious security risk on the Internet. If hackers can access your account, they can do untold damage. To avoid such attacks, you have the option of using secure sockets layer (SSL) instead of just the username and password to authenticate.

SSL is an industry-standard protocol for securing network connections. It authenticates a user's identity through the exchange of certificates. It protects data during transmission by using encryption and data-integrity algorithms. The SSL protocol is an appropriate way to certify that messages you received were indeed sent by the sender. The technologies for this security are rooted in public-key encryption, which is currently the only feasible way to implement security over an insecure network, such as the Internet.

SSL uses the following three mechanisms:

- **Encryption:** Encryption is a mechanism for scrambling and unscrambling data. SSL can use different encryption algorithms to encrypt messages. Examples of encryption algorithms include Advanced Encryption Standard (AES), RC4, and Data Encryption Standard (3DES).

Note: TLS is a protocol that provides communications privacy over the Internet. This protocol enables client-server applications to communicate in a way that prevents eavesdropping, tampering, or message forgery.

What Is SSL? (continued)

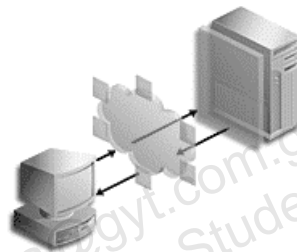
- **Authentication:** Authentication is a mechanism by which one party provides its identity to another party. When a client requests for SSL session, the server sends its certificate to the client.
The client verifies that the server is authentic by validating the server certificate. The server can also require the client to have a certificate in order to authenticate the identity of the client.
- **Data integrity:** Data integrity is a mechanism for verifying that all the data transmitted is received correctly. In this mechanism, the client hashes the message into a digest using a hash function and sends this message digest to the server. The server also hashes the message into a digest and compares the digests. It is not possible to produce the same digest from two different messages. Therefore, if the digests do not match, then it indicates that someone has tampered with the message. An example of a hash function supported by SSL is SHA1.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Private and Public Key Cryptography

SSL provides message integrity, authentication, and encryption:

- **Based on the concept of public key cryptography**
- **Through two types of encryptions:**
 - **Public key**
 - **Private or symmetric key**



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Private and Public Key Cryptography

SSL uses private and public key cryptography to provide message integrity, authentication, and encryption.

Public Key Encryption

Public key encryption employs public and private key pairs. A public key is used to encrypt messages that can be decrypted only by using the corresponding private key. The public key is securely stored, with other security credentials in an encrypted container (for example, an Oracle wallet). The private key is generated when the server and the client first establish an SSL connection. The private key is a randomly generated key for the session and is not stored. Public key algorithms do not verify the identities of the communicating parties. To verify the owner of the public key, you can use digital certificates.

Digital Certificates

Digital certificates are the electronic counterparts to driver licenses, passports, and membership cards. Digital certificates electronically prove your identity or your right to access information or services online.

Digital certificates, also known as digital IDs, bind an identity to a pair of electronic keys that can be used to encrypt and sign digital information.

Private and Public Key Cryptography (continued)

A digital ID makes it possible to verify someone's claim that they have the right to use a given key, helping to prevent people from using phony keys to impersonate other users. Used in conjunction with encryption, digital IDs provide a more complete security solution, assuring the identity of all parties involved in a transaction.

Typically, digital certificates contain the following information:

- The owner's public key
- The owners name
- The expiration date of the public key
- The name of the issuer (the CA)
- The serial number of the digital ID
- The digital signature of the issuer

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Public Key Infrastructure (PKI)

You can use public key certificates for the following:

- **Enabling secure and reliable authentication of users**
- **Ensuring the integrity of transmitted data**
- **Preventing unauthorized access to information when transmitted or stored**
- **Precluding repudiation of electronic transactions**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Public Key Infrastructure (PKI)

Reliable authentication relies on two factors:

- The proof of possession of a private key, which is verified by an automatic procedure that uses the public key.
- Validation by a Certificate Authority that a public key belongs to a specific identity. A PKI-based digital certificate validates that identity connection based on the key pair.

The certificate that binds the private key/public key to a user is issued by a trusted entity after it verifies the authenticity of the person who is requesting a certificate. Such certificates are practically impossible to forge or alter. Therefore, they are used to authenticate the identities of users and servers over closed or open networks. Authentication by PKI certificate is analogous to identifying oneself with an official driver's license or passport. The certificate contains the user's identification, the certificate's serial number and validity period, the user's public key, and the digital signature of the issuing authority.

Such reliable authentication reduces fraudulent use of digital identities to a minimum and supports secure use of public key encryption. PKI's certificates, key management, and encryption tools thus ensure the integrity of transmitted data, the reliability of electronic transaction signatures preventing repudiation, and strong confidentiality of transmitted or stored data.

Public Key Infrastructure (PKI) (continued)

The X.509 version 3 standard introduced extensions enabling separate certificates for SSL, encryption, and digital signatures.

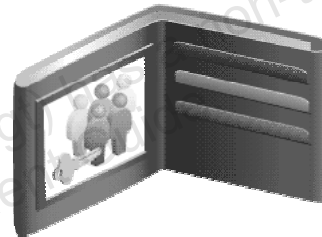
Trusting a certificate to legitimately represent prior verification of an identity linked with a public key means trusting the authority that issued it: the Certificate Authority (CA). The CAs often rely on another entity, a Registration Authority (RA), to validate the information supplied on requests for certificates, before the CA issues the certificate to the requesting party.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Storing Secure Credentials

A wallet:

- **Is a database that is used to manage authentication data**
- **Stores secure credentials such as digital certificates**
- **Manages security credentials on the server and client**



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Storing Secure Credentials

Organizations usually store credentials in addition to user and authorization information in an LDAP-compliant directory. With PKI, secure credentials such as digital certificates can be stored in containers called wallets.

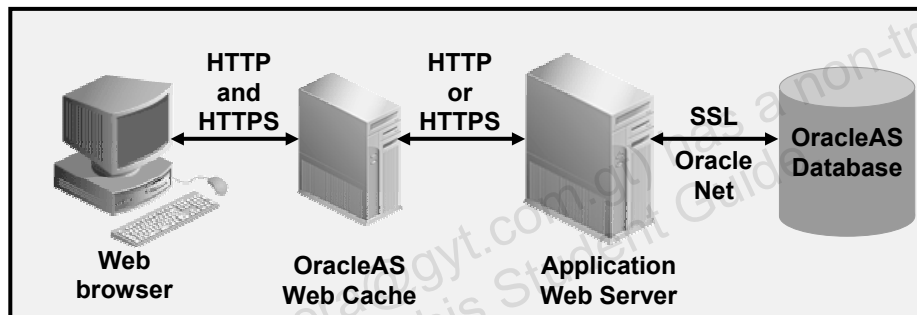
A wallet is a transparent database used to manage authentication data such as keys, certificates, and trusted certificates required by SSL. Security administrators often use Wallet Managers to manage security credentials on the server. Wallet owners use Wallet Managers to manage security credentials on the client.

Public Key Certificate Standard #12 (PKCS#12) is the standard for secure credential storage.

OracleAS Web Cache Security

You can implement security in OracleAS Web Cache by:

- Restricting administration
- Using secure sockets layer (SSL)
- Enabling SSL acceleration



ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Web Cache Security

You can use the following security-related features in OracleAS Web Cache:

- **OracleAS Web Cache restricts administration and invalidation operations by:**
 - Authenticating passwords
 - Controlling the ports used
 - Restricting IPs and subnets
- **Support for Secure Sockets Layer (SSL):** SSL protocol is a standard for network transport layer security. SSL provides authentication, encryption, and data integrity. By supporting SSL, OracleAS Web Cache is able to cache pages for HTTPS protocol requests.
- **SSL acceleration:** SSL operations can place a strain on server CPU resources, and also slow down the overall performance. Oracle Application Server supports nCipher's BHAPI-compliant hardware for deployment on servers running OracleAS Web Cache and Oracle HTTP Server. The nCipher hardware offloads the SSL key exchange processing from a server's CPU, thus improving response times.

Oracle HTTP Server Security

Oracle HTTP Server controls access to resources in the following ways:

- Through basic authentication
- By authenticating through OracleAS Single Sign-On using `mod_osso`
- Based on the request
- Defining filters
- Using the SSL protocol

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle HTTP Server Security

Oracle HTTP Server provides support for authentication, authorization, and data security. It allows users to access Oracle Application Server using standard Web protocols. The HTTP listener in Oracle HTTP Server supports HTTP and HTTPS. Oracle HTTP Server's security infrastructure is provided by the following modules:

Apache modules:

- **`mod_auth`**: Provides authentication based on username and password
- **`mod_access`**: Controls access to the server based on the request
- **`mod_security`**: Provides a noninvasive method to define filters to detect anomalies, such as SQL injections, and prescribes appropriate actions

Oracle modules:

- **`mod_oss1`**: Provides authentication and encryption with X.509 client certificates over SSL
- **`mod_osso`**: Enables single sign-on authentication for Web applications

J2EE Security and JAAS

- **Oracle Application Server Java Authentication and Authorization Service (JAAS) provides key security services to the Java programmer in the following areas:**
 - **Authentication to identify users**
 - **Authorization to limit what users can do**
 - **Delegation to enable code to be run securely**
- **Oracle JAAS implementation supports the following provider types:**
 - **XML-based provider**
 - **LDAP-based provider**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

J2EE Security and JAAS

Oracle Application Server introduces a comprehensive, integrated security framework supporting all Oracle Application Server components, as well as third-party and custom applications deployed on Oracle Application Server. The framework is based on Oracle Application Server Single Sign-On for authentication, Oracle Internet Directory for authorization and user provisioning, and Java Authentication and Authorization Service (JAAS) for security services in J2EE.

Because Java development is very important to the Web, Java security is a vital feature of Oracle Application Server. JAAS, which is part of the J2EE 1.3 specification, is the latest development in the Java security architecture. It extends the security architecture of the Java 2 platform with additional support to authenticate and enforce access controls upon users.

Oracle Application Server supports JAAS with OracleAS JAAS Provider. Oracle's implementation of JAAS provides core security services for developing Java-based applications with Oracle Application Server. The JAAS provider enables security for OC4J to enforce security constraints for Web servlets, JavaServer Pages (JSP), and Enterprise JavaBeans (EJB) components.

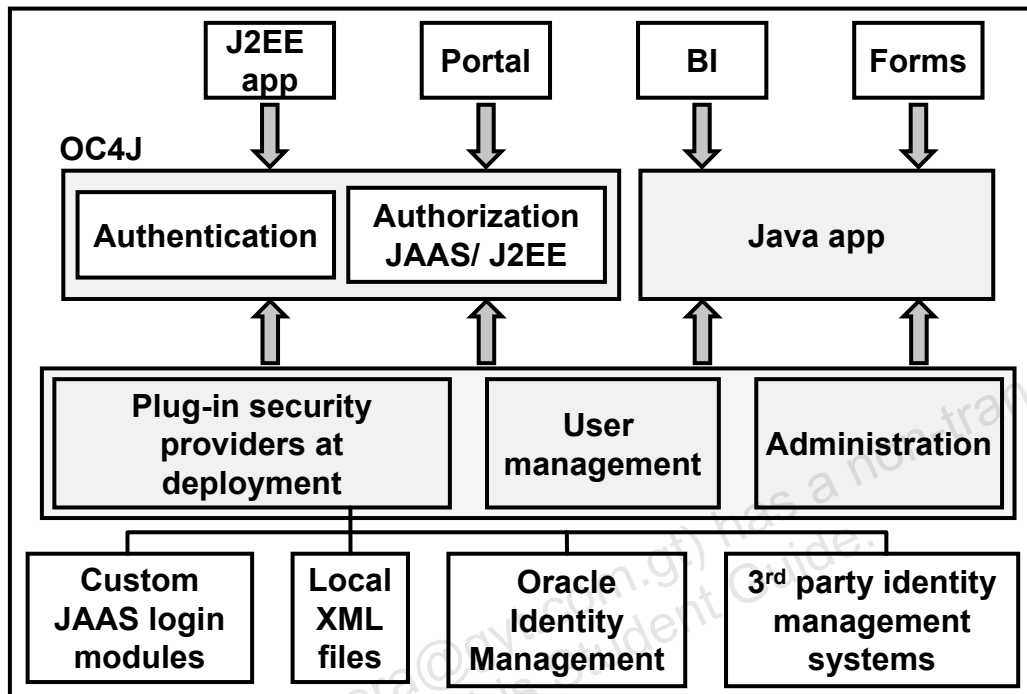
J2EE Security and JAAS (continued)

Oracle JAAS implementation supports two different provider types. The provider types implement a repository for secure, centralized storage, retrieval, and administration of provider data. The provider data consists of realm (users and roles) and JAAS policy (permissions) information. The provider types are:

- **XML-based provider:** Used for lightweight storage of information in XML files. The XML-based provider stores user, realm, and policy information in an XML file.
- **LDAP-based provider:** Based on Lightweight Directory Access Protocol (LDAP) for centralized storage of information in the LDAP-based Oracle Internet Directory

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

J2EE Security and JAAS



ORACLE

Copyright © 2005, Oracle. All rights reserved.

J2EE Security and JAAS (continued)

Oracle Application Server JAAS provides key security services in the following areas:

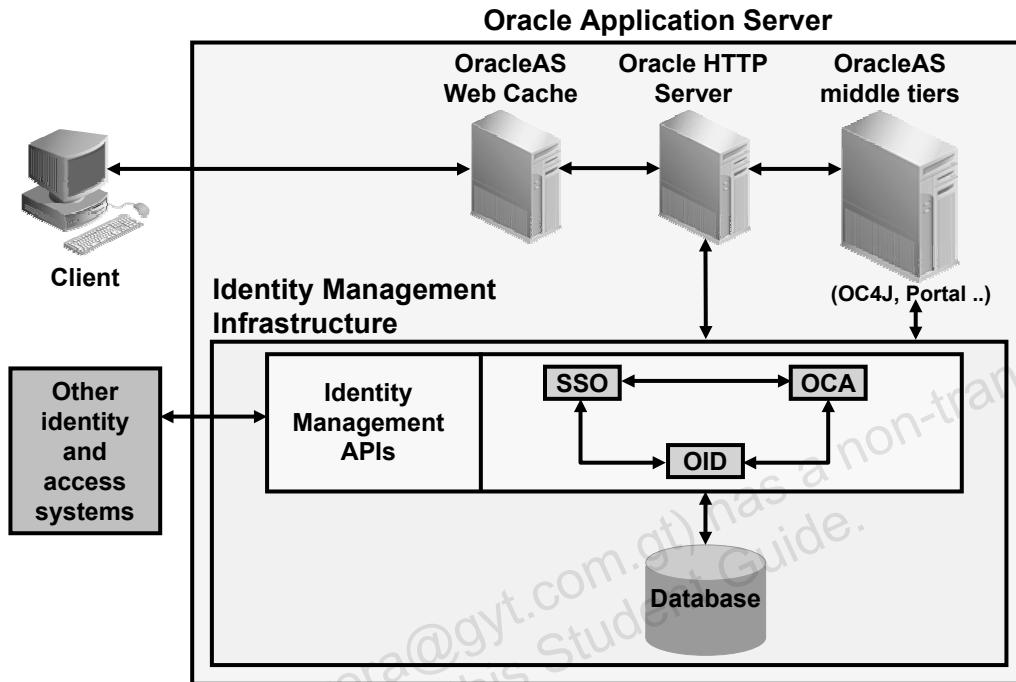
- Authentication to identify users
- Authorization to limit what users can do
- Delegation to enable code to run securely, with privileges of other users

User information is managed in a flat file in XML format for deployments that do not use Oracle Identity Management. For Oracle Application Server deployments in which Oracle Identity Management is installed, OracleAS JAAS can also leverage Oracle Identity Management. In the second type of deployment, user authentication and authorization information is managed in Oracle Internet Directory. OracleAS JAAS Provider can leverage Oracle Application Server Single Sign-On for user authentication. In addition, users can be provisioned using the Oracle Delegated Administration Services component of Oracle Application Server.

The benefits of the second deployment type include the following:

- Common framework for user authentication and authorization
- Easy integration with other Oracle products
- Support for user information management in a secure, highly available, reliable directory

Oracle Identity Management Security Solution



Copyright © 2005, Oracle. All rights reserved.

Oracle Identity Management Security Solution

Oracle Identity Management provides an important security solution for Oracle Application Server 10g. Each component of Oracle Identity Management is involved in providing important security features.

Oracle Internet Directory

Oracle Internet Directory is a key component that facilitates centralized user management. Users are defined centrally in Oracle Internet Directory. All Oracle Identity Management components as well as applications share this definition of user identity, credentials, profiles, and preferences.

Delegated Administration and Self-Service Interfaces

While centrally managing user identities, you can use Oracle Delegated Administration Services to delegate administrative functions to different administrators. Using a self-service interface, end users can update and reset their passwords, and manage preferences and profiles. A directory administrator can use the self-service interface to create and manage users and groups, customize user and group management interfaces, and customize end user self-service interface characteristics.

Oracle Identity Management Security Solution (continued)

OracleAS Single Sign-On

When you implement OracleAS Single Sign-On, you can log in to the Oracle Application Server environment as well as to other Web applications by using a single username and password. The single sign-on server consists of program logic in the Oracle Application Server database, Oracle HTTP Server, and OC4J server that enables you to log in securely to applications. There are two kinds of applications that can be authenticated through OracleAS Single Sign-On: Oracle Application Server applications (also called partner applications) and external applications. `mod_osso` is an Oracle HTTP Server module that provides integration with the single sign-on server to authenticate users.

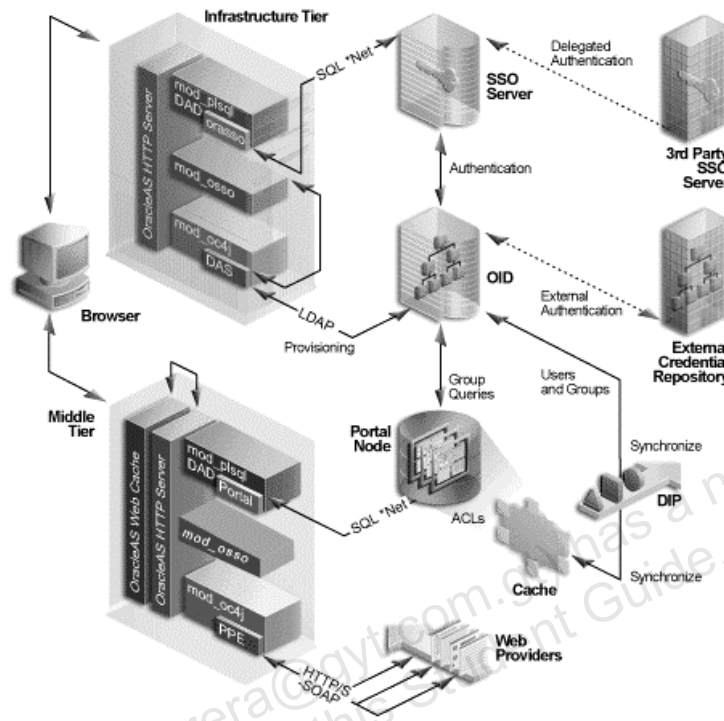
Secure Sockets Layer (SSL) and Certificates

The secure sockets layer (SSL) protocol is a standard for network transport layer security. SSL provides secure communication between client and server by allowing authentication, use of digital signatures for integrity, and encryption for privacy.

A digital certificate is an electronic means of establishing credentials when transacting on the Web. Certificates issued by authorized certificate authorities contain name, serial number, expiration dates, a copy of the certificate holder's public key (key for encrypting and decrypting messages and digital signatures), and the digital signature of the certificate-issuing authority. These certificates for the SSO users can be stored in OID.

A Certificate Authority (CA) is a third-party authority that issues, renews, and revokes authentication or digital certificates. An example of a CA is Verisign. Oracle Application Server provides OracleAS Certificate Authority (OCA) that allows you to create and manage certificates for use in Oracle software. OCA provides a simple, easy-to-use Web interface in which a user can submit a request online, provide authentication information, and acquire a certificate. As part of the OCA architecture, a server administrator can also use Oracle Wallet Manager to create, acquire, use, and store certificates. Wallet owners use it to manage security credentials on clients.

OracleAS Portal Security Architecture



Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal Security Architecture

OracleAS Portal provides a comprehensive security model that enables you to control what users can see and change on your Web site. The following components play an important role in implementing OracleAS Portal security model:

- Oracle Application Server Single Sign-On authenticates users.
- mod_osso redirects authentication requests to OracleAS Single Sign-On and keeps track of user activity in partner applications.
- OracleAS Web Cache serves pages generated by OracleAS Portal.
- Oracle Internet Directory is the storage repository for user credentials and group memberships.
- Oracle Delegated Administration Services adds or updates the information stored inside the directory.

Before users log in to OracleAS Portal, they can view only the public content such as public portlets. When a user first attempts to log in, if the current user is not authenticated with the OracleAS Single Sign-On environment, the user is challenged for a username and password. This information is redirected to OracleAS Single Sign-On for authentication. This, in turn, verifies the user credentials against those in Oracle Internet Directory.

OracleAS Portal Security Architecture (continued)

On successful authentication, OracleAS Single Sign-On creates a single sign-on session cookie. After the session is created, it is important to determine the pages and objects for which the user has the necessary access privileges. The Access Control Lists for all portal objects are stored in the Portal Schema in OracleAS Metadata Repository. When a user first logs in to OracleAS Portal, the user and group information is read from the directory and cached in the same repository as the Access Control Lists.

Oracle Delegated Administration Services generates a user interface to allow direct access to Oracle Internet Directory. This simplifies the provisioning of users and groups in Oracle Internet Directory for use in the portal.

OracleAS Portal thus leverages the components of Oracle Application Server and Oracle Database 10g to provide strong protection for your portal.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Summary

In this lesson, you should have learned how to:

- **List the risks in an Internet environment**
- **Describe the available security services**
- **Describe the Oracle Application Server security architecture**
- **Explain the role of individual Oracle Application Server components in the security architecture**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

12

Configuring Oracle Application Server Components in Oracle Internet Directory

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

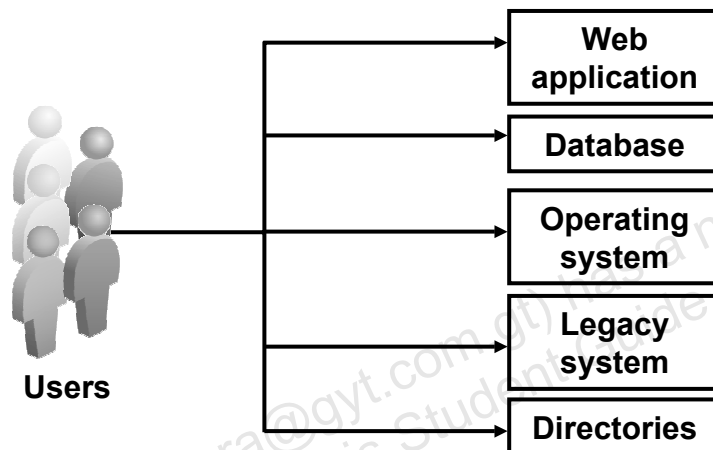
- **Start and stop Oracle Internet Directory processes**
- **Perform bulk data operation on the Oracle Internet Directory server**
- **Use Oracle Directory Manager for managing entries in the directory**
- **Manage users and groups**
- **Identify OracleAS Portal entries in the directory**
- **Configure Oracle Internet Directory settings in OracleAS Portal**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Identity Management: Overview

Identity management refers to managing the set of processes and strategies by which users are created and managed in the enterprise application environment.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Identity Management: Overview

Identity management is the process by which the complete security life cycle for network entities is managed in an organization. The network entities managed include users, devices, processes, applications, or anything else that needs to interact in a networked environment. Entities managed by an identity management process may also include users outside of the organization—for example, customers, trading partners, or Web services.

Using identity management, you can:

- Provision users for enterprise applications
- Manage user roles and permissions in applications
- Manage profile information (such as application preferences, passwords, and PINs)
- Personalize applications (such as Portals) for individual users

Benefits of Identity Management

- **For administrators:**
 - Lower costs of user administration
 - Improved user provisioning
 - Better security through centralized management of security policies and authorizations
 - Scalable administration through delegation
- **For users:**
 - Improved productivity through quicker access to applications
 - Improved usability with single-user identity and credentials, and application personalization

ORACLE

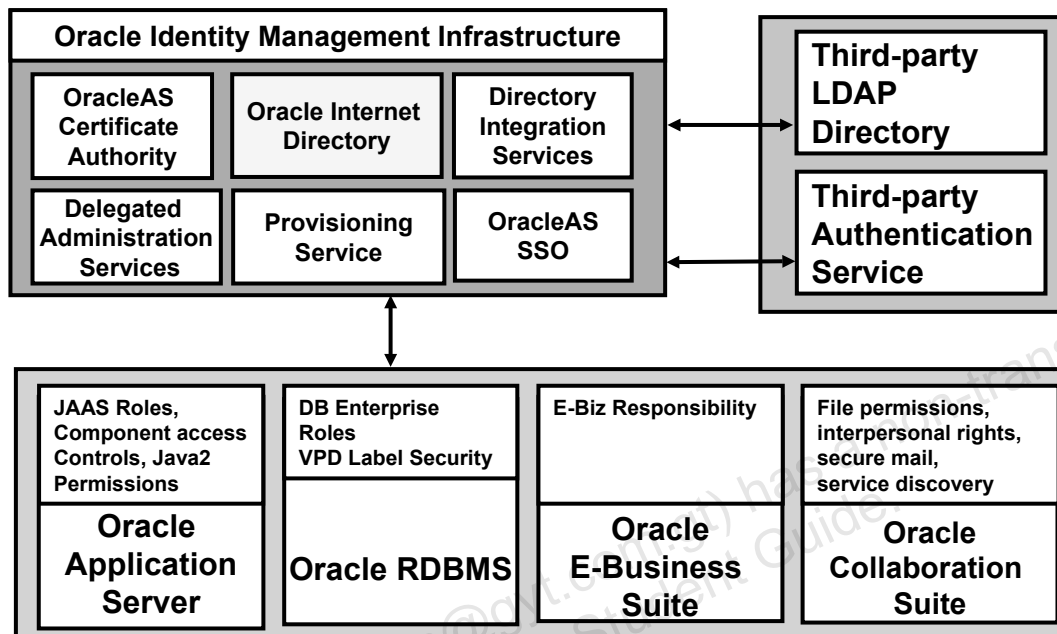
Copyright © 2005, Oracle. All rights reserved.

Benefits of Identity Management

Identity management reduces administrative costs that are associated with application deployments, while at the same time improving security. For example:

- For most enterprises, application user administration is a very expensive, laborious, and error-prone process. Identity management centralizes and automates much of these tasks, reducing administration costs while improving accuracy and security.
- An identity management strategy allows new users to gain access to their applications quickly, thereby eliminating wastage of employee time. Furthermore, identity management enables a customized application experience.
- An identity management strategy allows users to have their passwords and security credentials managed centrally. This improves usability while reducing the temptation for users to write this information in a handy place (a very insecure practice).

Oracle Identity Management



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Identity Management

Oracle Identity Management Infrastructure enables you to deploy enterprise identities and their access to various applications in the enterprise, centrally and securely. You can perform the following tasks using Oracle Identity Management:

- Creating enterprise identities and manage shared properties of these identities through a single enterprisewide console
- Creating groups of enterprise identities
- Provisioning events such as account creation, account suspension, and account deletion
- Managing policies that are associated with identities, such as authorization policies, authentication policies, and delegation of privileges to existing identities

The identity management functionality and the corresponding Oracle component that implements identity management are as follows:

1. Lightweight Directory Access Protocol (LDAP) directory service > Oracle Internet Directory
2. Directory integration > Oracle Internet Directory Integration Service
3. Application user provisioning > Oracle Internet Directory Provisioning Integration Service

Oracle Identity Management (continued)

4. Delegated administration > Oracle Delegated Administration Service
5. Web single sign-on > OracleAS Single Sign-On
6. Certificate Authority > OracleAS Certificate Authority

The centerpiece of Oracle Identity Management is Oracle Internet Directory, an LDAP v3-compliant directory service implemented on Oracle Database 10g. Oracle Identity Management is designed to support Oracle Database 10g, Oracle Application Server, Oracle Collaboration Suite, and Oracle E-Business Suite, as well as other enterprise applications and services.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Oracle Internet Directory

- **Oracle Internet Directory is Oracle's implementation of LDAP v3-compliant directory service.**
- **Oracle Internet Directory provides directory services to the Oracle database and Oracle Application Server.**
- **Oracle Internet Directory can support millions of entries and thousands of concurrent client accesses on a single directory node.**
- **Oracle Internet Directory implements sophisticated security management with a robust security model for protecting data from unauthorized access by LDAP clients.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Internet Directory

Oracle Internet Directory is Oracle's implementation of LDAP v3-compliant directory service. As Web applications and thin-clients become increasingly popular, you can use the Oracle Internet Directory server to store lightweight and sparingly updatable data. This data can be accessed by any LDAP-enabled application instantly and at a very high speed.

Oracle Internet Directory is not a security product, but rather a technology for managing enterprise data—including security data, such as usernames and passwords—for the Oracle 10g product stack.

The Oracle Internet Directory server is highly scalable and can support a large number of entries and concurrent users from a single node. Because Oracle Internet Directory is implemented on Oracle Database 10g, it takes advantage of database scalability. Oracle Internet Directory can also take advantage of Oracle Real Application Clusters to make it more scalable and more highly available.

Oracle Internet Directory is a secure platform for managing directory information. It implements the following three levels of directory user authentication:

- Anonymous
- Password-based
- Certificate-based through secure sockets layer (SSL)

Oracle Application Server 10g R2: Administration I 12-7

Security Benefits of Oracle Internet Directory

Oracle Internet Directory provides the following security benefits:

- **Data integrity**
- **Data confidentiality**
- **Password protection**
- **Data access control**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Security Benefits of Oracle Internet Directory

Oracle Internet Directory provides the following security benefits:

Security benefit	Description
Data integrity	Oracle Internet Directory uses SSL to ensure that data has not been modified, deleted, or replayed during transmission. SSL generates a cryptographically secure message digest, through cryptographic checksums, and includes it with each packet sent across the network.
Data confidentiality	Oracle Internet Directory ensures that data is protected against undesired disclosure during transmission by using encryption available with SSL.
Password protection	To protect passwords, Oracle Internet Directory uses the MD4 algorithm as the default. MD4 is a one-way hash function that produces a 128-bit hash, or message digest.
Data access control	Oracle Internet Directory supports access control down to the attribute level for read, write, or update of attributes.

Oracle Application Server Components and Oracle Internet Directory

Oracle Internet Directory enables Oracle Application Server components to:

- **Maintain single-user identity**
- **Store and manage the configuration information**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Components and Oracle Internet Directory

Oracle Application Server brings together all the Web-based applications developed using various application development products of Oracle, such as Oracle Developer, OracleBI Discoverer, and Oracle JDeveloper. You can deploy the applications developed by these products on Oracle Application Server and access them.

Oracle Application Server provides users access to these applications through OracleAS Single Sign-On (SSO). Through SSO, when you can log in once to Oracle Application Server, you are logged in to all the applications. This functionality can be achieved with the OracleAS SSO server by storing the user information and credentials in a single store, because user details are no longer stored in the application. Oracle Internet Directory provides a single store for all user information. Therefore, all the applications that are deployed on Oracle Application Server store their user information in Oracle Internet Directory.

Oracle Application Server Components and Oracle Internet Directory (continued)

Oracle Internet Directory enables the Oracle components to:

- **Maintain single-user identity:** You can store all the user information required by different applications in the Oracle Internet Directory server. This enables you to maintain a single identity for users. The user information can be administered from a single point in Oracle Internet Directory, which reduces development and administrative costs for the organization.
- **Store and manage the configuration information:** Along with user information, you can also store configuration information of different Oracle Application Server components and applications. Depending on the configuration information that is stored, you can grant administrative privileges on the Oracle Application Server components to different users.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Default Schema and Directory Information Tree (DIT)

- The Oracle Universal Installer (OUI) installs the default schema and directory information tree (DIT) for the Oracle directory-enabled products.
- The OUI installs the following DIT components:
 - Base schema elements
 - Root Oracle Context
 - Default Identity Management Realm
 - Identity Management Realm-Specific Oracle Context
 - Default password policy

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Default Schema and Directory Information Tree (DIT)

When you install Oracle Internet Directory, the Oracle Universal Installer (OUI) installs a default schema and directory information tree (DIT) for you to start using the Oracle components that use the directory. You can change the DIT as per your deployment requirements. During the installation, the OUI installs the following:

- **Base schema elements:** These elements are the basic attributes and object classes required by Oracle Internet Directory. Some of these are defined by the Internet Engineering Task Force (IETF) and others are specific to Oracle components.
- **Root Oracle Context:** In the Oracle Identity Management infrastructure, the Root Oracle Context is an entry in Oracle Internet Directory containing a pointer to the default Identity Management Realm in the infrastructure. It also contains information about how to locate an Identity Management Realm with a simple name of the realm.
- **Default Identity Management Realm:** After you install Oracle Application Server 10g, you have a default realm already configured and ready to use. This is the directory container that contains common information about all the Oracle components in the enterprise.

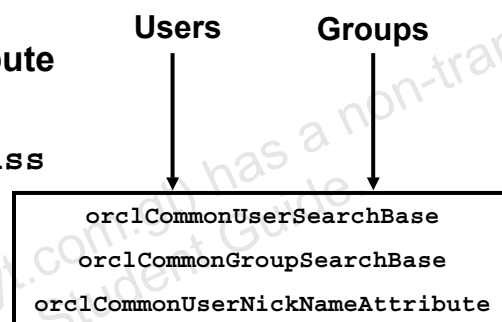
Default Schema and Directory Information Tree (DIT) (continued)

- **Identity Management Realm-Specific Oracle Context:** This is the container that contains the common information about all the Oracle components in the subscriber's subtree.
- **Default password policy:** This is a password policy for each subscriber and applies to all the users. For example, you may specify that passwords should be at least eight characters in length.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Identity Management Realm-Specific Common Entries

- Identity Management Realm-Specific common entries contain information for locating users and groups.
- Some of the attributes of the common entries are as follows:
 - User Search Base
 - User Nickname Attribute
 - Group Search Base
 - `orclUserObjectClass`



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Identity Management Realm-Specific Oracle Context's Common Entries

The common entry in the Identity Management Realm-Specific Oracle Context contains information for locating users and groups. Specifically, it includes the following:

- **User Search Base (`orclCommonUserSearchBase`):** Specifies the node in the subscriber DIT under which all the users are placed. This value is used while searching for a user in an Identity Management Realm.
- **User Nickname Attribute (`orclCommonUserNickNameAttribute`):** Specifies the nickname to be used when searching for a user under the user search base
- **Group Search Base (`orclCommonGroupSearchBase`):** Specifies the node under which you can find all the groups
- **`orclUserObjectClass`:** Lists the object classes to be used to create a user entry under the subscriber tree (for example, `person`, `organizationalPerson`, and `orclUser`)

Identity Management Realm-Specific Oracle Context's Common Entries (continued)

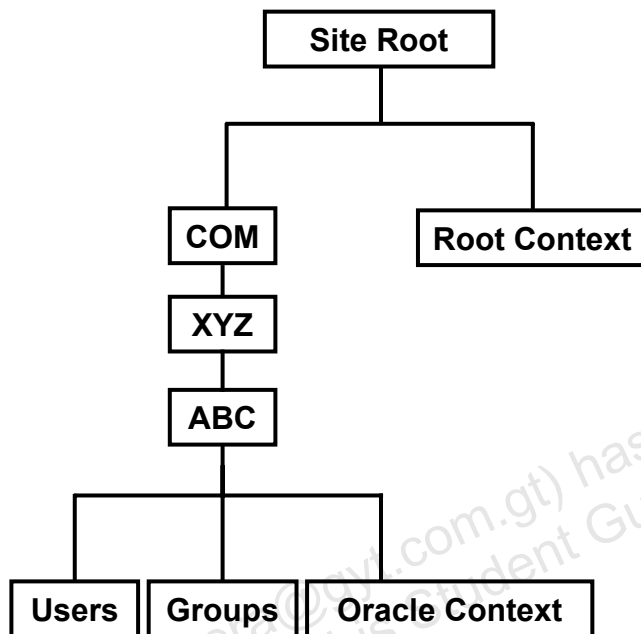
In a hosted environment, each Identity Management Realm can have its own instance of the application. In such a case, the instance information and other data required by the individual Identity Management Realm is stored in the Identity Management Realm-Specific Oracle Context. General information required by all Identity Management Realms is stored in the default/root Identity Management Realm-Specific Oracle Context.

You need not create the Identity Management Realm's user under the Identity Management Realm node. You can create and store users and their data in different ways:

- Directly under the Identity Management Realm node
- Outside the Identity Management Realm node: In this case, the `orclCommonUserSearchBase` attribute can specify the value to the node that contains the user and user data. This gives you the flexibility of having a new Identity Management Realm without migrating the old user distinguished names (DNs).

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Default Identity Management Realm Configuration



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Default Identity Management Realm Configuration

During the installation of Oracle Internet Directory, the OUI determines your domain information for the site. It creates the default DIT structure based on this information. For example, if the domain it identifies is `abc.xyz.com`, then it creates the default DIT as shown in the slide.

You can use the default DIT without configuring anything on the Root Oracle Context.

Starting and Stopping by Using OPMN

- **OPMN is responsible for monitoring Oracle Internet Directory as an Oracle Application Server component.**
- **Use OPMN to start and to stop oidmon:**

```
$ ./opmnctl startall
```

```
$ ./opmnctl startproc ias-component=OID
```

```
$ ./opmnctl stopall
```

```
$ ./opmnctl stopproc ias-component=OID
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting and Stopping by Using OPMN

OPMN enables you to manage Oracle Application Server components in an integrated way. If you use it to start an Oracle Internet Directory server, then you do not need to start OID Monitor separately or the directory-designated database. Instead, `opmnctl` starts those components for you. You can use `opmnctl` to do the following:

1. Start and stop a default or ready-to-use directory server instance.
2. On a specific node, stop, and then restart, all running Oracle Internet Directory servers such as directory servers, directory replication server, and directory integration and provisioning server.

To start all the Oracle Internet Directory server instances, use:

```
opmnctl startproc ias-component=OID
```

To stop all running Oracle Internet Directory server instances, use:

```
opmnctl stopproc ias-component=OID
```

When you start the Oracle Internet Directory component:

- OPMN issues an `oidmon start` command with appropriate arguments to `oidmon`
- OPMN issues `oidctl start` commands

Starting and Stopping by Using OPMN (continued)

Note: OPMN is responsible for monitoring Oracle Internet Directory as an Oracle Application Server component. OPMN knows only about `oidmon` and is unaware of the Oracle Internet Directory server instances. `oidmon` is a process that initiates, monitors, and terminates LDAP server processes. `oidmon` continues to be responsible for the direct starting, stopping, restarting, and monitoring of all Oracle Internet Directory server instances.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Oracle Internet Directory Server Processes

- You can connect to the Oracle Internet Directory server only if the Oracle Internet Directory server instance is running.
- To start the Oracle Internet Directory server, you must start the Oracle Internet Directory server processes in the following sequence:
 1. Start the `oidmon` utility.
 2. Start the server instances using the `oidctl` utility.
- You must stop the Oracle Internet Directory server by stopping the Oracle Internet Directory processes in the following sequence:
 1. Stop the server instance using the `oidctl` utility.
 2. Stop `oidmon`.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Internet Directory Server Processes

You can connect to an Oracle Internet Directory server only if the Oracle Internet Directory server instance is functional. You can start an Oracle Internet Directory server instance by starting the following Oracle Internet Directory server processes in the given sequence:

1. Start the `oidmon` utility.
2. Start an Oracle Internet Directory server instance process by using the `oidctl` utility.

You can start more than one Oracle Internet Directory server instance to accommodate the increasing number of users accessing the directory server. The new instances added must be started on different ports.

To shut down the Oracle Internet Directory server, you must first stop the Oracle Internet Directory server instance, followed by the `oidmon` process.

Note: You can use `opmnctl` to start Oracle Internet Directory as a component.

Starting and Stopping the `oidmon` Process

- The `oidmon` process must be running to process commands to start and stop the Oracle Internet Directory server instance using `oidctl` utility.
- To start `oidmon`:
 - Set `NLS_LANG` to a UTF8 appropriate language
 - Set the `TNS_CONNECT` string

```
oidmon connect=OID1 sleep=20 start
```

- You can stop the Oracle Internet Directory Monitor process by using the `oidmon` utility.

```
oidmon connect=OID1 stop
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting and Stopping the `oidmon` Process

You must start the `oidmon` process before starting any Oracle Internet Directory server instance. The `oidmon` process monitors all the Oracle Internet Directory server instances that are running and tries to recover them in case any of the server instances fail.

Syntax for Starting the `oidmon` Process

```
oidmon [connect=net_service_name] [sleep=seconds] start
```

where:

- `connect=net_service_name` specifies the net service name of the database to which you want to connect, that is, the database where OracleAS Infrastructure is installed. This network service name is set in the `tnsnames.ora` file. This is an optional argument.
- `sleep=seconds` specifies the number of seconds after which `oidmon` should check for new requests from `oidctl` and request to restart any servers that may have stopped. The default sleep time is 10 seconds. This argument is optional.
- `start` starts the `oidmon` process

Example

In the example, an `oidmon` process is started that connects to the database with the network service name of `OID1` and checks for new `oidctl` requests every 20 seconds.

Starting and Stopping the oidmon Process (continued)

Syntax for Stopping the OID Monitor Process

```
oidmon [connect=net_service_name] stop
```

where:

- `connect=net_service_name` specifies the net service name of the database to which you are connected. This is the network service name that is set in the `tnsnames.ora` file.
- `stop` stops the OID Monitor process.

Example

The example in the slide stops the OID Monitor process that is connected to the OID1 database.

Do not stop OID Monitor if you are simply ending one or more directory server or replication server instances, but only if you are shutting down the LDAP service altogether.

For additional information, refer to the *Oracle Internet Directory Administrator's Guide*.

Starting and Stopping an Oracle Internet Directory Server Instance

- You can start or stop an Oracle Internet Directory server instance only if the `oidmon` process is running.
- Use the `oidctl` utility to start or stop the Oracle Internet Directory server instance.

```
oidctl connect=OID1 server=oidldapd instance=2  
configset=3  
flags='-p 3062 -debug 1024 -l'  
start
```

```
oidctl connect=OID1 server=oidldapd instance=2  
stop
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting and Stopping an Oracle Internet Directory Server Instance

You use `oidctl` utility to start the Oracle Internet Directory server instance. You must ensure that `oidmon` is already running.

Syntax for Starting the Oracle Internet Directory Server Process

```
oidctl connect="net_service_name" server=oidldapd  
instance=server_instance_number [configset=configset_number]  
[flags=' -p port_number start
```

where:

- `connect=net_service_name` is the net service name specified in that file located in `ORACLE_HOME/network/admin`, if you already have a `tnsnames.ora` file configured. It points to the OracleAS Infrastructure database.
- `server=oidldapd` is the type of server to start (valid values are `OIDLDAPD`, `OIDREPLD`, and `ODISRV`). This is not case sensitive.
- `instance=server_instance_number` is the instance number of the server to start. You should specify a number between 1 and 1,000.

Starting and Stopping an Oracle Internet Directory Server Instance (continued)

- `configset=configset_number` is the `configset` number used to start the server. This defaults to `configset0`, if not set. This should be a number between 0 and 1,000.

The configuration parameters for each Oracle directory server instance are stored in a directory entry called a configuration set entry, or `configset`. Configuration set entries contain the server instance parameters, which are used by the `oidctl` utility to start the server instance. The default configuration set entry is `configset0`. You can create or modify a configuration set entry by using Oracle Directory Manager.

- `-p port_number` specifies a port number during server instance startup. The default port number is 389.
- `start` starts the server specified in the server argument

The example shown in the slide starts a directory server instance whose net service name is `OID1`, using `configset3`, at port 3062, with a debug level of 1024, and an instance number 2.

Syntax for Stopping the Oracle Internet Directory Server Instance

```
oidctl connect=net_service_name server=oidldapd  
instance=server_instance_number stop
```

where:

- `connect=net_service_name` is the net service name associated with the Oracle Internet Directory instances
- `server=oidldapd` is the type of server to start (valid values are `OIDLDAPD`, `OIDREPLD`, and `ODISRV`). This parameter value is not case sensitive.
- `instance=server_instance_number` is the instance number of the server to start. This should be the number of an existing running instance of the type specified.
- `stop` stops the server specified in the server argument

Example

The example in the slide stops instance 2 that is connected to the `OID1` database.

You can use the `oidctl` command to stop a running instance of the Oracle Internet Directory server. You need to specify the number of the Oracle Internet Directory server instance that you want to stop. Before you stop the instance, you must ensure that the OID Monitor process is running.

Using Bulk Tools

You can use the following bulk tools to perform bulk data operations on the Oracle Internet Directory server:

- **bulkload**
- **ldifwrite**
- **bulkmodify**
- **bulkdelete**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using Bulk Tools

Use the bulk tools to maintain a large number of directory entries. These entries commonly come from other applications or directories. To use bulk tools in the Windows environment, you must install UNIX shell utilities such as Cygnus and MKS. Bulk tools include the following:

- **bulkload:** Loads a large number of entries to the Oracle Internet Directory server using an LDIF file as input. These LDIF files may be generated or extracted from third-party applications using RFC 2849 (www.ietf.org/rfc/rfc2849.txt) as a guide.
- **ldifwrite:** Copies data from the Oracle Internet Directory information base into an LDIF file that can be read by any LDAP-compliant directory server. Use this LDIF file to transfer data between the directory servers. You can use **ldifwrite** in conjunction with **bulkload**. You can also use **ldifwrite** to back up information from all or parts of a directory.
- **bulkmodify:** Modifies a large number of existing entries efficiently. You can change attributes common to multiple entries simultaneously, including adding a new attribute value and replacing existing values across a set of entries you specify with a simple filter.
- **bulkdelete:** Deletes a subtree efficiently

Note: For details, refer to the *Introduction to LDAP and Oracle Internet Directory* eStudy.

Using LDAP Command-Line Tools

You can create and modify the data stored in the Oracle Internet Directory server by using the following commands:

LDAP Command-Line Tools	Example
<code>ldapadd</code>	<code>ldapadd -p 4032 -h edrsr25p1 cn=orcladmin -w welcome -f newentry.ldif</code>
<code>ldapbind</code>	<code>ldapbind [arguments]</code>
<code>ldapsearch</code>	<code>ldapsearch -p 4032 -h edrsr25p1 -b "" -s base -v "objectclass=*"</code>

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using LDAP Command-Line Tools

You can manipulate the entries and attributes in the Oracle Internet Directory server by using Oracle Directory Manager or command-line tools. These commands operate on directory objects as specified on standard I/O, or by using a text file written in LDAP Data Interchange Format (LDIF) as input.

The commands are as follows:

- **ldapadd:** Used to add one or more entries from standard I/O or an LDIF file
- **ldapaddmt:** Used for adding entries concurrently by using multiple threads
- **ldapbind:** Used to authenticate a user to the directory server
- **ldapcompare:** Used to find whether an entry contains a specific attribute value
- **ldapdelete:** Used to delete an entry
- **ldapmoddn:** Used to modify the DN and RDN of an entry, rename an entry or a subtree, or move an entry or subtree to a new parent
- **ldapmodify:** Used to create, update, and delete data in an entry
- **ldapmodifymt:** Used to modify multiple entries at a time by using multithreading
- **ldapsearch:** Used to search for an entry in the directory

Note: For more information, refer to the *Introduction to LDAP and Oracle Internet Directory* eStudy.

Using Oracle Directory Manager

- **Oracle Directory Manager is a Java-based graphical user interface tool to maintain and administer Oracle Internet Directory data.**
- **You can use Oracle Directory Manager for the following tasks:**
 - **Searching, viewing, and maintaining object classes**
 - **Searching and maintaining an attribute**
 - **Creating and dropping an index on an attribute**
 - **Searching, viewing, and maintaining an entry**
 - **Controlling access to Oracle Internet Directory entries**
 - **Managing a replication node**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using Oracle Directory Manager

Oracle Internet Directory provides both graphical user interface and command-line interface to manage data stored in the server. Oracle Directory Manager is a graphical user interface in Java, used to manage object classes, attributes, and entries. You can connect to multiple Oracle Internet Directory servers simultaneously and manage them.

Oracle Directory Manager cannot be used for:

- Starting and stopping the directory monitor process
- Starting and stopping the directory server instances
- Starting and stopping the directory replication server instances

Starting Oracle Directory Manager

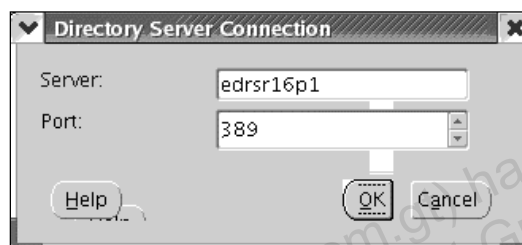
To start the Oracle Directory Manager tool, perform the following steps:

- **In Windows platform:** From the Start menu, select Programs > Oracle-Oracle Internet Directory_Home > Integrated Management Tools > Oracle Directory Manager.
- **In Linux/UNIX platforms:** Change to the bin directory under the corresponding Oracle Home directory. Enter `oidadmin` at the command prompt.

Connecting to the Oracle Internet Directory Server

To connect to an Oracle Internet Directory server, you must specify:

- Oracle Internet Directory server host name
- Oracle Internet Directory server port



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Connecting to the Oracle Internet Directory Server

When you start Oracle Directory Manager the first time, an alert appears indicating that you must select a server and the port where the Oracle Internet Directory server instance is running, and then connect to Oracle Directory Manager.

As shown in the slide, when you click OK, a dialog box is displayed to add a new server (host name or IP address), where the Oracle Internet Directory instance is running. The fields displayed in the dialog box are:

- **Server:** The server name (host name or IP address) where the Oracle Internet Directory instance is running
- **Port:** The port in which the instance is running. The default port for Oracle Internet Directory on Linux/UNIX platforms is normally 389. If the server is running on a different port, then enter that port number.

Oracle Directory Manager Connect Dialog Box



Oracle Directory Manager Connect Dialog Box

The Oracle Directory Manager Connect dialog box contains two tabs: Credentials and SSL. The fields in the dialog box are explained as follows:

- **User:** You can log in as a superuser or an anonymous user the first time you log in. To log in as a superuser, specify `orcladmin` as the username and `welcome1` as the password. To log in as an anonymous user, leave the user and the password fields empty. If you have set up the user entries, you can log in in the following two ways:
 1. Browse and select an entry by clicking the button to the right of the User field.
 2. Enter the DN for the user's entry, for example,
`cn=Jane, ou=st, ou=acme, c=us.`
- **Password:** Enter the corresponding password for the user entered in the user field. For the `orcladmin` superuser, the password is the password that you had specified during the installation of OracleAS Infrastructure for `ias_admin`. For an anonymous user, leave this field blank.
- **Server:** From the drop-down list, select the name of the server where the Oracle Internet Directory server instance is running. Click the button to the right of the server field to see whether the Oracle Internet Directory instance is currently running on the server. Click the Add button to add a new server that is running an Oracle Internet Directory server instance.

Oracle Directory Manager Connect Dialog Box (continued)

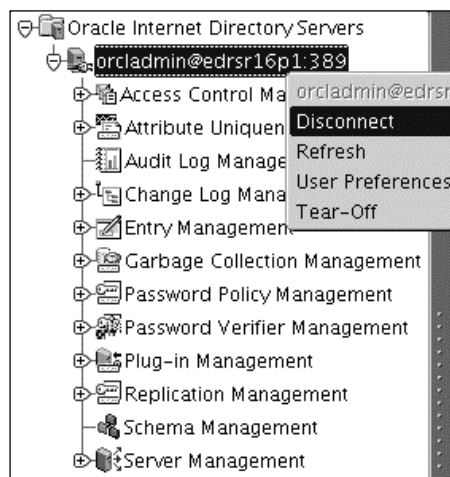
- **Port:** The default port is 389 or 3060. The value of the port is set automatically when you select the server, because the port is configured with the server.
- **SSL Enabled:** Select this check box if you want the communication between the client and the server to be over the secure sockets layer (SSL). To connect with SSL, the server should listen to an SSL-enabled port, otherwise the request will not be authenticated. If you have selected the SSL Enabled check box, then you must enter data in the fields on the SSL tabbed page.

SSL Tabbed Page

You can connect to the Oracle Internet Directory server through a secure connection using SSL. In an SSL connection, the data is transmitted in an encrypted format between the client and the server. To connect through SSL, you must select the SSL Enabled check box on the Credentials tabbed page and enter information on the SSL tabbed page.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Disconnecting from the Oracle Internet Directory Server



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Disconnecting from the Oracle Internet Directory Server

To disconnect from a directory server, perform one of the following:

- Select Disconnect from the File menu.
- Disconnect from the toolbar.
- Right-click the Oracle Internet Directory server and select Disconnect.

When you exit Oracle Directory Manager, connections between all directory servers and the directory are automatically disconnected. When you restart Oracle Directory Manager, all previously connected server connections appear in the Directory Server Login window.

Oracle Application Server Bootstrap Model

The Oracle Internet Directory installation creates the following set of users to facilitate Oracle Application Server deployment bootstrap:

- **Oracle Internet Directory superuser**
(ou=orcladmin)
- **Oracle Internet Directory enterprise subscriber superuser**
(cn=orcladmin,cn=users,<Subscriber DN>)

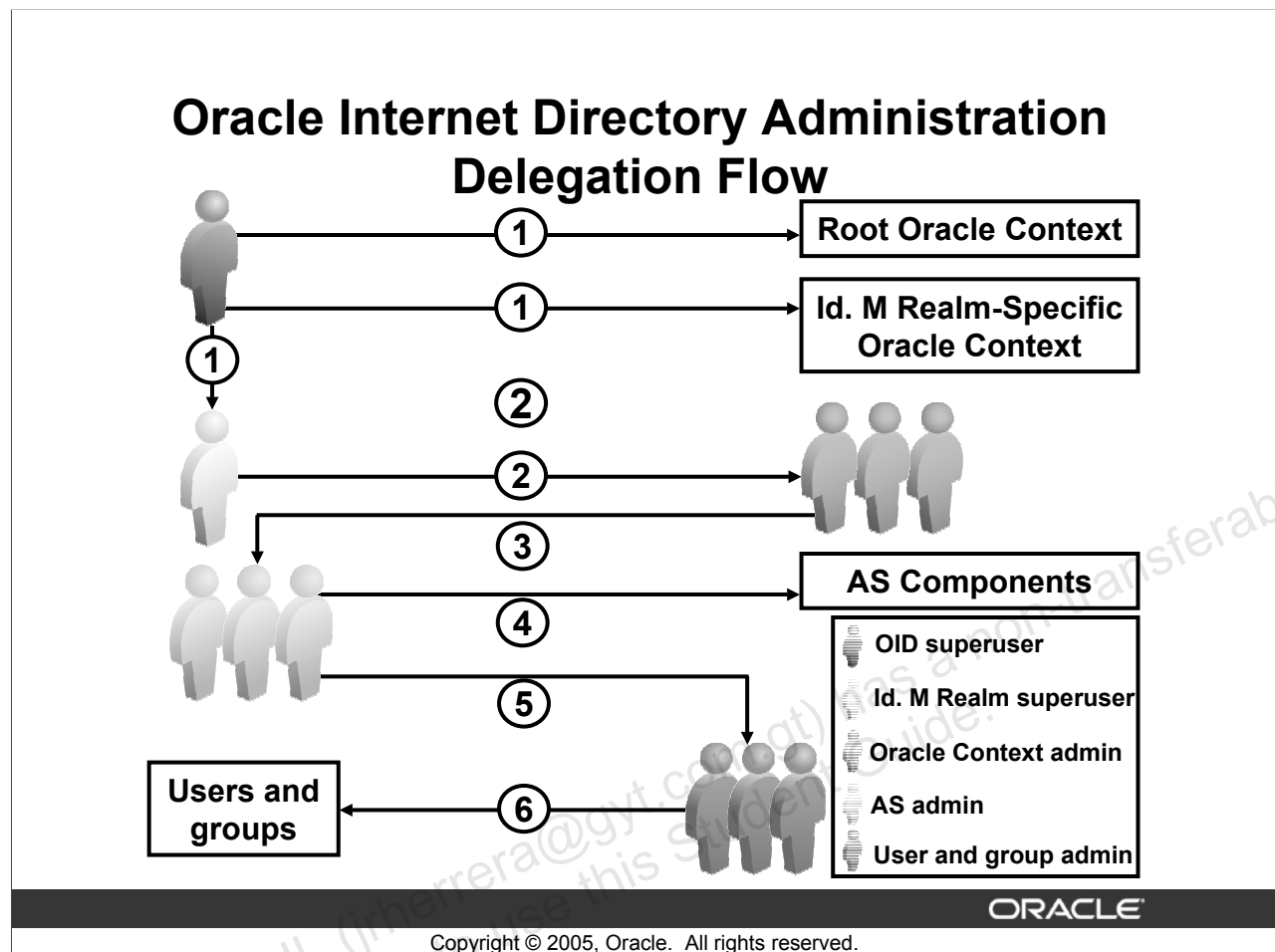
ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Bootstrap Model

To facilitate bootstrap of an Oracle Application Server deployment, Oracle Internet Directory installation creates the following users:

- Oracle Internet Directory superuser (cn=orcladmin): This user is allowed to perform all operations in the directory. No Access Control List (ACL) policy can restrict access to the directory for this user.
Note: This user cannot log in through the SSO server. Therefore, the superuser typically uses the Oracle Directory Manager (ODM) tool to perform various administrative actions on Oracle Internet Directory.
- A superuser for the enterprise is identified as cn=orcladmin, cn=users, <Subscriber DN>. The default ACL set up during the installation of a new Oracle Internet Directory enables this user to perform all administrative operations within the enterprise subtree. This user can create new users by using Delegated Administration Services (DAS), assign them privileges, and also delegate the privilege assignment to the newly created users.



Oracle Internet Directory Administration Delegation Flow

The graphic in the slide displays how the administration rights are delegated to various levels of users. The Oracle Internet Directory Administration delegation involves the following steps:

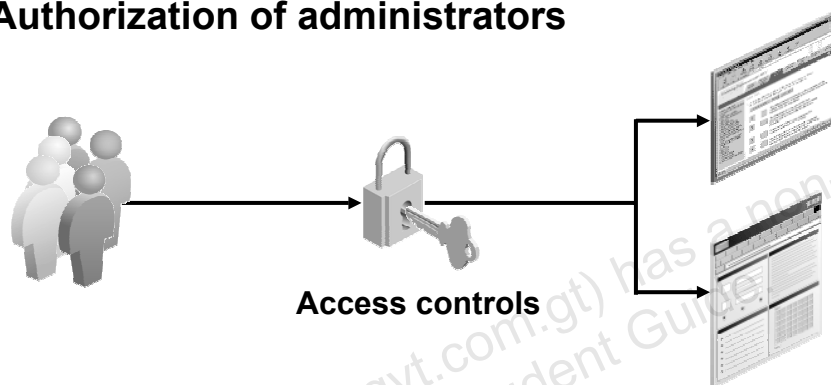
1. The Oracle Internet Directory superuser creates the Default DIT as part of Oracle Internet Directory installation (Default DIT includes creation of Root Oracle Context, a node for the enterprise and an Oracle Context associated with the enterprise node). Also, as part of the default DIT, the subscriber superuser is created.
2. The subscriber superuser delegates the administration of Oracle Context to Oracle Context administrators.
3. The Oracle Context administrators delegate the administration of Oracle Application Server and its components to Oracle Application Server administrators.
4. Oracle Application Server administrators have the necessary privileges to install and bootstrap all Oracle Application Server components.
5. Oracle Application Server administrators additionally delegate the responsibility of user and group administration to appropriate users.
6. The user and group administrators are responsible for managing users and groups.

For additional information, refer to *Oracle Internet Directory Administrator's Guide 10g Release 2*.

Delegated Directory Administration

You can implement access control using Oracle Internet Directory at two levels:

- **Authorization of users**
- **Authorization of administrators**



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Delegated Directory Administration

Oracle Internet Directory stores access control to different applications as LDAP attributes or entries. To give a user administrative access to a particular application, you can have access policies defined on these attributes or entries. In a hosted environment, you can enable access to a particular subscriber by giving access to that user on the subscriber root node. Similarly, you can give access to the departmental administrators for their departments in a nonhosted environment.

You can implement access control by using Oracle Internet Directory at two levels:

- **Authorization of users:** You can store the access control policies to access the external applications in Oracle Internet Directory. When the user performs an operation on the application data, the application checks whether the user is authorized to do such an operation or not from the access policies stored in Oracle Internet Directory.
- **Authorization of administrators:** You can use Oracle Internet Directory to serve as a single trusted point of administration for all the access control policies of the application. You can set up access controls for the access control policies of a specific application to decide who can administer these policies. If any user tries to make changes to any of the access control policies of the application, Oracle Internet Directory checks whether the user is authorized to make such a change.

Oracle Application Server 10g R2: Administration I 12-32

Directory Roles

Oracle Internet Directory can have the following roles associated with it:

- **Oracle Internet Directory global administrator**
- **Subscriber-specific or domain administrator**
- **Application-specific roles**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Directory Roles

Different domains in the Oracle Internet Directory server are managed by different administrators. There can be one or more administrators for a given domain. For every domain, there are directory roles that are responsible for the administration of the domain. The following are the roles:

- **Oracle Internet Directory global administrator:** The global administrator has privileges all over the Oracle Internet Directory server.
- **Subscriber-specific or domain administrator:** This administrator has privileges over this domain. These privileges are delegated to the administrator by the global administrator.
- **Application-specific:** Application administrators administer application data under a subscriber node. The application administrator is responsible for delegating rights to the users on the application data.

Oracle Application Server Administration Model

- **An Oracle Application Server administrator should be a member of the `iASAdmins` group in Oracle Internet Directory to configure various Oracle Application Server components.**
- **The DN of the `iASAdmins` group is `cn=iASAdmins,cn=Groups,<Oracle Context DN>`.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Administration Model

An Oracle Application Server administrator should be a member of the `iASAdmins` group in Oracle Internet Directory to configure the Oracle Application Server components entries in the Oracle Internet Directory server. A user should be a member of the `iASAdmins` group in Oracle Internet Directory to administer various Oracle Application Server components individually.

In a stand-alone model, where Oracle Internet Directory is used only for Oracle Application Server, the Oracle Application Server administrator bootstraps the Oracle Application Server environment by using the seed accounts (`cn=orcladmin`) that are set up as part of the Oracle Internet Directory installation. Having installed various products, an Oracle Application Server administrator can create users by using tools, such as DAS, and delegate Oracle Application Server administration to other users by adding them to the `iASAdmins` group.

Oracle Application Server Administration Model (continued)

The administrative needs in shared mode, where Oracle Internet Directory is used not only for Oracle Application Server but also for other applications, may be quite different.

Therefore, to perform installations in such an environment, the Oracle Application Server administrator should seek privileges equivalent to the iASAdmins privileges from the Oracle Internet Directory administrator.

The iASAdmins group is created under the groups container in Oracle Context. Its DN is `cn=iASAdmins, cn=groups, <Oracle Context DN>`.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

User Administration

- **All Oracle Application Server users are represented as user objects in Oracle Internet Directory.**
- **The Oracle Application Server administrator can delegate user management to other users by adding them to:**
 - **The User Create group to delegate user creation**
 - **The User Edit group to delegate user edit**
 - **The User Delete group to delegate user deletion**
- **All these groups are created under the groups container of the Oracle Context.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

User Administration

Oracle Application Server users are represented as user objects in Oracle Internet Directory. The default Oracle Internet Directory configuration facilitates easy administration of users in Oracle Internet Directory.

In a stand-alone mode, the Oracle Application Server administrator uses the seed users (cn=orcladmin) to create users in Oracle Internet Directory. The Oracle Application Server administrator may choose to add some of these users to the User Create group to delegate user creation to these users. Similarly, to edit a user's properties, an Oracle Application Server administrator adds the user to the User Edit group to give them the user edit privilege, and User Delete group to give the user delete privilege.

The User Create group, the User Edit group, and the User Delete group are all created under the groups container in the Oracle Context. Their corresponding DNs are:

- **User Create:** cn=oracleDASCreateUser, cn=groups, <Oracle Context DN>
- **User Edit:** cn=oracleDASEditUser, cn=groups, <Oracle Context DN>
- **User Delete:** cn=oracleDASDeleteUser, cn=groups, <Oracle Context DN>

Group Administration

- **An Oracle Application Server administrator can delegate group management to other users by adding them to:**
 - **Group Create group to delegate group creation**
 - **Group Edit group to delegate group edit**
 - **Group Delete group to delegate group deletion**
- **All these groups are created under the groups node of the Oracle Context.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Group Administration

The Oracle Application Server administrator delegates the rights of managing groups to other users. The group management rights are delegated to the users by adding them as members in the following groups:

- Create Group to delegate rights to create groups
- Edit Group to delegate rights to edit the group properties
- Delete Group to delegate rights to delete groups

All the groups mentioned are stored under the groups node in the Oracle Context.

The DN of the groups are as follows:

- **Create Group:** cn=oracleDASCreateGroup, cn=groups, <Oracle Context DN>
- **Edit Group:** cn=oracleDASEditGroup, cn=groups, <Oracle Context DN>
- **Delete Group:** cn=oracleDASDeleteGroup, cn=groups, <Oracle Context DN>

Administrative Groups

- The Oracle Application Server components read user and group information from Oracle Internet Directory.
- Oracle Internet Directory enables this by granting privileges to various administrative groups.
- The administrative groups are as follows:
 - Authentication Services
 - Users Security Administration
 - User Proxy Privilege

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Administrative Groups

Some of the Oracle Application Server components read user- and group-related information from Oracle Internet Directory. The default ACL configuration in Oracle Internet Directory facilitates this by granting privileges to various administrative groups.

The groups are as follows:

- **Authentication Service:** Various products authenticate end users by comparing the presented password with the one stored in Oracle Internet Directory. For this purpose, the individual product entity adds itself as a member of “Authentication Service,” which has compare permission on the user password attribute of a user. The DN for the group is `cn=authenticationServices,cn=groups,<Oracle Context DN>`.
- **User Security Admin:** Similar to authentication service, this group in Oracle Internet Directory has permissions to read, compare, and reset the user password of a user in Oracle Internet Directory. The DN for the group is `cn=oraclUserSecurityAdmins,cn=groups,<Oracle Context DN>`.

Administrative Groups (continued)

- **User Proxy Privilege:** Most of the Oracle products proxy to Oracle Internet Directory on behalf of their end users. To be able to proxy on behalf of an end user (whose identity is stored in Oracle Internet Directory), these products add their identity as a member of the User Proxy Privilege group that has permissions to proxy on behalf of the Oracle Internet Directory user. The DN for the group is `cn=userProxyPrivilege,cn=groups,<Oracle Context DN>`.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Storage of User Credentials

The user authentication credentials stored in the Oracle Internet Directory server are as follows:

- **Credentials for directory usage**
- **Credentials for authenticating a user to Oracle components**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Storage of User Credentials

In an enterprise, there are many applications, and each of them has its own authentication mechanism. In this scenario, it is difficult to manage the user information, especially the password. If a user leaves or changes jobs, all the authentication privileges in all the applications have to be removed or changed. This increases the administrative efforts and also the cost. To solve this, you can use the Oracle Internet Directory server as a centralized storage for user authentication credentials. Here, you can remove or change the authentication information of the user if the user leaves or changes jobs.

The Oracle Internet Directory server can store user authentication information of the following types:

- **Passwords to authenticate the user to Oracle Internet Directory:** You can store user information, such as username and password, in the Oracle Internet Directory server as an entry. The user can log in to Oracle Internet Directory using this username and password.
- **Password to authenticate the users to Oracle components:** You can store the usernames and passwords of the users of Oracle components. When the user logs in to the Oracle component, the username and password are verified with the ones that are stored in Oracle Internet Directory.

Storage of User Credentials (continued)

You can store the user credentials of the users for an LDAP-enabled application that is not an Oracle component. To store the user credentials for such an application, you must create a separate container under the products entry.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Modifying Password Policies by Using ODM

- Password policies are a set of rules that govern how the password is used.
- Each Identity Management Realm has its own password policy that is applicable for all users under that Identity Management Realm.
- You can modify the password policies by using ODM.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Modifying Password Policies by Using ODM

The passwords stored in the Oracle Internet Directory servers follow a set of rules. These rules are defined in an entry known as password policies. Password policies ensure that the password of a user fulfills certain criteria. These rules or criteria can be the following:

- Minimum number of characters in the password
- Number of numeric characters in the password
- Time for which the password is valid, and so on

The OUI creates the password policy entry for each subscriber at the time of installation. The entry is created below the common entry, which resides under the product entry, and the product entry is below the Identity Management Realm entry. This password policy is applicable to all the users in this Identity Management Realm. The password policy is applicable to the `userPassword` attribute of the user. You must set the appropriate value of `orclcommonusersearchbase` in the common entry of the subscriber to enforce the password policy.

To create a password policy, you use the `pwdpolicy` auxiliary object class. During the installation, a password policy entry is created by using this object class.

Modifying Password Policies by Using ODM (continued)

You can view and modify password policies by performing the following steps:

- In the navigation pane, expand Oracle Internet Directory Servers > directory_server_instance > password policy management. The password policy entries are displayed under this node.
- Select the password policy that you want to view. In the right pane, the attributes of the policy are displayed.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Modifying the Oracle Internet Directory Administrator Password

The screenshot shows a dialog box titled "Modifying the Oracle Internet Directory Administrator Password". It contains several input fields and "Browse" buttons. The fields are: "Super User Name" (containing "cn=orcladmin"), "Super User Password" (masked with asterisks), "Guest Login Name" (containing "cn=guest"), "Guest Login Password" (masked with asterisks), "Proxy Login Name" (containing "cn=proxy"), and "Proxy Login Password" (masked with asterisks). Each field has a "Browse" button to its right. The dialog box is part of a larger application window with tabs labeled "System Operational A...", "System P...", "Audit Ma...", "Debug Fl...", and "Query Op...".

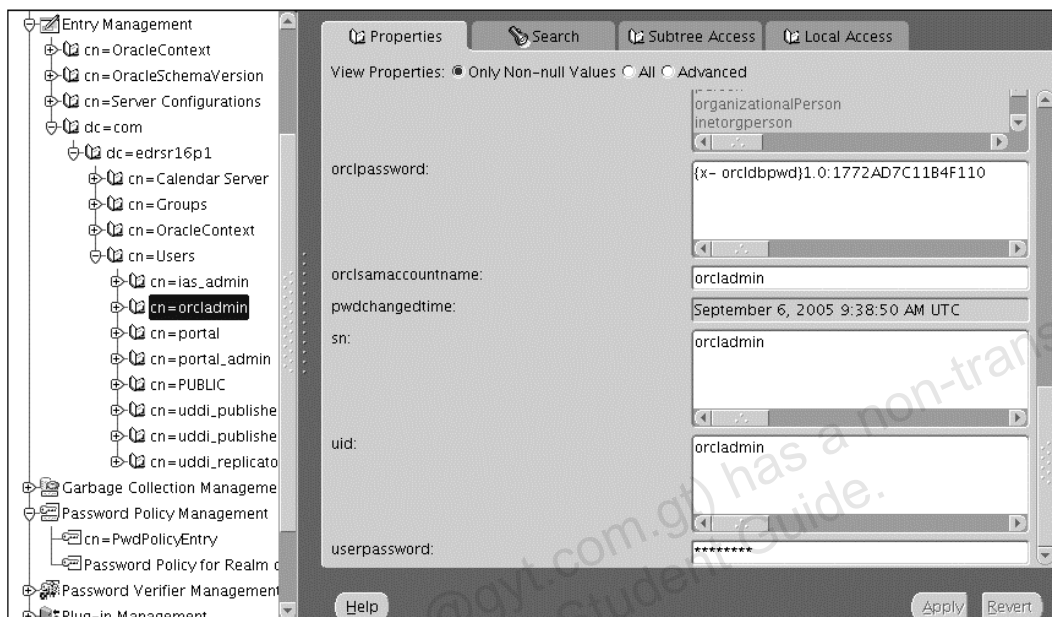
Modifying the Oracle Internet Directory Administrator Password

You can change the `orcladmin` password of the Oracle Internet Directory administrator by using ODM. To change the password, perform the following steps:

1. Log in to Oracle Internet Directory by using ODM.
2. Expand the Oracle Directory Servers node.
3. Select the Oracle Internet Directory server node. In the right pane, various properties of the server are displayed on tabbed pages.
4. Click the System Passwords tab. Various usernames and passwords are displayed.
5. In the Super User Password field, change the password to the desired value.
6. Click the Apply button to save the new password.

You can change the Oracle Internet Directory administrator password by using the Oracle Internet Directory Self-Service Console too.

Modifying the Realm-Specific Administrator Password



Copyright © 2005, Oracle. All rights reserved.

Modifying the Realm-Specific Administrator Password

You can use ODM to change the Oracle Application Server administrator password. Both the Oracle Internet Directory administrator and the Oracle Application Server administrators are `orcladmin`; however, both are located in a different hierarchy tree. The Oracle Internet Directory administrator is located at the starting node of the Oracle Internet Directory server and is responsible for managing the entire Oracle Internet Directory server and all included realms. The Oracle Application Server administrator is the superuser for a specific realm or subscriber. The Oracle Internet Directory administrator delegates administrative rights to various subscriber administrators.

To change the Oracle Application Server administrator password, perform the following steps:

1. Log in to Oracle Internet Directory by using ODM.
2. Expand the Oracle Directory Servers node > Entry Management > dc=com > dc=Oracle > dc=us > cn=Users.
3. Select the `cn=orcladmin` node. Various properties of the user are displayed in the right pane.
4. Scroll down in the right pane until the `userpassword` field. Change the password to a desired value, and click the Apply button.

Modifying the Realm-Specific Administrator Password (continued)

Use the same steps to change the password of any user in Oracle Application Server. You can change the Oracle Application Server administrator password using Oracle Internet Directory Self-Service Console too. To change the Oracle Application Server instance password, you must use the `restiaspasswd` command.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Relationship Between OracleAS Portal and Oracle Internet Directory

OracleAS Portal requires the following interaction with Oracle Internet Directory:

- **OracleAS Portal–specific entries stored in the directory**
- **Group attributes stored in the directory**
- **User attributes stored in the directory**
- **Caching of user and group information from the directory**
- **Populating of user and group list of values from the directory through Delegated Administration Services**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

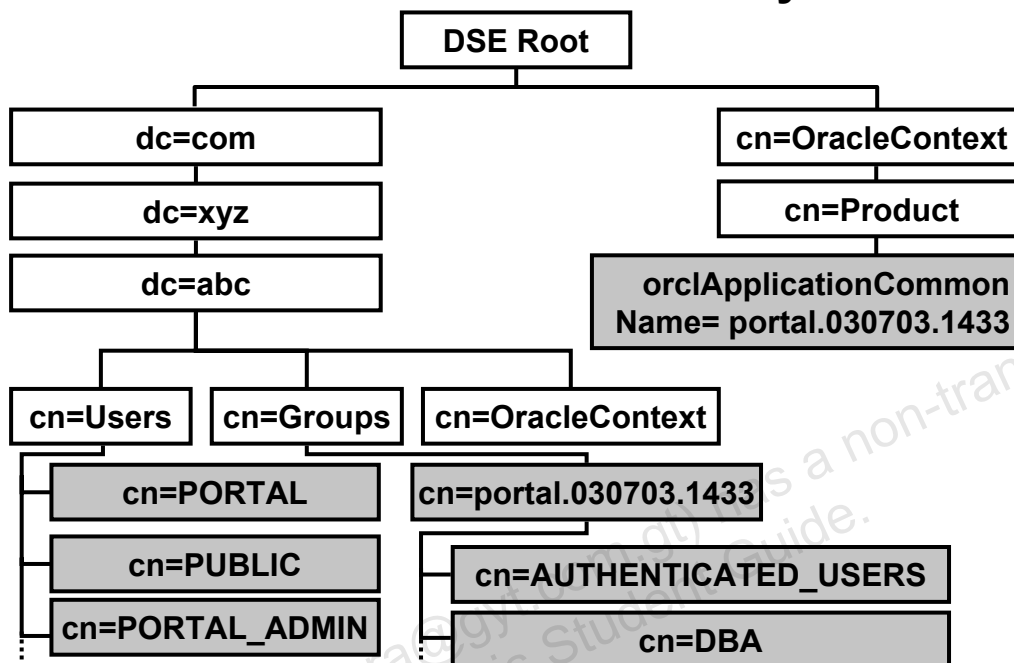
Relationship Between OracleAS Portal and Oracle Internet Directory

To provide a more comprehensive security solution, OracleAS Portal makes use of a variety of components in the Oracle Identity Management infrastructure. In particular, Oracle Internet Directory stores information about OracleAS Portal users and groups, as well as the group membership and privileges granted to portal users and groups.

Given that model, OracleAS Portal requires the following interaction with Oracle Internet Directory:

- OracleAS Portal–specific entries stored in the directory
- Group attributes stored in the directory
- User attributes stored in the directory
- Caching of user and group information from the directory
- Populating of user and group list of values from the directory through Delegated Administration Services

OracleAS Portal Directory Entries in Oracle Internet Directory



ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal Directory Entries in Oracle Internet Directory

For security to function properly, OracleAS Portal requires the following entries in the directory's DIT structure:

- OracleAS Portal default user accounts (cn=PORTAL, cn=PUBLIC, cn=PORTAL_ADMIN) are created in the default identity management realm's user container (for example, cn=Users, dc=abc, dc=xyz, dc=com).
- The OracleAS Portal group container is created in the default identity management realm's group container (for example, cn=Groups, dc=abc, dc=xyz, dc=com). The name of the OracleAS Portal group container is derived from the following in OracleAS Portal:

- Portal schema name
- Date and time of association with the Infrastructure

The format of the name is *portal_schema_name.yymmdd.hh.mi*.

- OracleAS Portal default groups (cn=AUTHENTICATED_USERS, cn=DBA, cn=PORTAL_ADMINISTRATORS, cn=PORTAL_DEVELOPERS, cn=PORTLET_PUBLISHERS, cn=RW_ADMINISTRATOR, cn=RW_DEVELOPER, cn=RW_POWER_USER, cn=RW_BASIC_USER) are created in the OracleAS Portal group container.

OracleAS Portal Directory Entries in Oracle Internet Directory (continued)

- The OracleAS Portal application entity (orclApplicationCommonName=*portal_schema_name.yymmdd.hh.mi*) is created in the Root Oracle Context (cn=Portal, cn=Products, cn=OracleContext). OracleAS Portal uses this entity to bind to the directory when it needs to query it or perform actions on it (for example, adding a user) on behalf of the user. When OracleAS Portal binds to the directory for a user, it uses a proxy connection to connect as the user. This method ensures that the user's authorization restrictions are properly enforced by the directory. The OracleAS Portal application entity obtains the privileges to initiate proxy connections by its membership in the user proxy privileges group (cn=UserProxyPrivilege, cn=Groups, cn=OracleContext).
- The OracleAS Portal directory synchronization subscription, a provisioning profile entry, is created in the provisioning profile of the directory (cn=Provisioning Profiles, cn=changelog identity management realm, cn=oracle internet directory). This entry indicates that the directory must notify OracleAS Portal when user or group privilege information has changed. It enables OracleAS Portal to keep its authorizations synchronized with the information stored in the directory. The registration is performed from the directory provisioning subscription tool.

Configuring Oracle Internet Directory Settings in OracleAS Portal

The screenshot displays the Oracle Application Server Portal Builder interface. The top navigation bar includes links for Home, Builder, Navigator, Help, Edit, Customize, Account Info, and Logout. Below this, a secondary bar shows Welcome, Build, and Administer. The main content area is titled 'Oracle Internet Directory Settings' and contains the following text:

Oracle Portal maintains Groups in the Oracle Internet Directory (OID), Oracle's Lightweight Directory Access Protocol (LDAP) Server. Use the Groups portlet to create or edit groups. When groups are created, they are created under a node of the LDAP Directory Information Tree (DIT). A node is identified by its distinguished name (DN). Specify the DN under which you want the groups to be created:

Group Creation Base DN (example: cn=PORTAL_GROUPS,cn=Groups,o=oracle,dc=com)

Similarly, when listing groups for selection, for editing or granting access, etc., the scope of groups displayed can be limited to a "Local" set of groups. Specify the DN for the search base to use when searching for local groups:

Local Group Search Base DN (example: cn=PORTAL_GROUPS,cn=Groups,o=oracle,dc=com)

Note that the group search base displaying 'All' groups is specified in configuration information in the Oracle Internet Directory. The following OID server is configured for use with the portal. It can be changed by running the MIDTIER install.

OID Host: edrsr16p1
OID Port: 389

The bottom of the page features the Oracle logo and the copyright notice: Copyright © 2005, Oracle. All rights reserved.

Configuring Oracle Internet Directory Settings in OracleAS Portal

Oracle Internet Directory settings in OracleAS Portal can be accessed and configured in the SSO/OID tab when you click the Global Settings link in the Services portlet. The following Oracle Internet Directory settings can be configured in OracleAS Portal:

- **Group Creation Base DN:** When portal groups are created through the Group portlet, they are created under a node in the DIT that is defined by the DN specified in the Group Creation Base DN global setting.
- **Local Group Search Base DN:** Just as you need to define the node in which you want to create groups, you must also define the node in which you want OracleAS Portal to search for existing groups. For example, you need to specify where OracleAS Portal searches when it displays the group's list of values in the Group portlet.

Both Oracle Internet Directory settings are particularly useful if you adapt OracleAS Portal to interact with an existing DIT.

Caching Oracle Internet Directory Information in OracleAS Portal

Cache for OID Parameters

The following parameters are cached in Portal and are used for accessing the Oracle Internet Directory and the Delegated Administration Service (DAS). If any of these values change in the directory, refresh the cache in order for Portal to function properly.

Users Search Base DN	cn=users,dc=edrsr16p1,dc=com
Global Group Search Base DN	cn=groups,dc=edrsr16p1,dc=com
User Nickname Attribute	uid
Subscriber Nickname Attribute	dc
DAS Host Name	http://edrsr16p1:7777/
Change Password URL	oiddas/ui/oracle/ldap/das/mypage/AppChgPwdMyPage
Profile URL	oiddas/ui/oracle/ldap/das/mypage/AppViewMyPage
Create User URL	oiddas/ui/oracle/ldap/das/user/AppCreateUserInfoAdmin
Edit User URL	oiddas/ui/oracle/ldap/das/user/AppEditUserAdmin
Delete User URL	oiddas/ui/oracle/ldap/das/user/AppDeleteUserAdmin
User LOV URL	oiddas/ui/oracle/ldap/das/search/LOVUserSearch
Create Group URL	oiddas/ui/oracle/ldap/das/group/AppCreateGroupInfoAdmin
Edit Group URL	oiddas/ui/oracle/ldap/das/group/AppEditGroupAdmin
Delete Group URL	oiddas/ui/oracle/ldap/das/group/AppDeleteGroupAdmin
Group LOV URL	oiddas/ui/oracle/ldap/das/search/LOVGroupSearch

Refresh Cache for OID Parameters

Select the following checkbox and click on OK or Apply above to refresh the cache with new or changed OID parameters.

☐ Refresh Cache for OID Parameters

ORACLE

Copyright © 2005, Oracle. All rights reserved.

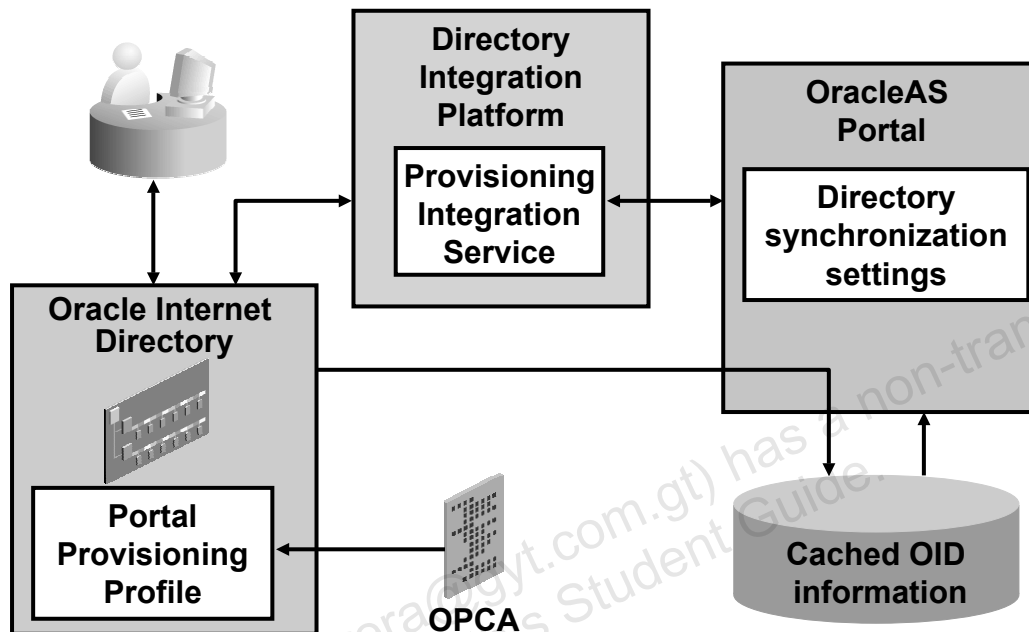
Caching Oracle Internet Directory Information in OracleAS Portal

To improve performance, OracleAS Portal caches some directory information locally. In particular, OracleAS Portal caches the following:

- Directory connection information for OracleAS Portal
- URLs for Delegated Administration Services
- orclGUIDs of certain privilege groups for authorization checks on directory portlets (for example, the User and Group portlets)
- Group memberships and default group for each user

You can refresh this cache with updated information from the directory by selecting the Refresh Cache for OID Parameters check box on the SSO/OID tabbed page, and submitting the selection by clicking the Apply button or the OK button.

Synchronizing Cached Oracle Internet Directory Information in OracleAS Portal



Copyright © 2005, Oracle. All rights reserved.

Synchronizing Cached Oracle Internet Directory Information in OracleAS Portal

The majority of information cached by OracleAS Portal is fairly static (for example, directory connection information). For those items that are more dynamic, such as group memberships and default groups, OracleAS Portal relies on the Oracle Directory Provisioning Integration service of the Oracle Directory Integration Platform (DIP) for updates. The service notifies OracleAS Portal whenever a change is made in the directory that must be reflected in OracleAS Portal. Updated information is pushed to the portal, which, in turn, updates it in its cached Oracle Internet Directory information store.

To support this notification mechanism, OracleAS Portal requires the following:

- The DIP must be running. To start the DIP, you can use the `oidctl` command. For example:
`oidctl instance=1 odisrv start`
- The subscription-provisioning profile must be created in Oracle Internet Directory for OracleAS Portal. The profile defines a list of specified events in the directory to which OracleAS Portal is subscribed (for example, user and group deletion). The initial profile is created by OracleAS Portal Configuration Assistant (OPCA) during the installation. You can also delete an existing profile or create a new profile by running OPCA in the MIDTIER mode manually, which may be required when you relink the portal with another Oracle Internet Directory or when you upgrade the portal.

Oracle Application Server 10g R2: Administration I 12-52

Enabling Directory Synchronization in the OracleAS Portal Instance

Directory Synchronization

The Directory Integration Platform (DIP) server can be configured to send notification messages to the Portal. The messages can be sent when events such as user and group deletions and modifications occur so that the Portal can perform actions such as cleanup and cache invalidation.

Indicate whether you want directory synchronization enabled. The checkmark indicates the current status. If synchronization is enabled, you can specify how often the Directory Integration Platform (DIP) server sends event notification messages.

☒ Enable directory synchronization

Send event notifications every seconds

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Enabling Directory Synchronization in the OracleAS Portal Instance

The directory synchronization should be enabled in the OracleAS Portal instance. This can be done in the Directory Synchronization section on the SSO/OID tabbed page of the Global Settings page by selecting the Enable directory synchronization check box. If this check box is not selected, the portal will not be notified of any directory integration server subscribed events. You can also specify the interval of time between event notifications sent by the Oracle Directory Provisioning Integration service to the OracleAS Portal instance in seconds in the Directory Synchronization section.

Summary

In this lesson, you should have learned how to:

- **Describe Identity Management**
- **Explain the default Identity Realm**
- **Describe the OracleAS Administration Model**
- **Explain application-specific access control**
- **Manage users and groups**
- **Describe relationship between OracleAS Portal and Oracle Internet Directory**
- **Identify OracleAS Portal entries in the directory**
- **Configure Oracle Internet Directory settings in OracleAS Portal**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

13

Managing the OracleAS Portal

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Describe OracleAS Portal administrative services**
- **Describe tools to monitor the OracleAS Portal instance**
- **Manage OracleAS Portal users, groups, and schemas**
- **Administer the Portlet Repository**
- **Deploy portlets to OracleAS Portal by using Web providers, WSRP producers, and database providers**
- **Perform export and import of portal content**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal Administrative Services: Overview

OracleAS Portal administrative services:

- **Enable you to:**
 - **Manage portal users and groups**
 - **Set up security, search, and self-registration features**
 - **Configure language and mobile support**
 - **Migrate content between OracleAS Portal instances**
 - **Monitor the performance of OracleAS Portal instances**
- **Are provided in the form of:**
 - **Application Server Control**
 - **Administrative portlets**
 - **Configuration scripts**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal Administrative Services: Overview

The OracleAS Portal framework provides administrative services, such as access to monitoring and configuration tools, single sign-on, directory integration, caching, and security. The services enable you to perform configuration and administrative tasks after the installation is complete. For example, as a portal administrator, you need to:

- Manage users and groups
- Set up security, search and self-configuration features to configure language and mobile support in your portal
- Perform portal page administration

To perform most of the administrative and configuration tasks in the OracleAS Portal instance, you must log in to the portal as a portal administrator and use administrative portlets. Some of the administrative tasks should only be performed by using Application Server Control or by running configuration scripts that are copied into your Oracle home directory during the installation of OracleAS Portal.

Managing the OracleAS Portal Instance by Using Application Server Control

You can use Application Server Control to monitor and administer the OracleAS Portal instance.

System Components				
<div>Start Stop Restart Delete OC4J Instance</div> <div>Enable/Disable Components Configure Component Create OC4J Instance</div>				
Select All Select None				
Select	Name	Status	Start Time	Memory Usage (MB)
<input type="checkbox"/>	home	↑	Sep 6, 2005 4:41:20 AM	24.13
<input type="checkbox"/>	HTTP_Server	↑	Sep 6, 2005 4:41:17 AM	149.68
<input type="checkbox"/>	OC4J_Portal	↑	Sep 6, 2005 4:41:20 AM	69.23
<input type="checkbox"/>	OC4J_Temp	↑	Sep 8, 2005 4:03:46 AM	43.23
<input type="checkbox"/>	Portal:portal	↑	N/A	N/A
<input type="checkbox"/>	Web Cache	↑	Sep 6, 2005 4:41:17 AM	15.45
<input type="checkbox"/>	Management	↑	Sep 6, 2005 4:43:22 AM	106.15

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing the OracleAS Portal Instance by Using Application Server Control

Application Server Control provides the management tools to monitor and administer the OracleAS Portal instance. In the System Components section of the OracleAS instance home page, you can find two entries that are related to the OracleAS Portal instance: OC4J_Portal and Portal:portal.


OC4J_Portal is an OC4J instance that contains Web applications related to the OracleAS Portal instance (for example, the Parallel Page Engine). This instance is configured and started during the installation. You can start or stop the OC4J_Portal similarly to the way you start or stop any other OC4J instance.

Portal:portal is a link to the home page of the OracleAS Portal instance in Application Server Control. This is the first place to go to check the condition of the OracleAS Portal instance. From the home page of the OracleAS Portal instance, you can manage and monitor all the components that make up the OracleAS Portal instance, such as Oracle HTTP Server, mod_plsql, Web Cache, and providers.

OracleAS Portal Instance Home Page

Portal:portalPage Refreshed Oct 8, 2005 1:20:17 AM

General



Status

Up

Average Page Requests Per Hour

6

Homepage Download (seconds)

0.449

OracleAS Metadata Repository Used By Portal

Status

Up

Name

infra

Start Time

Oct 4, 2005 1:10:35 AM

Database Version

10.1.0.4.2

Repository Version

10.1.4.0.0

Administration

[Portal Web Cache Settings](#)[Portal Cache Settings](#)[Portal DAD Settings](#)

Related Links

[Portal End User Default Homepage](#)[All Metrics](#)

Component Status

OracleAS components used by Portal.

Component	Up/Down
HTTP Server	↑
Parallel Page Engine Services	↑
Providers	↑
Ultra Search	↑

Severity Status

OracleAS components used by Portal that indicate severity status.

Component	Severity
Parallel Page Engine Services	✓
Providers	✗

OK ✓ Warning ⚠ Critical ✗ Unknown ?

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal Instance Home Page

From the home page of the OracleAS Portal instance, you can see the overall status of the instance, data on how the instance is using OracleAS Metadata Repository, and status of all other Oracle Application Server components that the OracleAS Portal instance is dependent on. For components that are specifically used by the OracleAS Portal instance, you can also see the severity status.

You can monitor the status of PPE from the Parallel Page Engine Services home page, or you can monitor all components registered with the OracleAS Portal instance providers and their portlets from the Providers home page.

The Administration section contains the Portal Web Cache Settings link that enables you to reconfigure the OracleAS Portal instance when there are changes in the Web Cache configuration.

The Portal End User Default Homepage link in the Related Links section takes you to the Welcome page of the OracleAS Portal instance. The All Metrics link provides a single comprehensive list of all the metrics available for this OracleAS Portal instance.

Monitoring the OracleAS Portal Instance

Available tools and services:

- **Oracle Enterprise Manager 10g Application Server Control**
- **OracleAS Portal activity reports:**
 - OracleAS Portal logging service
- **OracleAS Portal performance reports**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Monitoring the OracleAS Portal Instance

By monitoring the OracleAS Portal instance, you can easily analyze and understand better the type and volume of activities that are taking place in the portal. With this information, you can make better-informed decisions and take appropriate administrative actions to improve performance, usability, navigation, and so on.

The following tools and services are available to you as a portal administrator:

- **Oracle Enterprise Manager 10g Application Server Control (Application Server Control):** This management interface is installed with every instance of Oracle Application Server. The interface immediately provides you with the tools to monitor the OracleAS Portal instance, start and stop services, view logs and ports, and configure settings and metrics related to the OracleAS Portal instance.

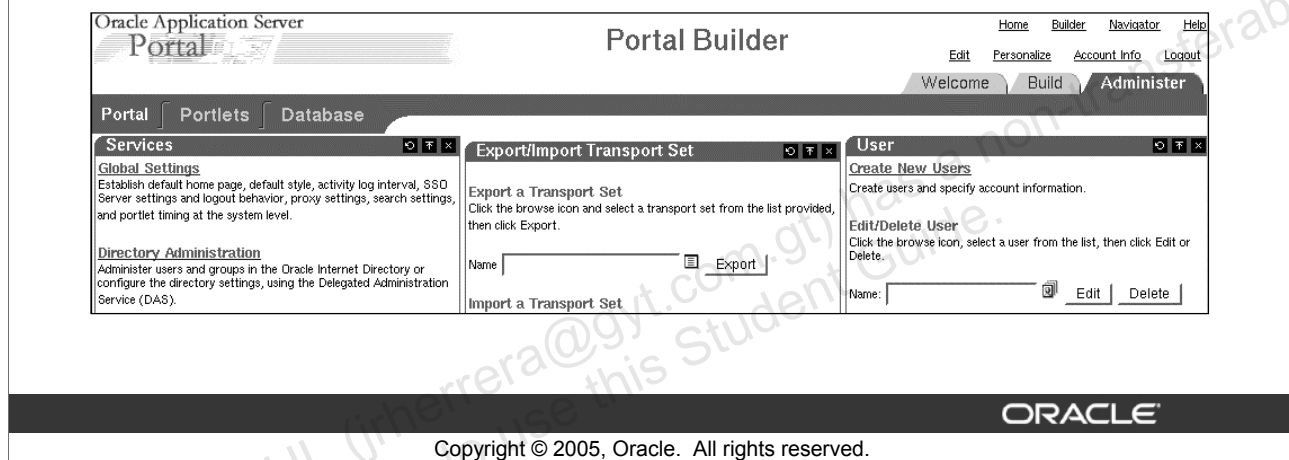
Monitoring the OracleAS Portal Instance (continued)

- **OracleAS Portal activity reports:** To analyze the data stored in the Activity Log tables, OracleAS Portal provides access to several Activity Log views. These views exist in the OracleAS Portal product. Access to the Activity Log views is granted to the public. However, the logs are secure according to the portal object's security. If required, you can create simple reports based on these views.
 - **OracleAS Portal logging service:** You can log objects and actions in OracleAS Portal and generate reports for analyzing the data. The OracleAS Portal logging service collects information about registered events into OracleAS Portal Activity Log tables. You can choose which events are logged in the Activity Log tables by managing the Log Registry records in the Services portlet.
- **OracleAS Portal performance reports:** You can generate performance reports based on the statistics collected by the `mod_plsql` performance logging service. This can be accomplished by running the Performance Reporting SQL scripts that are located in the `$ORACLE_HOME/portal/admin/plsql/perf` directory. The `README.html` file from the same directory provides you with instructions on how to load the logging data into the database and generate performance reports.

Managing the OracleAS Portal Instance by Using Administrative Portlets

OracleAS Portal administrative portlets are:

- Grouped into three subtabs on the Administer tabbed page
- Integrated with other Oracle Application Server components



Managing the OracleAS Portal Instance by Using Administrative Portlets

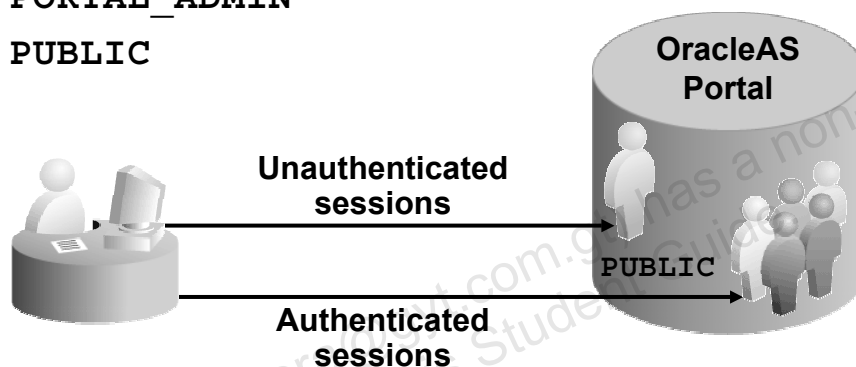
To perform various administrative functions, you need to log in to the OracleAS Portal instance as an administrator. The Portal Builder page is displayed after you have logged in to the portal. The Administer tabbed page includes three subtabs that group related administrative portlets:

- **Portal:** Portlets on this subtab enable you to create portal users and groups, configure global settings of the portal instance, administer other services (for example, the SSO server, Delegated Administration Service, Oracle Ultra Search, OracleAS Web Cache, and proxy settings), perform export and import of OracleAS Portal objects, and so on.
- **Portlets:** From this subtab, you can manage the Portlet Repository that stores information about registered providers and portlets in the portal. You can also register new remote providers and provider groups.
- **Database:** Portlets on this subtab enable you to create and edit database schemas, create and edit database roles, and monitor database information, such as database parameters, memory consumption, and database storage details.

Default Portal Users

The following portal users are created upon installation:

- ORCLADMIN
- PORTAL
- PORTAL_ADMIN
- PUBLIC



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Default Portal Users

The following default portal users are created upon installation of OracleAS Portal:

- **orcladmin:** This user is created for Oracle Application Server administrators and has the highest privileges in the portal. Some of the OracleAS Portal tools (for example, the SSO Server Administration portlet) are available only for this user.
Note: The `orcladmin` Oracle Internet Directory superuser and the `orcladmin` portal user are two different users that are stored in the same directory.
- **portal:** This user is the superuser for the portal, and is granted all the privileges available in the portal.
- **portal_admin:** This user is a privileged OracleAS Portal user with administrative privileges excluding those that would give the user the ability to obtain higher privileges or access the database administration features, such as schema creation and management. This user is typically intended for an administrator who manages and provisions portal users.
- **public:** This user identifies unauthenticated access to the portal. All sessions before authentication use this account.

The initial password for the `orcladmin`, `portal`, and `portal_admin` portal users is the same as the password supplied for the `ias_admin` user supplied during the installation of OracleAS Infrastructure.

Default Portal Groups

- **Basic groups:**
 - AUTHENTICATED_USERS
 - DBA
 - PORTAL_ADMINISTRATORS
 - PORTAL_DEVELOPERS
 - PORTLET_PUBLISHERS
- **Groups that support OracleAS Reports Services:**
 - RW_BASIC_USER
 - RW_POWER_USER
 - RW_DEVELOPER
 - RW_ADMINISTRATOR

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Default Portal Groups

OracleAS Portal creates several groups upon installation to implement the base user privileges and additional portal-level privileges:

- **AUTHENTICATED_USERS:** Users are made a member of the AUTHENTICATED_USERS group when they log in to Portal. The purpose of this group is to provide a convenient mechanism to assign the default privileges that you want every logged-in user to have in the portal.
- **DBA:** Members of this group have the maximum privilege level in the system. All global privileges are granted to this group. Initially, this group has only one member, the user with the name of the product schema (for example, portal).
- **PORTAL_ADMINISTRATORS:** This group includes users with most of the global privileges, except for the database-related privileges. Members of this group do not have the necessary privileges to administer OracleAS Single Sign-On. This group initially comprises the portal_admin user and includes the DBA group.
- **PORTAL_DEVELOPERS:** Members of this group have privileges to build and manage local database providers and their portlets, as well as shared components.

Default Portal Groups (continued)

- **PORTLET_PUBLISHERS:** This group includes users who have privileges to add portlets to portal pages and make the portlets available to other portal users.

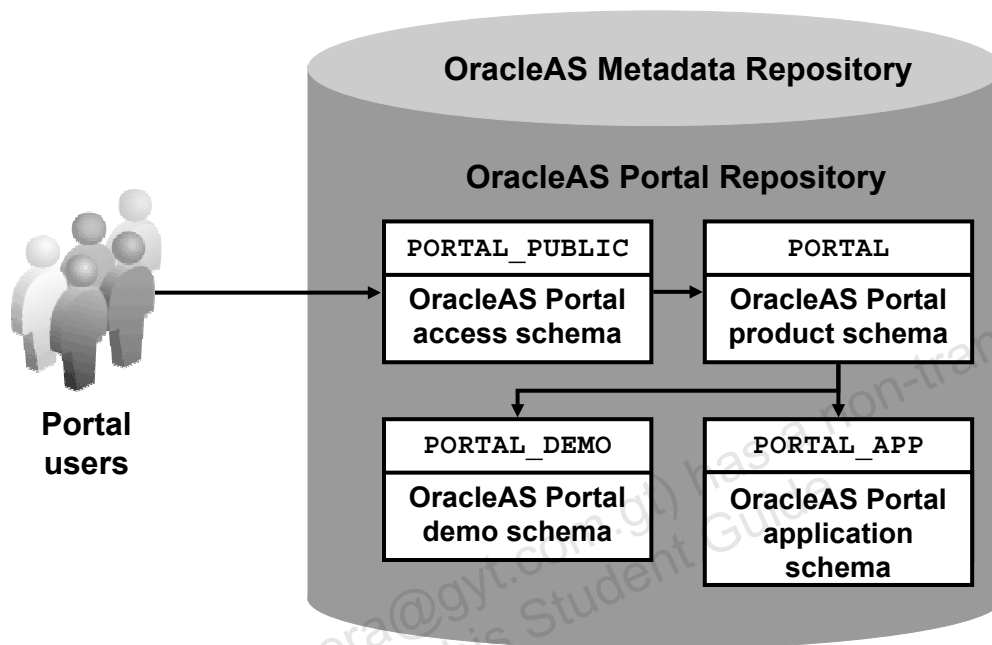
To support integration with OracleAS Reports Services, OracleAS Portal creates a set of reports-related groups. The reports-related privileges of these groups are defined as follows:

- **RW_BASIC_USER:** Users of this group can only execute deployed reports and view less detailed error messages than members of the other group.
- **RW_POWER_USER:** In addition to the privileges of the RW_BASIC_USER group, this group includes users who receive more detailed error messages from OracleAS Reports Services.
- **RW_DEVELOPER:** In addition to the privileges of the RW_POWER_USER group, users of this group can run special Web commands that generate the Oracle Reports system environment.
- **RW_ADMINISTRATOR:** In addition to the privileges of the RW_DEVELOPER group, users of this group have access to the administrator's functionality of the Oracle Reports Queue Manager administer reports built with Oracle Reports, as well as Oracle Reports printer and server definitions. By using these administrative privileges, members of this group can manage the server queue and perform rescheduling, deletion, and reordering of jobs in the server.

Every portal user who needs access to reports built with Oracle Reports and deployed to the OracleAS Middle Tier should belong to one of the reports-related groups. In addition, the portal administrator should grant appropriate portal privileges to the reports-related groups. For example, to enable members of the RW_BASIC_USER group to run a report from the portal, the Execute privilege should be granted on the corresponding Oracle Reports report object to the RW_BASIC_USER group.

Note: For more information about deploying reports in OracleAS Portal, refer to *OracleAS Reports Services Publishing Reports to the Web*.

OracleAS Portal Schemas



Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal Schemas

OracleAS Portal is installed primarily in the Oracle database with some supporting components installed on the middle tier of Oracle Application Server. During a typical installation, the following database schemas are created:

- **PORTAL:** This is the product schema for OracleAS Portal that contains database objects of the Portal Repository and PL/SQL code. It is a highly privileged database schema and acts as a proxy user for the interaction of the middle tier with the database, which allows the middle tier to have secure access to other schemas in the database.
- **PORTAL_PUBLIC:** Portal users do not have distinct Oracle database accounts and schemas. Each portal user must be mapped to a database schema that should be other than the product schema. The PORTAL_PUBLIC schema is the schema that portal users map to by default.
- **PORTAL_DEMO:** This schema contains the OracleAS Portal demonstration code.
- **PORTAL_APP:** This schema contains the OracleAS Portal applications.

Managing Passwords for the OracleAS Portal Schemas

- Passwords are stored in Oracle Internet Directory.
- You must change the passwords using Application Server Control.

Change Schema Password			
Select the schema and enter the new password. The password will be changed in the database as well as in the central storage in Internet Directory.			
Select Schema	Component	Database	
<input checked="" type="radio"/> PORTAL	Portal	ldap://edrsr16p1:389/infra,cn=oraclecontext	
<input type="radio"/> PORTAL_APP	Portal	ldap://edrsr16p1:389/infra,cn=oraclecontext	
<input type="radio"/> PORTAL_DEMO	Portal	ldap://edrsr16p1:389/infra,cn=oraclecontext	
<input type="radio"/> PORTAL_PUBLIC	Portal	ldap://edrsr16p1:389/infra,cn=oraclecontext	
<input type="radio"/> UDDISYS	OC4J	ldap://edrsr16p1:389/infra,cn=oraclecontext	
<input type="radio"/> WKPROXY	Ultra Search	ldap://edrsr16p1:389/infra,cn=oraclecontext	
<input type="radio"/> WKSYS	Ultra Search	ldap://edrsr16p1:389/infra,cn=oraclecontext	
* Password		<input type="text"/>	
* Confirm Password		<input type="text"/>	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing Passwords for the OracleAS Portal Schemas

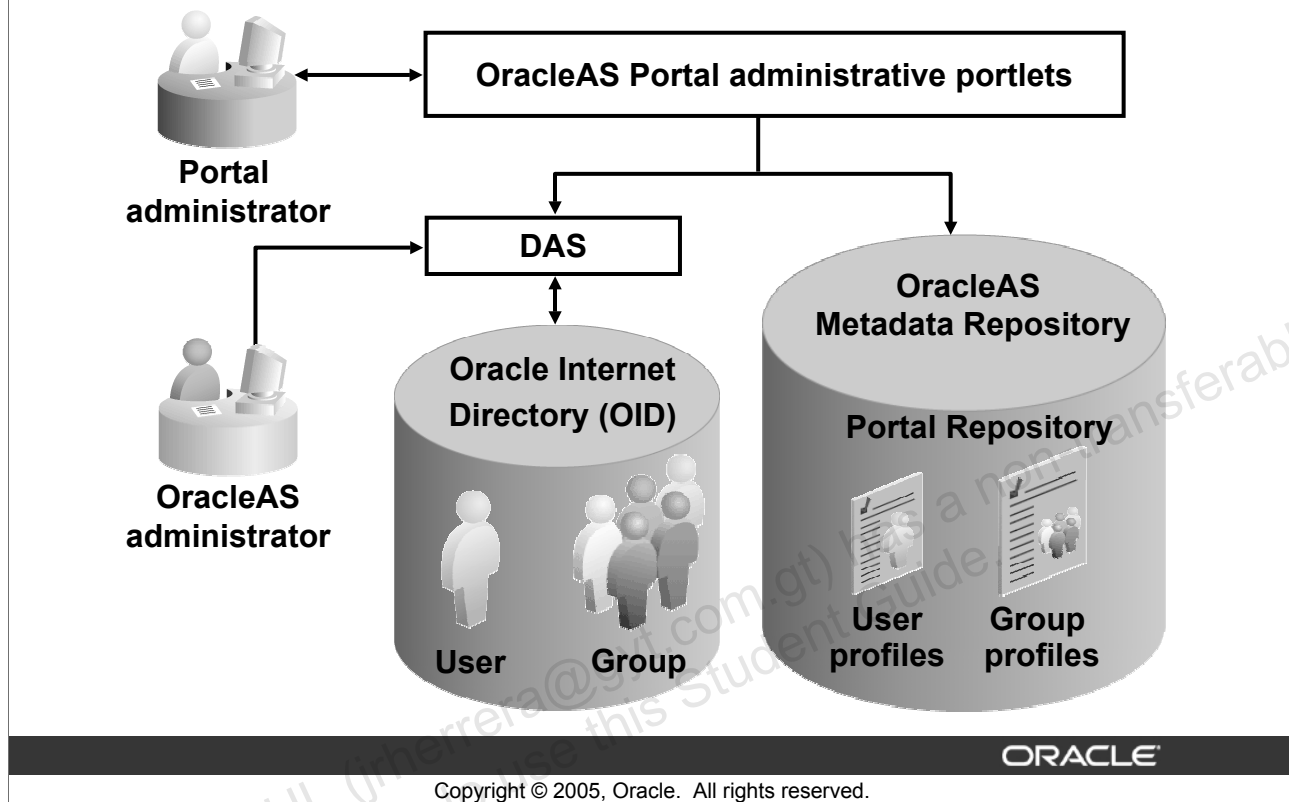
During the installation, passwords for the OracleAS Portal schemas are randomized and stored in Oracle Internet Directory. You can retrieve these passwords from Oracle Internet Directory using Oracle Directory Manager. For example, to retrieve the password for the product schema of OracleAS Portal, you navigate to the following entry in the DIT:

```
Entry management > cn=OracleContext > cn=Products > cn=IAS >
  cd= IAS Infrastructure Databases > orclReferenceName=your
  Infrastructure DB name> cnOrclResourceName=PORTAL.
```

If you want to change the password for any of the OracleAS Portal schemas, you must do so using Application Server Control:

1. Click the Infrastructure tab on the home page of the middle-tier OracleAS instance.
2. In the Metadata Repository section, click the Change Schema Password link.
3. On the Change Schema Password page, select a schema (for example, PORTAL), enter a new password in the Password field and the Confirm Password field, and click OK. You should receive the following confirmation message: "The operation Change Schema Password was successful."

Managing Portal Users and Groups



Managing Portal Users and Groups

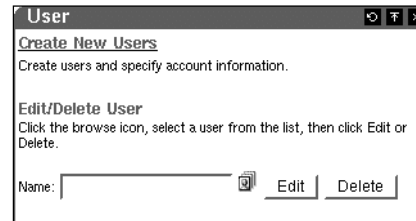
Portal users and groups are stored in Oracle Internet Directory. The management interface that is provided by Delegated Administration Service (DAS) is used to enter information about portal users and groups into Oracle Internet Directory. Oracle Application Server administrators and portal administrators can access DAS using direct URLs or via the User and Group portlets from the Administer tab on the OracleAS Portal Builder page.

Portal users are single sign-on (SSO) user accounts, which allow a user to access multiple applications including OracleAS Portal by providing his or her credentials just once.

To manage user and group information that pertain specifically to the portal, OracleAS Portal creates user and group profiles for each portal user and group stored in Oracle Internet Directory. OracleAS Portal stores the user and group profiles in the Portal Repository. The user and group profiles are created automatically when a portal administrator first attempts to edit the user or group profile of a user or group. The user profile also is created when a portal user first attempts to log in to OracleAS Portal using his or her credentials. To manage user profiles and group profiles, you use the Portal User Profile portlet and Portal Group Profile portlet, respectively.

Creating Portal Users

- **Use the User portlet.**
- **Specify the following:**
 - **Basic information**
 - **Personal details**
 - **Organizational details**
 - **Photograph**
 - **Telephone numbers**
 - **Home/office addresses**
 - **Group membership**
 - **Privilege assignment**
 - **Resource access information for Reports and Forms applications**



ORACLE

Copyright © 2005, Oracle. All rights reserved.

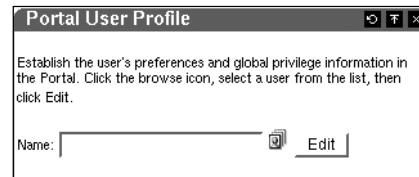
Creating Portal Users

When you create a portal user by using the User portlet, you are creating a single sign-on user account for that portal user in Oracle Internet Directory. You must enter basic information, such as a username, password, and e-mail address, for each user that you create. You can also provide optional personal information, job-related information, telephone numbers, and addresses. You can upload a photograph in GIF or JPEG format. You can also assign the user to be a member of the existing portal groups, for example, the default OracleAS Portal groups. You can also assign portal privileges to each user, which is discussed later in this lesson. You are provided with a section to enter resource access information for Reports and Forms applications. For example, you can save connection string information to a data source that is used when running reports that need access to that particular data source.

Editing Portal User Profiles

- Use the Portal User Profile portlet.
- Configure the following:

- Enabling access to the portal
- Database schema to use
- The portal page for the user's personal use
- Default group and style
- Default home page
- Default mobile home page
- Invalidation of the user's portal content in the Web Cache
- Global portal privileges



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Editing Portal User Profiles

The Portal User Profile portlet enables you to define the user information that pertains specifically to the OracleAS Portal instance. For example, on the Preferences tabbed page, you can configure the following settings:

- **Allow User To Log On:** When you edit a portal user profile, you can enable or disable the user's ability to log in to the OracleAS Portal instance via the Allow User To Log On check box.
- **Database Schema:** Portal users do not have database privileges. However, because portal pages are displayed by executing procedures in the database, portal users must have execute privileges on those procedures. Therefore, each portal user must be associated with a database schema that has the appropriate privileges to display portal pages. By default, new portal users are associated with the PORTAL_PUBLIC schema.
- **Default Group:** You can select a default group for the user. The default group determines the preferences for the user if no personal preferences are specified.

Editing Portal User Profiles (continued)

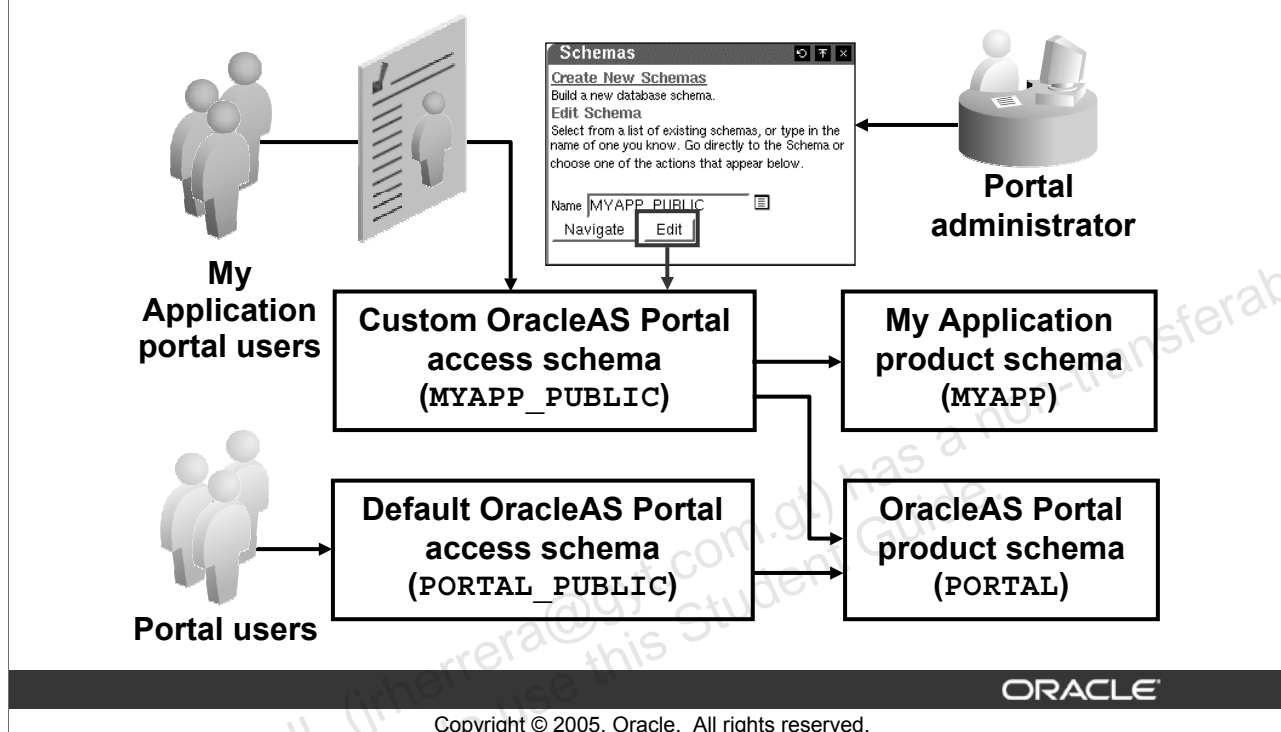
- **Default Style:** You can choose the style to be used as the user's default style. The user's default style is used when pages are set to use the user's default style. If the user does not have a default style, the style of the user's default group is used instead. If the user does not have a default group, or the group default style is also not set, then the system default style is used.

Note: Users can change this setting on the Account Information page if they want.

- **Default Home Page:** The home page is the first page that is displayed to a user after logging in to OracleAS Portal. If the user has specified a personal home page, that page is displayed when the user logs in. If the user has not selected a personal home page, but the portal administrator has set one for him or her, then the default home page specified for that user is displayed. If the user has not selected a personal home page, but belongs to a default group, then the Default Home Page specified for that group is displayed. If there is no default home page for the user's default group, the system's default home page is displayed.
- **Default Mobile Home Page:** Similar to the Default Home Page, you can set the Default Mobile Home Page if the mobile support is enabled in the portal. If you select the Default Mobile Home page here, it overrides the Default Mobile Home Page of the user's default group.
- **Clear the Cache in Web Cache for User:** Select this check box to invalidate the pages associated with this user in the Web Cache. This enables new pages to be generated for this user, which may be desirable when, for example, a new default group is selected.

On the Privileges tabbed page, you can set OracleAS Portal global privileges for the user. You can also reset the privileges.

Mapping Portal Users to a Custom OracleAS Portal Access Schema



Mapping Portal Users to a Custom OracleAS Portal Access Schema

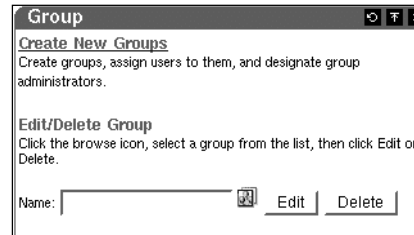
In some cases, you may want to map OracleAS Portal users with another database schema that plays the same role as the default OracleAS Portal access schema (PORTAL_PUBLIC). For example, if your company has a legacy database application that can be accessed only by a limited number of portal users, then you can create a new database schema—that is, a custom OracleAS Portal access schema (for instance, MYAPP_PUBLIC)—and map those portal users who need access to the legacy application to the new database schema. The custom OracleAS Portal access schema should also be granted all necessary database privileges from the application product schema to be able to run the application code and access the application data.

To map a portal user to a custom OracleAS Portal access schema, perform the following steps:

- Add the custom OracleAS Portal access schema to the list of database schemas to which portal users can map. In the Schemas portlet, select a database schema, click Edit, select the Use this Schema for Portal Users check box, and apply the change.
- Edit the user profile, and select the custom OracleAS Portal access schema in the Database schema field.

Creating Portal Groups

- Use the Group portlet.
- Specify the following:
 - Basic information
 - Group information:
 - Public
 - Private
 - Enable group to be privileged
 - Owners
 - Members
 - Privilege assignment



ORACLE

Copyright © 2005, Oracle. All rights reserved.

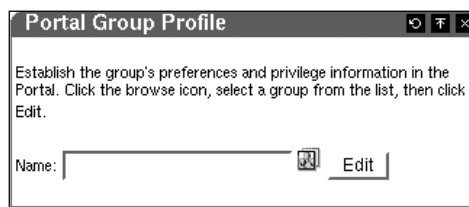
Creating Portal Groups

When you create a portal group, you specify basic information such as name, display name, and description of the group. You also specify the visibility of the group as public or private. If you specify the group as private, the group is visible only to its owners. The default visibility is public. If you specify the group to be privileged, then you can assign privileges to the group.

The creator of the group is automatically the group owner. You can also specify additional owners for the group. You can add users and groups as members of the group and you can also assign privileges to the group.

Editing Portal Group Profiles

- Use the Portal Group Profile portlet.
- Configure the following:
 - The Default Home Page
 - The Default Mobile Home Page
 - Global portal privileges



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Editing Portal Group Profiles

The Portal Group Profile portlet enables you to define the group information that pertains specifically to the OracleAS Portal instance. You can specify the default home page for users that have the group as their default group. However, users may override this setting by choosing their own personal home pages.

You can also specify global privileges for the group.

Note: You can choose a style that will be set as the default style for all OracleAS Portal pages.

Assigning Privileges to OracleAS Portal Users and Groups

You can assign the following privileges:

- **Oracle Application Server privileges:**
 - Stored in Oracle Internet Directory
 - Managed using DAS
- **OracleAS Portal object privileges:**
 - Stored in the Portal Repository

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Assigning Privileges to OracleAS Portal Users and Groups

Within OracleAS Portal, you decide at what level of granularity you want to control access. You can assign privileges on a per-user or per-group basis. Privileges that you can assign to OracleAS Portal users and groups can be grouped into the following types:

- **Oracle Application Server privileges:** These privileges enable users to perform user and group management, assign access rights to other users and groups, and configure user entries and subscriber information using the Delegated Administration Service. The Oracle Application Server privileges are stored in Oracle Internet Directory along with the user or group information.
- **OracleAS Portal object privileges:** These privileges give a user or group a certain level of privileges on only a particular instance of a portal object, rather than all objects of that type.

User and Group Lists of Values

- **User, Group, Portal User Profile, and Portal Profile portlets include lists of values (LOVs).**
- **LOVs support is implemented in Oracle Application Server 10g Release 2.**

**Oracle Application Server
10g Release 2 (Portal and
Delegated Administration
Services)**

**Earlier version of OracleAS
Portal and Delegated
Administration Services**

**Callback method available
to implement LOV**

Execute:

**– secjsdom.sql to reset
common domain
– secdaslc.sql**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

User and Group Lists of Values

The User, Group, Portal User Profile, and Portal Group Profile portlets include lists of values (LOVs) for users or groups. These LOVs must be populated with information stored in the directory.

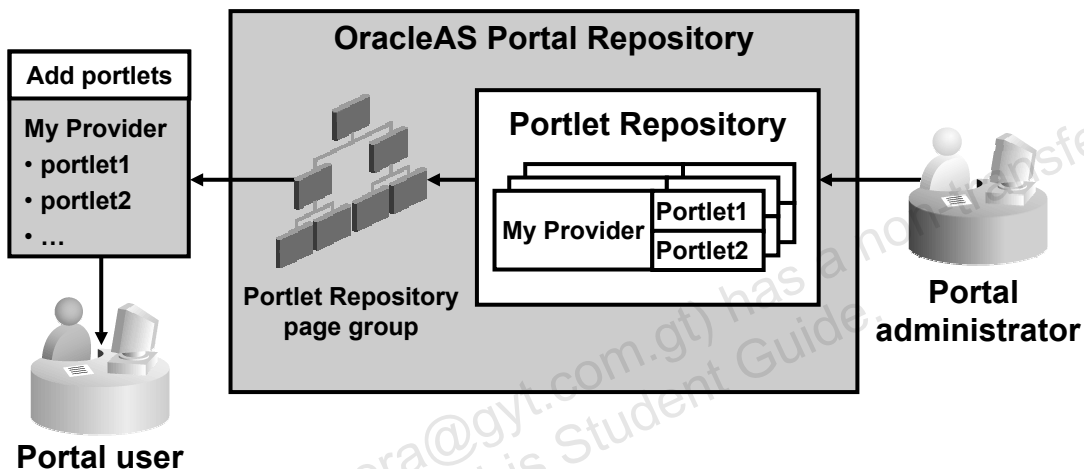
User and Group LOVs work efficiently in OracleAS Portal 10g Release 2 through the implementation of a new callback method. Oracle Delegated Administration Services posts the selected values to the callback method in OracleAS Portal's domain to avoid the cross-domain JavaScript issues. This requires support from both OracleAS Portal and Oracle Delegated Administration Services, which is available in Oracle Application Server 10g Release 2 of these components.

However, if you have upgraded from an earlier version of OracleAS Portal 10g Release 2, or if OracleAS Portal is used against an older version of Oracle Delegated Administration Services that does not support the callback method, then you must perform the following configuration steps:

- Execute the `secjsdom.sql` script to reset the common domain that was defined.
- If OracleAS Portal was configured to use a locally deployed Oracle Delegated Administration Services servlet, then reconfigure it to point to the Infrastructure tier by running the `secdaslc.sql` script.

What Is the Portlet Repository?

The Portlet Repository stores registration information about providers and their portlets that are available in the OracleAS Portal instance.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is the Portlet Repository?

To store information about providers and their portlets available in the portal, OracleAS Portal uses the Portlet Repository that is created during installation. The initial Portlet Repository stores information about built-in providers and their portlets that are installed and configured for portal administration, portal development, and general use by portal users. When you register a new provider, information about the provider and its portlets is automatically added to the Portlet Repository.

The Portlet Repository is implemented as a part of the Portal Repository in the portal product schema. To display the Portlet Repository to portal users, OracleAS Portal is shipped with a special page group, that is, the Portlet Repository page group. The Portlet Repository page group content populates the Add Portlets page that is displayed when the portal user wants to add a portlet to a portal page. What the portal user can see on the Add Portlets page depends on the user's portal privileges, which can be defined by the portal administrator.

Accessing the Portlet Repository

You can access the Portlet Repository from the Providers tabbed page of the Portal Navigator.

Oracle Application Server Portal Navigator

Home Builder Navigator Help
Customize Account Info Logout

Page Groups Providers Database Objects

Browse the Provider Groups and Providers available to you.

Find: Go

Path: Providers

Type ▲▼	Name ▲▼	Actions	Creator ▲▼	Last Modified ▲▼ ?
<input type="checkbox"/> Locally Built Providers	Locally Built Providers			06-SEP-2005
<input type="checkbox"/> Registered Providers	Registered Providers			
<input type="checkbox"/> Provider Groups				

Path: Providers > Registered Providers

Type ▲▼	Name ▲▼	Actions	Creator ▲▼	Last Modified
Database Provider	QIP Provider	Edit Registration , Refresh , Grant Access	PORTAL	15-AUG-2005
Database Provider	Oracle Instant Portals	Edit Registration , Refresh , Grant Access	PORTAL	15-AUG-2005

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Accessing the Portlet Repository

You can access the Portlet Repository from the Providers tabbed page of the Portal Navigator. All the providers available in the OracleAS Portal instance are grouped into three provider groups:

- **Locally Built Providers:** These are providers that are created by using tools available in OracleAS Portal. For example, when a portlet developer creates a form portlet or a report portlet by using the Portlet Builder, a new database provider is created in the Locally Built Providers group. The registration of the locally built providers is handled by the OracleAS Portal instance internally.
- **Registered Providers:** These are providers that are registered with the OracleAS Portal instance through the registration process by the portal administrator.
- **Provider Groups:** A Provider Group is a logical collection of Web Providers that is defined by a remote Provider Groups Service. After it is registered, a Provider Group simplifies the process of registering the providers in the group.

Managing the Portlet Repository

You can perform the following management tasks:

- **Register providers.**
- **Update provider registration information.**
- **Refresh the Portlet Repository and individual providers.**
- **Organize the Portlet Repository page group.**
- **Secure the Portlet Repository page group.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing the Portlet Repository

As a portal administrator, you can perform the following management tasks related to the Portlet Repository:

- Registering providers is the most frequent management task that adds information about new providers and their portlets to the Portlet Repository. Portlet developers typically submit setup instructions that include registration details about the provider.
- You can update registration details about the existing providers in the Portlet Repository. For example, you can change the provider's display name or timeout message, or you can change access to the provider for portal users.
- When there are changes in the provider implementation (for example, a new portlet has been added to the provider), you need to refresh the provider registration information in the Portlet Repository. You can perform this task in two ways: refresh an individual provider or refresh the entire Portlet Repository. Refreshing an individual provider is cheaper and requires less time. In this case, the portal contacts the provider and updates the list of the provider's portlets in the Portlet Repository. Refreshing the entire Portlet Repository updates information about all providers that are registered in the Portlet Repository.

Managing the Portlet Repository (continued)

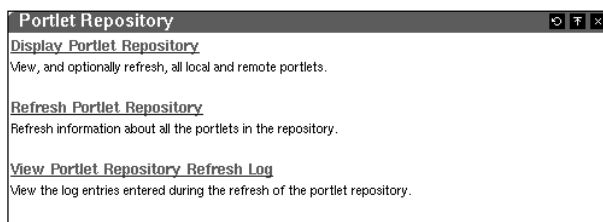
This task may take a long time, depending on the size of the Portlet Repository, and should be performed when the load on the OracleAS Portal instance is minimal. During this task, the portal contacts all the registered providers and updates their registration information in the Portlet Repository.

- You can customize the display of the Portlet Repository by organizing the Portlet Repository page group content to make it easier for the portal users to locate required portlets. For example, if you have many portlets that relate to a sales theme, you can create a page called Sales within the Portlet Repository page group, and move portlet items from their original pages to the Sales page.
- You can secure access to the Portlet Repository by granting access to pages and portlet items of the Portlet Repository page group. Portal users may see different lists of available portlets on the Add Portlets page, depending on their privileges in the Portlet Repository page group.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Refreshing the Portlet Repository

- Updates registration information about providers and their portlets
- Updates the Portlet Repository page group
- Invalidates cache entries in Web Cache for pages that contain updated portlets



Refreshing the Portlet Repository

Path: [Providers](#) > **Registered Providers**

Type ▲▼	Name ▲▼	Actions	Creator ▲▼	Last Modified
Database Provider	QIP Provider	Edit Registration , Refresh , Grant Access	PORTAL	15-AUG-2005
Database Provider	Oracle Instant Portals	Edit Registration , Refresh , Grant Access	PORTAL	15-AUG-2005

Refreshing individual providers

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Refreshing the Portlet Repository

The Portlet Repository portlet enables you to display the Portlet Repository, refresh the Portlet Repository, and view the Portlet Repository Refresh Log.

When you refresh the Portlet Repository, registration information about providers and their portlets in the Portlet Repository is updated. The process of updating a provider in the Portlet Repository involves creating portlet items for each new portlet along with the portlet translations. Exceptions raised during the refresh process are captured in the Refresh Log. The pages that contain updated portlets have their page caches invalidated.

Displaying the Portlet Repository Page Group

From the Portal Navigator

Oracle Application Server Portal Navigator

Page Groups

These are the page groups available to you. The Shared Objects page group contains objects that are groups.

Create New... [Page Group](#) Find

Path: **Page Groups**

Type ▲▼	Name ▲▼	Actions	Create
Page Group	Portal Design-Time Pages	Properties , View Root Page , Edit Root Page , Copy Root Page , Convert Root Page to Template	PORTAL
Page Group	Portlet Repository	Properties , View Root Page , Edit Root Page , Copy Root Page	PORTAL

From the Portlet Repository portlet

Portlet Repository

- [Portlet Builders](#)
Portlets for building reports, charts, and forms from different data sources, including databases, Web Services, and XML files.
- [Portal Content Tools](#)
Portlets for viewing, searching, and managing Portal content.
- [Published Portal Content](#)
Portlets for published content organized by page groups.
- [Portlet Staging Area](#)
Newly registered and uncategorized providers and portlets.
- [Portal Community News](#)
Portlets for staying informed of the latest product, technical, and community news.
- [Administration Portlets](#)
Portlets for administering Portal, SSQ, OID, Oracle database, and Oracle Reports.
- [Shared Portlets](#)
Portlets with customizations that are shared across pages.

Portlet Repository

[Display Portlet Repository](#)
View, and optionally refresh, all local and remote portlets.

[Refresh Portlet Repository](#)
Refresh information about all the portlets in the repository.

[View Portlet Repository Refresh Log](#)
View the log entries entered during the refresh of the portlet repository.

Copyright © 2005, Oracle. All rights reserved.

Displaying the Portlet Repository Page Group

The Portlet Repository page group displays information about portlets available in the OracleAS Portal instance. Similar to any other page group in OracleAS Portal, the Portlet Repository page group is organized in the form of hierarchy of portal pages. The Portal Repository pages display information about available portlets as portlet items.

You can view the Portlet Repository page group from the Portal Navigator or from the Portlet Repository administrative portlet.

Organizing the Portlet Repository Page Group

You can organize the Portlet Repository page group as required by:

- **Creating standard pages**
- **Moving portlet items between pages**
- **Rearranging portlet items on the page**
- **Editing the Portlet Repository style**
- **Editing the Portlet Repository template**



Copyright © 2005, Oracle. All rights reserved.

Organizing the Portlet Repository Page Group

After the installation, the Portlet Repository page group consists of pages that display information about portlets that are shipped with OracleAS Portal (for example, the administrative portlets). When you register a new provider, a new portal page is created under the Portlet Staging Area page in the Portlet Repository page group. The new page's name is the same as the provider's display name; the new page also contains portlet items for each of the provider's portlets.

You can organize content of the Portlet Repository page group to help portal users browse the Portlet Repository easily, or you can apply a corporate look-and-feel style to the Portlet Repository pages. For example, you can create additional pages within it, rearrange portlet items on pages in the Portlet Repository, move portlet items between pages within the Portlet Repository, change the appearance of the Portlet Repository by editing the Portlet Repository style, or change the layout of all the pages in the Portlet Repository by editing the layout of the Portlet Repository page template.

Securing the Portlet Repository Page Group

The screenshot shows two overlapping web forms. The top form is titled 'Edit Page: My Provider' and has tabs for Main, Template, Access, Items, Optional, Parameters, and Events. The 'Access' tab is selected. Below the tabs, there are 'Apply', 'OK', and 'Cancel' buttons. The 'Access Settings' section contains a text box with the instruction: 'Decide whether to inherit access settings from the page's template, the page's parent, or to define new access settings for this page.' The bottom form is titled 'Edit Portlet: My Portlet' and has tabs for Home, Builder, Navigator, and Help. The 'Builder' tab is selected. Below the tabs, there are 'Apply', 'OK', and 'Cancel' buttons. The 'Item Attributes' section contains a text box with the instruction: 'Enter a display name for the item's link text which appears in the page area. Select a category that best describes the item's content, then enter a description. The URL, provider ID, provider name, and portlet ID cannot be changed.' Below this instruction are several input fields: 'URL' (PORTAL.wwpob_page.render_portlet_screer), 'Display Name' (My Portlet), 'Image URL' (empty), 'Category' (General), '* Provider ID' (91286076), '* Provider Name' (My Provider), and '* Portlet ID' (1). A large diagonal watermark 'HERNAN RAUL (hernan.raul@gyt.com.gt) has a non-transferable license to use this Student Guide.' is visible across the forms.

ORACLE

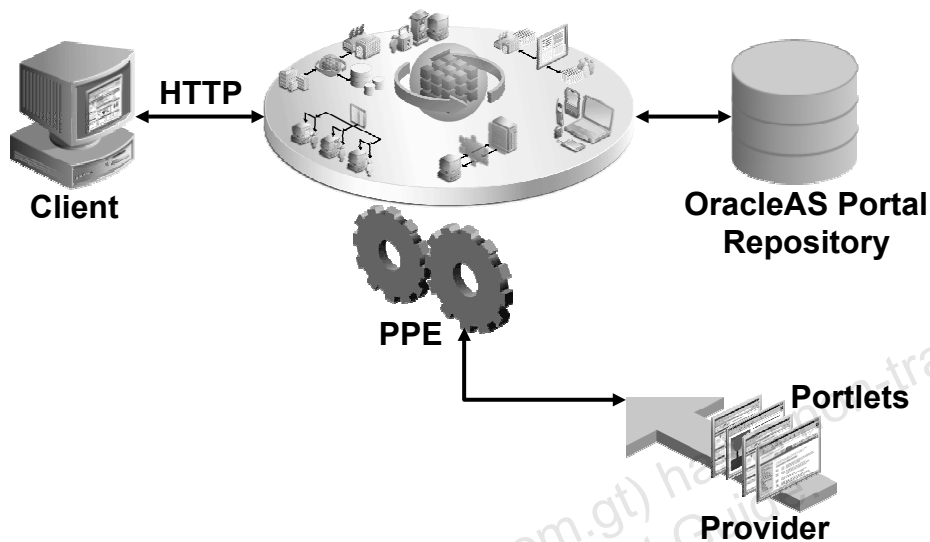
Copyright © 2005, Oracle. All rights reserved.

Securing the Portlet Repository Page Group

Users are allowed to see those providers and portlets for which they have the necessary privileges to view. In addition to the ability of controlling access to providers by editing their registration information, you can also secure information about available portlets by controlling the user access to the Portlet Repository page group. In particular, you can control which users can access pages in the Portlet Repository page group. This is done by granting privileges in the Access Settings section of the Access tabbed page for the Portlet Repository pages.

You can also control security at the portlet level, which controls who can see the portlet on a page. This can be done by editing the portlet item access privileges in the Portlet Repository on the Edit Portlet page.

Registering a Provider



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Registering a Provider

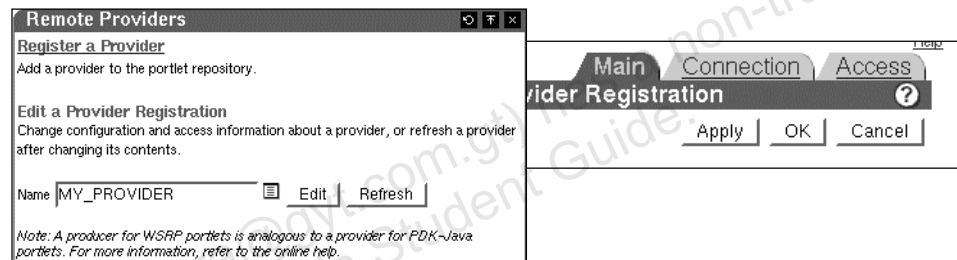
Before using the portlets, you need to register a provider with the portal. Provider registration is the process by which OracleAS Portal is informed about how the provider can be accessed. Providers are registered through the OracleAS Portal Web user interface. After you register a provider, the provider and its portlets become available in the Portlet Repository. They are also listed in the OracleAS Portal Navigator. Portlets can be deployed to OracleAS Portal through three types of providers: Web providers, WSRP producers, and database providers.

When you register a new provider, you define the provider connection information that specifies how the provider can be contacted by the portal. The provider returns registration information to the portal, which includes the list of the provider portlets and their attributes. During the registration, the provider can also perform provider-level initializations. For example, the provider can load error messages and strings that are used in its portlets into the Portal Repository. The portal saves the provider registration information in the Portlet Repository, creates a new page in the Portlet Repository page group, and adds portlet items for each of the provider portlets to that page. Finally, the portal grants the Manage privilege on the provider to the user who registers the provider and sets the provider status to ONLINE.

Updating the Provider Registration Information

You can use the Remote Providers portlet to:

- Change the display name of the provider
- Update connection information of the provider
- Grant and change access to the provider
- Change the provider status
- Clear the Web Cache entries for the provider



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Updating the Provider Registration Information

The Remote Registration portlet enables you to update the registration information of a previously registered provider in the Portlet Repository. You can either enter the provider name in the Name field or select the provider from the pop-up list that is displayed when you click the List icon next to the Name field. By clicking the Edit button, you open the Edit Provider Registration page that has three tabbed pages on which you can make the following changes:

- **Main:** Enables you to change the display name of the provider. You can also enter information about how long to wait for a response from the provider.
- **Connection:** Enables you to edit the connection information of the provider
- **Access:** Enables you to control security of the provider by granting access to the provider to portal users and groups. You can also take a provider offline when the provider is temporarily unavailable. If the provider is offline, the portal does not contact the provider until the provider comes back online. In the Cache Invalidation section, you can clear the Web Cache entries for the provider. This is required if you have changed the privileges for the provider to make sure that those changes are effective immediately.

Updating the Provider Registration Information (continued)

Note: If you need to update a WSRP type of provider, the Edit Provider Registration page provides the following two additional tabs:

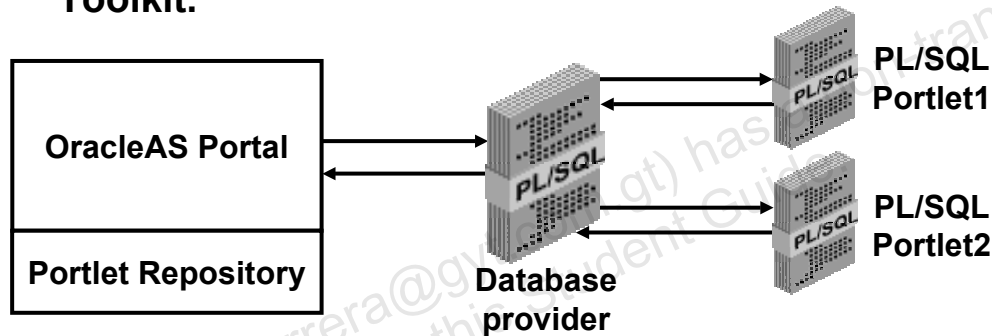
- **Properties tab:** Enables you to enter values for registration properties. If your producer has registration properties, you can add or change the values here.
- **User Categories tab:** Enables you to map WSRP producer user categories to portal groups. Some standard user categories are mapped to Portal groups by default.

Note: In a production environment, the provider deployment and registration process is performed by a portal administrator.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Database Providers and PL/SQL Portlets

- Database providers are PL/SQL packages that communicate with OracleAS Portal.
- PL/SQL portlets are program units that implement business logic and produce HTML output.
- Database providers and PL/SQL portlets use APIs from Portal Developer Kit (PDK) and Web PL/SQL Toolkit.



Copyright © 2005, Oracle. All rights reserved.

Database Providers and PL/SQL Portlets

In the OracleAS Portal architecture, the portal never communicates with a portlet directly. Instead, it talks to the provider. The database provider is a PL/SQL package that implements communication methods that are required by OracleAS Portal. These methods are used to retrieve information about or to display the provider's portlets.

A PL/SQL portlet is a collection of program units organized in a PL/SQL package that implements business logic and produces the required HTML output to be displayed on a portal page.

PL/SQL portlets communicate with OracleAS Portal through a database provider. You have to register the database provider explicitly. Database providers are implemented in PL/SQL and deployed in the Oracle database where OracleAS Portal is installed.

Note: Data-driven portlets, built with Portlet Builder, communicate with OracleAS Portal through database providers. You do not need to register the Portlet Builder providers with OracleAS Portal explicitly; they are automatically registered by OracleAS Portal.

Portlet developers use APIs specified in the Portal Developer Kit (PDK) and the Web PL/SQL Toolkit to code database providers and PL/SQL portlets.

Installing the Database Provider and Its PL/SQL Portlets

1. Create a schema to store PL/SQL packages:

```
SQL> CREATE USER ORADBxx IDENTIFIED BY pwd;  
SQL> GRANT CONNECT, RESOURCE TO ORADBxx;
```

2. Create synonyms to OracleAS Portal PL/SQL APIs:

```
SQL> CONNECT PORTAL/PORTAL_PWD  
SQL> @PROVSYSN.SQL ORADBxx
```

3. Install the PL/SQL packages in the schema:

```
SQL> CONNECT ORADBxx/pwd  
SQL> @MY_PROVIDER.SQL
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Installing the Database Provider and Its PL/SQL Portlets

To install the database provider and its portlets, perform the following tasks:

1. Create a provider schema to store PL/SQL packages of the database provider and its PL/SQL portlets. It is recommended that the schema is different from the schema where OracleAS Portal is installed.
2. Create synonyms to OracleAS Portal PL/SQL APIs stored in the portal schema. Log in to the portal schema and grant execute privilege to call the PL/SQL APIs to the provider schema. This can be done by running the `provsyns.sql` script that is located in the `$ORACLE_HOME/portal/admin/plsql/wwc` directory, where `$ORACLE_HOME` is the root directory in which OracleAS Portal is installed.
Note: You grant execute privileges on OracleAS Portal PL/SQL APIs only once per provider schema.
3. Install the PL/SQL packages of the database provider and its PL/SQL portlets in the provider schema. Log in to the database as the provider schema, and run the scripts to create the database provider and its PL/SQL portlets. For example, the `MY_PROVIDER.SQL` script may contain the PL/SQL code that creates packages for the database provider and its PL/SQL portlets.

Registering the Database Provider with OracleAS Portal

Register Provider

Step 1 of 2

Provider Information

Enter a unique name for the provider, the timeout (in seconds), the timeout message, and the provider implementation style. The timeout message is presented to the user when a portlet takes longer than the specified time to execute.

Note: A producer for WSRP portlets is analogous to a provider for PDK-Java portlets. For more information, refer to the online help.

* Name: MAP_PROVIDER

Display Name: MapProvider

Timeout: 30 seconds

Timeout Message: Map Provider has timed out.

Implementation Style: Database

Next > Cancel

Registering the Database Provider with OracleAS Portal

1. Log in to OracleAS Portal as a portal administrator, and click the Administer tab on the Portal Builder page.
2. Click the Portlets subtab.
3. Click the Register a Provider link in the Remote Providers portlet.
4. In the first step of the Register Provider Wizard, enter values for the provider properties:
 - **Name:** Is a unique name of up to 200 characters for the database provider
 - **Display Name:** Appears on the Add Portlets page with the provider's portlets listed under it
 - **Timeout:** Is the number of seconds OracleAS Portal should attempt to connect to this provider before displaying the timeout message
 - **Timeout Message:** Is the text of the message that you want to display when OracleAS Portal cannot establish contact with the database provider within the number of seconds specified in the Timeout field
 - **Implementation style:** Is the type of implementation style chosen for this provider: Database
5. Click Next to proceed to the second step of the wizard.

Registering the Database Provider with OracleAS Portal

Define Connection

Step 2 of 2 >>

General Properties
Register the name of the schema containing the provider implementation and the name of the package that implements its functionality.

* Owning Schema

* Package Name

User/Session Information
Specify the frequency by which the call to the provider for session initialization is performed.

Login Frequency

< Previous Finish Cancel

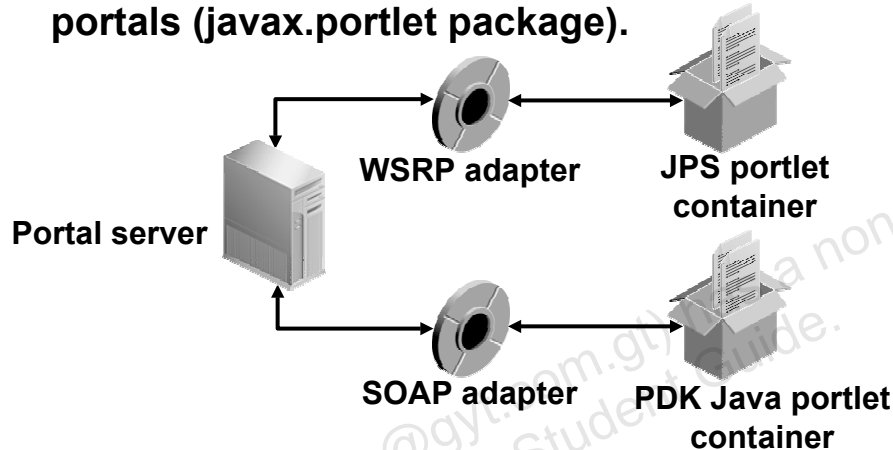
Copyright © 2005, Oracle. All rights reserved.

Registering the Database Provider with OracleAS Portal (continued)

6. The display of the second step of the wizard depends on the provider's implementation style that is selected in the first step. Enter values for the following properties required for a database provider:
 - **Owning Schema:** Specifies the provider schema
 - **Package Name:** Specifies the name of the PL/SQL package that implements the provider
 - **Login Frequency:** Determines the frequency of calls that OracleAS Portal makes to the provider to perform special processing before any portlet is executed. The value for this field is usually specified in the provider installation documentation. In most cases, the value should be set to Never.
7. Click Finish to complete the provider registration.

Using a WSRP Provider

- **WSRP is a communication protocol between portal servers and portlet containers.**
- **JSR 168 is a Java API for portlets to work with portals (javax.portlet package).**



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using a WSRP Provider

Web Services for Remote Portlets (WSRP) and JSR 168 standards enable the development of portlets that interoperate with different portal products and, therefore, widen the availability of portlets within an organization.

WSRP is a communication protocol between portal servers and portlet containers, whereas JSR 168 describes the Java Portlet API for building portlets. Combining these standards enables you to integrate applications from any internal or external source as portlets with WSRP portals.

WSRP is a standard that can be used for communication between portlets and OracleAS Portal. It is a Web services standard that enables integration of visual, user-facing Web services with portals or other intermediary Web applications. WSRP enables interoperability between the standards-enabled containers and any WSRP portal. These containers can be based on any particular language, such as JSR 168, .NET, or PERL. In addition, a portlet deployed to a WSRP-enabled container can be rendered on any portal that supports this standard.

WSRP producers contain information specific to them, such as the WSDL URL and the session-handling information supplied by the producer.

Registering a WSRP Provider

Register Provider

Step 1 of 2 >>>

Provider Information

Enter a unique name for the provider, the timeout (in seconds), the timeout message, and the provider implementation style. The timeout message is presented to the user when a portlet takes longer than the specified time to execute.

Note: A producer for WSRP portlets is analogous to a provider for PDK-Java portlets. For more information, refer to the online help.

* Name: MyWSRPProvider

Display Name: WSRP Provider

Timeout: 25 second

Timeout Message: This Provider has t...

Implementation Style: WSRP

Define Connection

Step 2 of 3 >>>

Provider Description

Specify the URL to a WSDL document that contains an abstract description of provider. Clicking Next will fetch the WSDL document and request the provider service information.

* WSDL URL: http://portalstandards.oracle.com/wsrf/jaxrpc?WSDL

Proxy Settings

There are no proxies currently specified in the Proxy Settings Page.

Registering a WSRP Provider

To register a WSRP provider with OracleAS Portal, perform the following steps:

1. Log in to OracleAS Portal as a portal administrator, and click the Administer tab on the Portal Builder page.
2. Click the Portlets subtab.
3. Click the Register a Provider link in the Remote Providers portlet.
4. In the first step of the Register Provider Wizard, enter values for the provider properties, such as Name, Display Name, Timeout, and Timeout Message, and select the implementation style as WSRP.

5. Click Next to proceed to the second step of the wizard.
6. Enter the URL of the Web Services Description Language (WSDL) document that contains the description of the WSRP producer in the WSDL URL field.

When you click Next, Portal fetches the WSDL document and retrieves the producer information.

7. If your producer contains registration properties, you will be able to enter values for those properties on this screen. Click Finish.

Note: Two additional steps are further involved for adding any required property values and setting the security for the providers.

Registering a WSRP Provider (continued)

For more information about the WSRP implementation style for Web providers, see the *Oracle Application Server 10g Release 2 (10.1.2) Portlet Developer Guide*.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

What Are Web Providers?



ORACLE

Copyright © 2005, Oracle. All rights reserved.

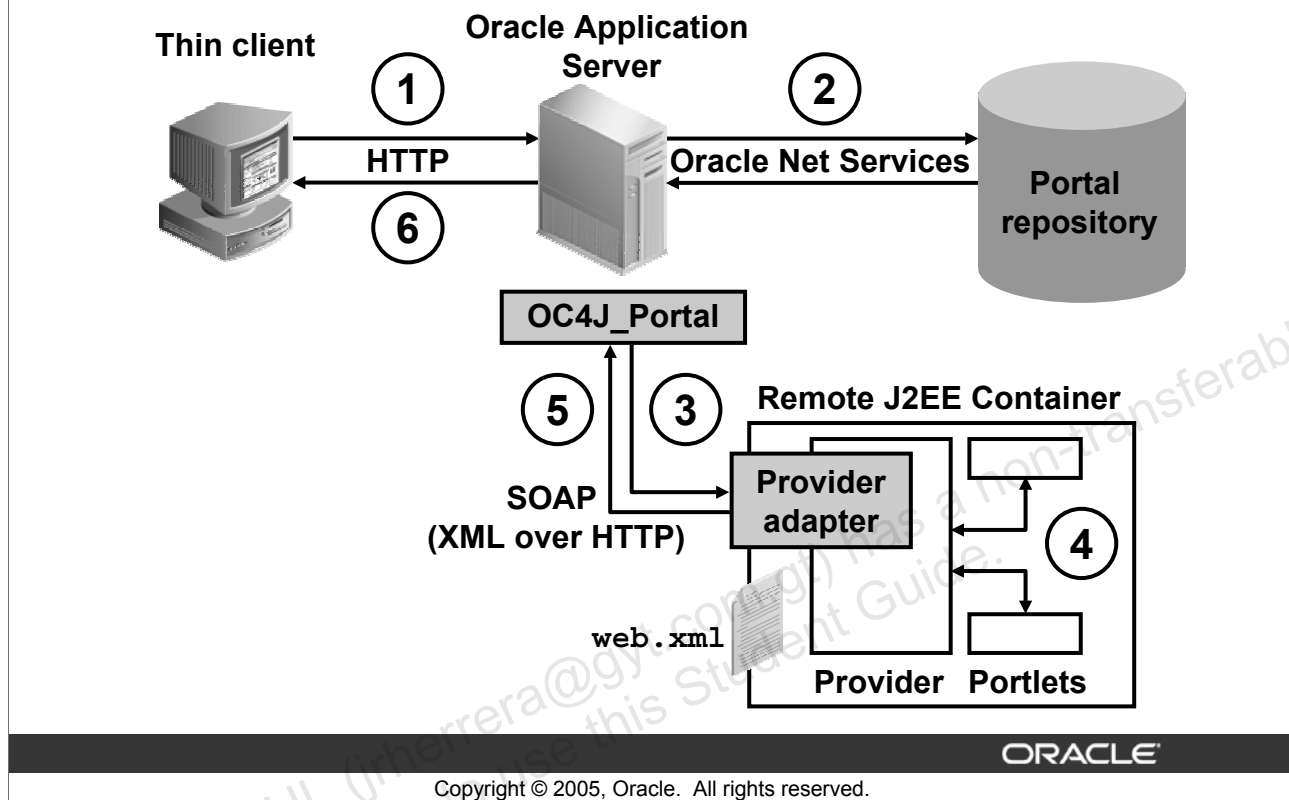
What Are Web Providers?

Web providers are regular J2EE applications and are distributed by portlet developers in the form of Enterprise Archive (EAR) files. Web providers may reside on the same application server as OracleAS Portal, on a remote application server, or anywhere on the network. A Web provider could be implemented using virtually any Web technology. However, the Oracle Application Server Portal Developer Kit provides a Java framework that simplifies the task of building Web providers. Because Web providers can be deployed to a J2EE container, they do not put an additional load on the OracleAS Portal Repository database.

Note: You can deploy Web providers to the Oracle Application Server Middle Tier just like any other J2EE application.

Web providers use open standards, such as XML, SOAP, HTTP, or J2EE, for deployment, definition, and communication with OracleAS Portal. To expose your portlets by using a Web provider, you must create a provider that manages your portlets and can communicate with OracleAS Portal by using SOAP.

Accessing Web Providers



Accessing Web Providers

The slide shows the flow of the requests that are made to a Web provider:

1. The request to display a portal page arrives at the OracleAS Middle Tier instance from the client browser.
2. The Parallel Page Engine (PPE), running as a servlet on the middle-tier OC4J_Portal instance, retrieves the page metadata information from the Portal Repository.
3. The PPE contacts the Web provider through the provider adapter for the portlet content using SOAP over HTTP.
Note: The provider adapter translates the SOAP requests to native, usually Java, and calls to hide the complexity of the SOAP communication from the provider.
4. The Web provider makes the necessary calls to the portlets so that they generate the HTML or XML output.
5. The Web provider returns the HTML or XML code to the PPE.
6. The PPE assembles the page and Oracle Application Server returns the portal page to the client browser.

Testing Web Providers

`showTestPage=true`



`showTestPage=false`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Testing Web Providers

The testing of Web Providers is part of the deployment process. You can test the provider to make sure that the deployment is successful. Enter the URL of the provider adapter servlet:

`http://host name:port/context_root/providers`

If your deployment succeeds, the test page is returned to the browser.

Note: Access to the provider test page can be denied. You can specify the default value of the `showTestPage` JNDI variable in the `web.xml` file:

```
<env-entry>
  <env-entry-name>
    oracle/portal/sample/showTestPage
  </env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>true</env-entry-value>
</env-entry>
```

If the test page is disabled, a “403 Forbidden” response is returned to the test page request.

Registering Web Providers: Provider Information

The screenshot shows two overlapping windows from the Oracle Register Provider wizard. The background window is titled 'Register Provider' and is at 'Step 1 of 2'. It contains a 'Provider Information' section with the following fields: Name (MyFirstWebPr), Display Name (My First Web), Timeout (30 seconds), Timeout Message (My First Web), and Implementation Style (Web). The foreground window is titled 'Define Connection' and is at 'Step 2 of 2'. It contains a 'General Properties' section with the following fields: URL (http://139.185.35.116:7776/portlet_demos/providers) and Service ID. A tip at the bottom of the foreground window states: 'TIP (example: "urn:ADAPTER_PROVIDER" or "urn:webProvider")'.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Registering Web Providers: Provider Information

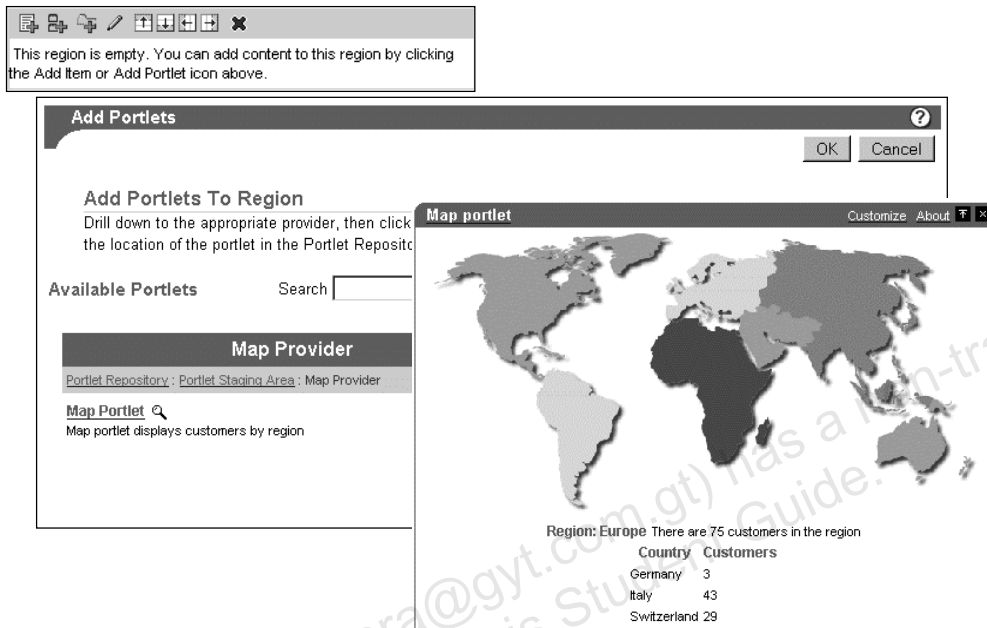
Web providers contain information specific to Web providers, such as the URL of the provider, the user's identity communicated to the provider, and proxy information. Web providers can reside on any host that is accessible from the OracleAS Portal. Besides the URL, the administrator must specify whether the portlet provider takes advantage of the session-handling PDK-Java service.

To register a Web provider, perform the following steps:

1. In the Remote Providers portlet, click the Register a Provider link to launch the Register Provider Wizard.
2. Enter values for the provider properties, such as Name, Display Name, Timeout, and Timeout Message, and select the Implementation Style as Web.
3. Click Next to proceed to the second step of the wizard.

In the final step of the wizard, you define the provider's general properties of which the URL property is the only property that depends on the deployed environment. This is the URL that you used for testing the Web provider, omitting the Web service name from the end of the URL. The values for the other general properties in this step should be provided by the portlet developer in the Web provider installation instructions.

Adding the Portlet to a Portal Page



Copyright © 2005, Oracle. All rights reserved.

Adding the Portlet to a Portal Page

As a result of the installation and registration process, the PL/SQL portlet can be added to a portal page. To add the portlet to a page, perform the following steps:

1. Edit or customize the page. Select the portlet region in which you want the portlet to appear, and click the Add Portlet icon.
2. In the Add Portlets window, select the portlet that you want to display on your page. Click the portlet title link to add it to the region, and then click OK.
3. The portlet is displayed on your page.

Note: The Portlet Staging Area page of the Portlet Repository is where the portlets appear when a provider is first registered with the portal.

Exporting and Importing Objects in OracleAS Portal

- **Use the Export/Import utilities to:**
 - Support staging content on one or more OracleAS Portal development instances for deployment to OracleAS Portal production instances
 - Consolidate multiple OracleAS Portal instances
 - Deploy identical content across multiple OracleAS Portal instances
- **Perform the Export/Import process between source and target OracleAS Portal instances of the same version**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

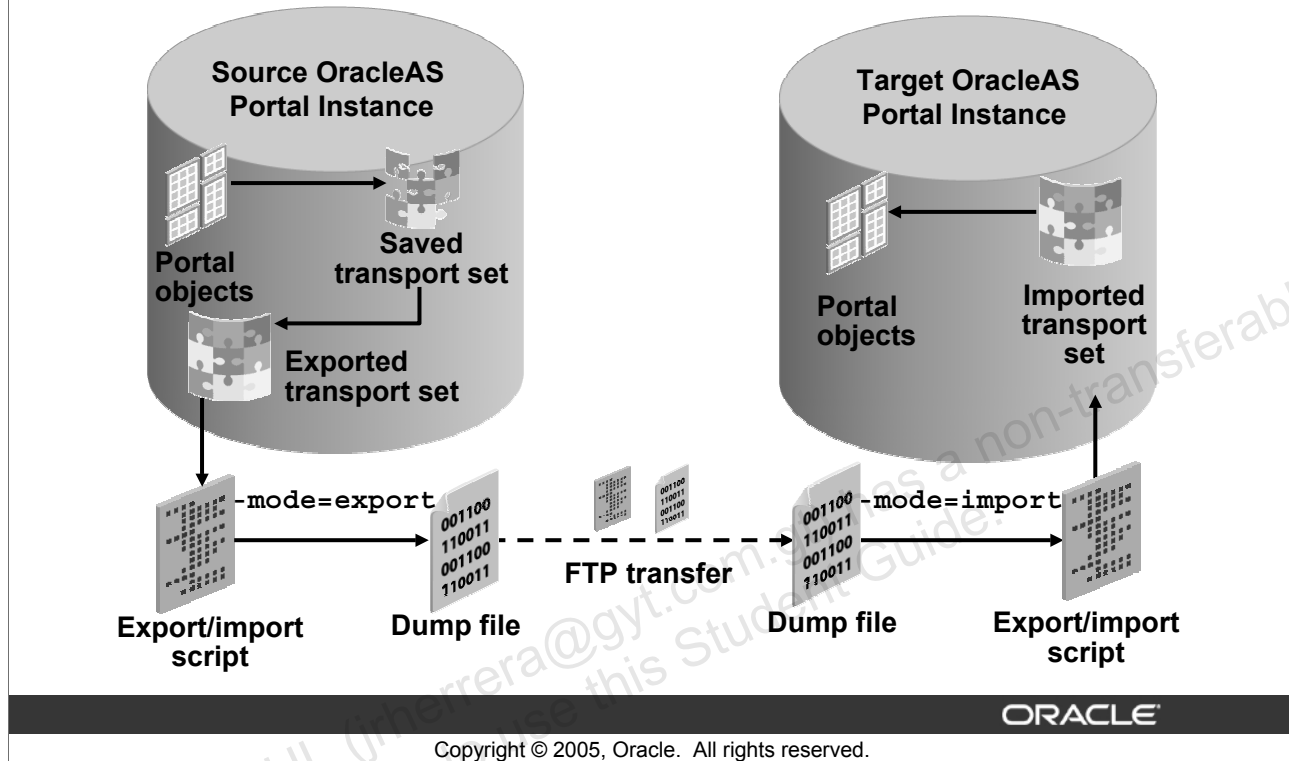
Exporting and Importing Objects in OracleAS Portal

OracleAS Portal provides a set of export/import utilities that enable you to migrate portal content between different OracleAS Portal instances. A typical example where these utilities would be used is to copy or update portal objects between a development instance and a production instance of OracleAS Portal.

Export/import utilities are also useful for consolidating multiple OracleAS Portal instances. Consolidation may be driven by a need to reduce the number of active instances or other business considerations. For example, you may want to merge your multiple OracleAS Portal instances to a single OracleAS Portal instance.

Another possible example is deploying identical content across multiple OracleAS Portal instances. In this case, the OracleAS Portal objects can be created in one instance and propagated to multiple instances using the export/import utilities.

Exporting and Importing Objects in OracleAS Portal



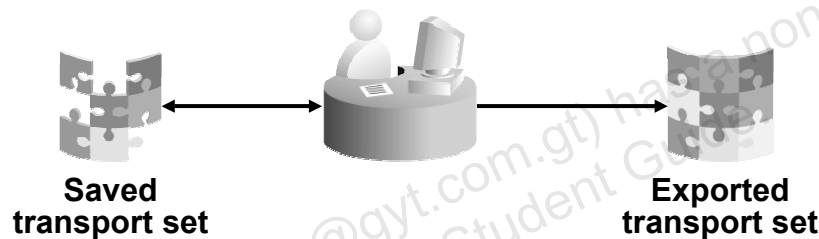
Exporting and Importing Objects in OracleAS Portal (continued)

The export and import process is a multistep process that you perform using the OracleAS Portal user interface and a set of command-line scripts:

1. In the source OracleAS Portal instance, create a transport set and populate it with the list of portal objects that you want to export to the target OracleAS Portal instance.
2. Export the transport set and generate the export/import script. During this step, the portal objects that are listed in the transport set are copied into the transport tables in the source OracleAS Portal instance.
3. Execute the export/import script in the EXPORT mode (`-mode=export`) to create a dump (.dmp) file that contains the transport set. The export/import script uses the `exp` database utility to export data from the transport tables.
4. Transfer the export/import script and the export dump file to a machine that hosts the target OracleAS Portal instance.
5. Import the transport set from the dump file by executing the export/import script in IMPORT mode (`-mode=import`). The export/import script uses the `imp` database utility to import data to the transport tables in the target OracleAS Portal instance.
6. In the target OracleAS Portal instance, merge portal objects from the transport set by using the Export/Import Transport Set portlet.

Creating a New Transport Set

1. In the Navigator, select a portal object and click the Export action.
2. Enter a meaningful title for the transport set.
3. Save the transport set for future editing or export the transport set.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Creating a New Transport Set

To create a transport set, open the Portal Navigator, select a portal object that you want to export, and click the Export action for this object. The top-level portal objects to export are page groups and providers. To export an individual object, such as a category, a style, or a perspective, the object's page group must already exist on the target OracleAS Portal instance. For this reason, the first export performed should migrate the entire page group from the source OracleAS Portal instance to the target OracleAS Portal instance.

When exporting page groups, all objects within the page group, as well as referenced shared objects, are exported. This includes pages, categories, perspectives, styles, custom types, Web providers, and access control lists associated with the page group. During the migration of the Web provider metadata, OracleAS Portal attempts to register the provider and its portlets. If the provider cannot be contacted during registration, then the provider is not migrated and a message is written to the import log file.

At the end of the creation process, you can immediately export the transport set to a file or you can save the transport set for editing it and exporting it at a later time.

Editing a Saved Transport Set

- **To edit a saved transport set from the Export/Import Transport Set portlet:**
 - Select a transport set from the list of available saved transport sets
 - Use the wizard-based interface to:
 - Modify the security of portal objects in the set
 - Remove nonrequired portal objects from the set
- **Add new portal objects to the set by selecting the Add to An Existing Transport Set option of the Export action from the Portal Navigator.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Editing a Saved Transport Set

You can edit a saved transport set by adding or removing portal objects, changing the name, and changing the security options before exporting the transport set.

To edit the transport set, you select the required set from the list of available transport sets in the Export/Import Transport Set portlet, and click Edit. A wizard-based interface enables you to include the security of the portal objects included in the set, as well as remove those objects that are not required.

To add a new portal object to the existing set, you choose the portal object from the Portal Navigator and select the corresponding option of the Export action.

Exporting a Transport Set

To export a transport set, perform the following steps:

1. Select a transport set from the list of available saved transport sets.
2. Export the transport set.
3. View the export log output.
4. Download the export/import script.
5. Run the script in EXPORT mode to generate a dump file.

```
expimp.csh -mode EXPORT  
           -d mycompany_portal.dmp  
           -c infra_db -s portal -p fs61qat9
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Exporting a Transport Set

After a transport set is ready to export, perform the following steps:

1. Select the transport set from the list of available saved transport sets.
2. Export the selected set by clicking Export Now in the Edit Transport Set Wizard. After the set is exported, the transport set is considered as complete and cannot be edited anymore.
3. When the export completes, the portal objects become available for migration. Review the export log for errors that may occur during the export.
4. OracleAS Portal uses the exp and imp database utilities in this migration. The wizard provides you with the export/import scripts for performing this operation from a Linux/UNIX shell or a Windows NT command window. Download the script that is relevant to your source and target operating system.

Exporting a Transport Set (continued)

5. Run the `export/import` script in `EXPORT` mode to generate a dump file that contains the transport set ready for migration. The following parameters should be defined to run the script:
 - `-mode` defines the `export/import` script mode. This parameter must be set to `EXPORT`.
 - `-d` defines the name of the dump file.
 - `-c` defines the connect string to the database that contains the source OracleAS Portal instance.
 - `-s` defines the name of the OracleAS Portal product schema. In a typical installation, this is the `portal` schema.
 - `-p` defines the password for the OracleAS Portal product schema. The password is randomized and can be extracted from Oracle Internet Directory.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Importing the Transport Set

1. Run the script in **IMPORT** mode to load the dump file into the target OracleAS Portal instance.

```
expimp.csh -mode IMPORT -d mycompany_portal.dmp
            -company mycompany
            -c infra_db -s portal -p fs61qat9
            -pu mc_admin -pp mc123
```

2. Select the transport set from the list of transport sets ready for import.
3. Select the import mode.
4. Import the transport set.
5. Analyze the import log for possible errors.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Importing the Transport Set

After you transfer the export/import script and the dump file to the target machine, perform the following steps:

1. Run the export/import script in **IMPORT** mode. This loads the dump file into the target Portal Repository using the **imp** database utility. The following parameters should be defined to run the script:
 - **-mode** defines the export/import script mode. This parameter must be set to **IMPORT**.
 - **-d** defines the name of the dump file.
 - **-company** defines a company name in the configuration that provides hosted environment. Default value is **NONE**.
 - **-c** defines the connect string to the database that contains the target OracleAS Portal instance.
 - **-s** defines the name of the OracleAS Portal product schema. In a typical installation, this is the **portal** schema.
 - **-p** defines the password for the OracleAS Portal product schema. The password is randomized and can be extracted from Oracle Internet Directory.
 - **-pp** defines the password for the portal user defined by the **-pu** parameter.
 - **-pu** defines the name of a portal user to log in to the target OracleAS Portal instance.

Oracle Application Server 10g R2: Administration I 13-52

Importing the Transport Set (continued)

2. Log in to the target OracleAS Portal instance as a portal administrator, click the Administer tab on the Builder page, select the imported transport set from the list of transport sets ready for import in the Export/Import Transport Set portlet, and click Import.
3. Before starting the import, the Import Transport Set Wizard enables you to set the import mode. There are three import modes available. In Overwrite mode, the existing objects with the same names as objects in the transport set are overwritten. If you want to ignore warnings raised during the import, then select Ignore Warnings During Import. This isolates objects that prompt errors and allows the import of successful objects. It is recommended to initially run the import in Check-only mode. This allows you to view potential conflicts, warnings, or errors before changes are made in the target portal node. This also provides information about which objects are overwritten or reused. From this, you can decide whether to run the import in overwrite mode or not.
4. Start import and perform periodical checks on the process by viewing the import log. When finished, analyze the log file for possible errors.

Browsing Transport Sets

- View the status of the transport sets in the OracleAS Portal instance.
- View the log of import and export actions.
- Delete transport sets from the OracleAS Portal instance.

Actions: Delete Reuse						
<input type="checkbox"/>	Name	Owner	Status	Last Updated	Unix Script	NT Script
<input type="checkbox"/>	PMOSKOVI DocLib2	PMOSKOVI	Export Complete	07-AUG-03	PMOSKOVI DocLib2	PMOSKOVI DocLib2
<input type="checkbox"/>	PMOSKOVI DOCLIB1	PMOSKOVI	Import Complete	22-JUL-03		
<input type="checkbox"/>	My Company Portal	ORCLADMIN	Import Complete	11-SEP-03		
<input type="checkbox"/>	PMOSKOVI - 27-AUG-2003 19:08:39	PMOSKOVI	Pre-check Failed	27-AUG-03		
<input type="checkbox"/>	PMOSKOVI DocLib1	PMOSKOVI	Export Complete	27-AUG-03	PMOSKOVI DocLib1	PMOSKOVI DocLib1
<input type="checkbox"/>	My Company Portal	PORTAL	Export Complete	12-SEP-03	My Company Portal	My Company Portal

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Browsing Transport Sets

In addition to creating, editing, exporting, and importing transport scripts, you can also view the list of transport sets existing in the OracleAS Portal instance and check their current statuses. You can also view the log of actions, view referenced objects, and download export/import scripts for transport sets with a status of Export Complete by clicking the appropriate links. Additionally, you can delete transport sets from the system or reuse a transport set.

Summary

In this lesson, you should have learned how to:

- Describe OracleAS Portal administrative services
- Describe tools to monitor the OracleAS Portal instance
- Manage OracleAS Portal users and groups
- List OracleAS Portal schemas
- Administer the Portlet Repository
- Deploy portlets to OracleAS Portal by using Web providers, WSRP producers, and database providers
- Perform export and import of portal content

ORACLE

Copyright © 2005, Oracle. All rights reserved.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

14

Configuring OracleAS Portal

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Describe the OracleAS Portal configuration tasks**
- **Configure the self-registration feature to enable users to create their own portal accounts**
- **Configure OracleAS Portal for Web-based Distributed Authoring and Versioning (WebDAV)**
- **List the configuration modes**
- **Configure language support**
- **Configure dependencies of the OracleAS Portal instance by using the Portal Dependency Settings file**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal Configuration Tasks: Overview

The OracleAS Portal configuration tasks include:

- **Setting up the self-registration and search features**
- **Configuring language and mobile support**
- **Configuring OraDAV support for OracleAS Portal access**
- **Relinking the OracleAS Portal instance with other Oracle Application Server components**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal Configuration Tasks: Overview

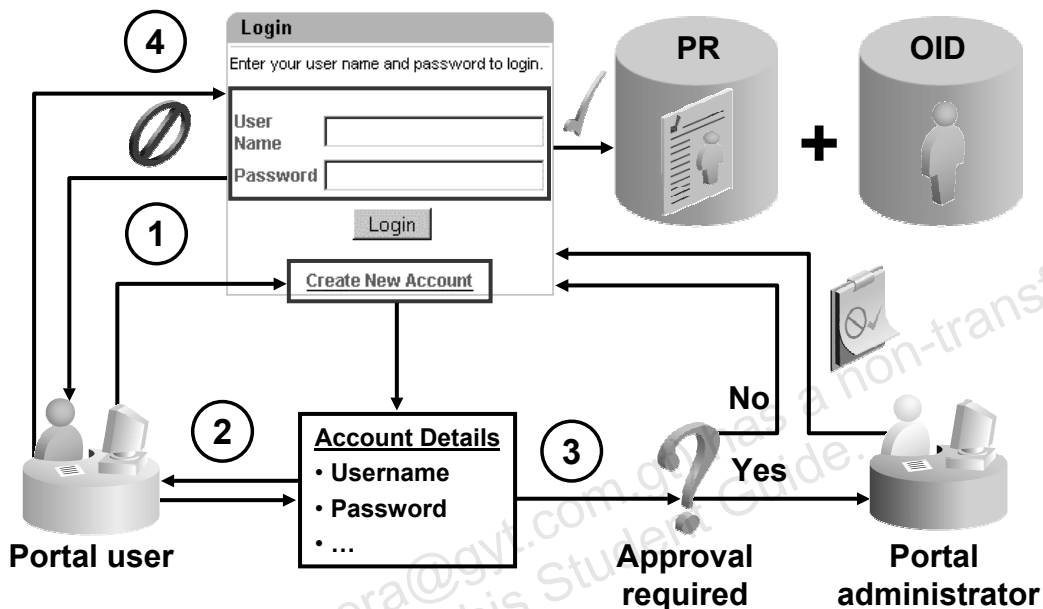
The OracleAS Portal framework provides administrative services that enable you to perform configuration tasks after the installation is complete. For example, these tasks may include:

- Setting up the self-registration and search features
- Configuring language and mobile support
- Configuring OraDAV support for OracleAS Portal access
- Relinking the OracleAS Portal instance with other Oracle Application Server components

As with administrative services, to perform most of the configuration tasks, you use the administrative user interface in OracleAS Portal and Oracle Application Server Control. You can also use the configuration scripts that were copied into the Oracle Home directory of the middle tier during the installation of Oracle Application Server that includes OracleAS Portal.

For further details about setting up the search feature and configuring mobile support, refer to the *Oracle Application Server Portal Configuration Guide*.

Self-Registration Feature in OracleAS Portal



Copyright © 2005, Oracle. All rights reserved.

Self-Registration Feature in OracleAS Portal

One of the features that you as a portal administrator may want to implement is to enable end users to create their own portal accounts. The diagram in the slide shows the flow of actions when a public user requests for a portal account:

1. The user opens a portal page, which contains the Login portlet, and clicks the Create New Account link.
2. The Self-Registration form is displayed. The user enters the preferred username, password and e-mail address, and optional personal information, such as first name, last name, and work phone number.
3. Depending on whether the user request requires an approval or not, a portal account can be created immediately after the request submission or the request is sent to the portal administrator for approval. If the portal administrator approves the request, the portal account is created and the user can log in to the portal. If the portal administrator rejects the user request, then the user is denied access to the portal. The user is always notified by e-mail when the request is approved or rejected.
4. The user logs in to the portal by using his or her username and password.

Configuring the Self-Registration Feature in OracleAS Portal

The screenshot shows the Oracle Application Server Portal Builder interface. At the top, there's a header with 'Oracle Application Server Portal' and 'Portal Builder'. Below this is a navigation bar with 'Portal', 'Portlets', and 'Database'. The 'Services' portlet is active, showing a list of services including 'Global Settings', 'SSO Server Administration', and 'User'. The 'Global Settings' service is selected, displaying the 'Self-Registration Options' configuration page. This page has a tabbed interface with 'Main' and 'E-Mail (SMTP) Host' tabs. The 'Main' tab is active, showing options to 'Enable Self-Registration' (checked), 'Approval Required' (radio button), and 'No Approval Required' (radio button). There are links for 'Configure' and 'Create'. The 'E-Mail (SMTP) Host' tab is also visible, with instructions on how to enter the host name and port.

Oracle Application Server Portal

Portal Builder

Welcome

Portal Portlets Database

Services SSO Server Administration User

Global Settings Edit SSO Server Configuration Create

Self-Registration Options

Select whether to allow users to register their own user accounts. Select whether self-registered users need to be approved before they are able to log on, and click Configure to specify the approval process. If approval is not required, self-registered users can log on immediately after registering.

☒ Enable Self-Registration

☐ Approval Required [Configure](#)

☐ No Approval Required

E-Mail (SMTP) Host

Enter your e-mail (SMTP) host name and port so that self-registered users can be contacted via e-mail when their accounts are approved. You can obtain this information from your e-mail server administrator. The default port for SMTP is usually 25. If you enable self-registered users to log on immediately, this information is not needed.

Configuring the Self-Registration Feature in OracleAS Portal

To configure the self-registration feature in the OracleAS Portal instance, perform the following steps:

1. Click the Global Settings link in the Services portlet. A tabbed page is displayed.
2. In the Self-Registration Options section on the Main tabbed page, enable the feature by selecting the corresponding check box. If you decide that the user's request should be approved, then you select the Approval Required option to establish the approval process. If no approval is required, then the user can log in to the portal immediately after registering.
3. If you choose the Approval Required option, enter a host name and port in the E-Mail (SMTP) Host section, so that when the account has been approved or rejected, the user is notified via e-mail.
4. Confirm your changes by clicking OK.

Enabling the Self-Registration Feature in the Login Portlet

Edit the default settings of the Login portlet to:

- Enable the self-registration feature link
- Define the text and URL of the self-registration link

The screenshot displays the OracleAS Portal interface. At the top, there is a 'Self-Registration' portlet with a 'Create' button and a 'Cancel' button. Below it, the 'Account Details' section is visible. The 'Login' portlet is prominently displayed in the foreground, featuring a text input field for 'User Name', a password input field for 'Password', and a 'Login' button. Below the password field, there is a link labeled 'Create New Account'. The Oracle logo is visible in the bottom right corner of the page.

Enabling the Self-Registration Feature in the Login Portlet

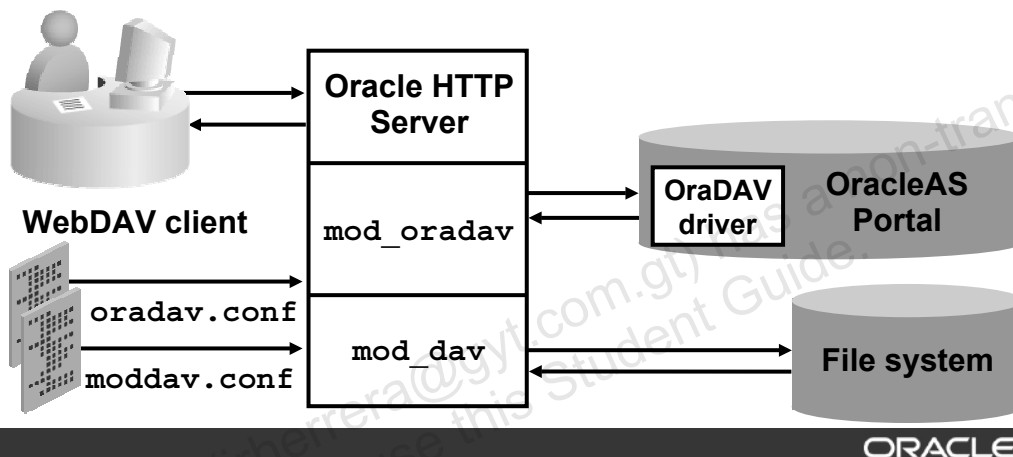
After you configure the self-registration feature in the OracleAS Portal instance, the self-registration link can be displayed in the Login portlet, which requires additional configuration steps on the portlet instance level. Typically, for a production portal that is imported into the OracleAS Portal instance, these configuration steps are performed by the page designer who builds the portal.

As a portal administrator, you may also want to add another instance of the Login portlet to a page accessible by a public user and enable the self-registration link in that portlet. To accomplish this, you need to edit the Login portlet default settings as follows:

- Enable the Self-Registration feature in the portlet.
- Optionally, change the link text and the URL for the Self-Registration form if you are not using the standard user interface provided by OracleAS Portal.

OraDAV Architecture

- **WebDAV is a protocol extension to HTTP 1.1 that supports distributed authoring and versioning.**
- **OraDAV extends implementation of WebDAV to support connections to an Oracle database.**



Copyright © 2005, Oracle. All rights reserved.

OraDAV Architecture

Web-based Distributed Authoring and Versioning (WebDAV) is a protocol extension to HTTP 1.1 that supports Web-based collaboration. With WebDAV, the Internet becomes a transparent read-and-write medium, where content can be checked out, edited, and checked in to a URL address. **mod_dav** is Apache's native implementation of WebDAV that supports read-and-write access to local files.

The **mod_oradav** module for Oracle HTTP Server extends implementation of **mod_dav** to support connections to an Oracle database to read and write the content, and query and lock documents in various schemas. The Oracle database must have an OraDAV driver installed. **mod_oradav** calls this driver to map the WebDAV activity to the database activity.

When Oracle Application Server is installed, all the required OraDAV parameters are set with values that are designed to enable the Oracle database content to be accessed through a Web browser or a WebDAV client. If the default values do not meet your needs, you can modify the values for required parameters and specify values for optional parameters.

Configuring OraDAV Support for OracleAS Portal Access

The parameters in the `oradav.conf` file specify:

- **DB connection**
- **OraDAV driver**
- **Password and package name**

```
<Location /dav_portal/portal>
DAV Oracle
DAVParam ORASERVICE cn=iasdb,cn=oraclecontext
DAVParam ORAUSER portal
DAVParam ORACRYPTPASSWORD
        BQtXpWPMebG29ifH3Mrw7mQrOtqk0utDvw==
DAVParam ORAPACKAGENAME
        portal_schema.wwdav_api_driver
</Location>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring OraDAV Support for OracleAS Portal Access

The OraDAV configuration parameters are stored in the `oradav.conf` file and start with `DAV` and `DAVParam`. These parameters are specified within a `<Location>` directive. The `oradav.conf` file is included in the `httpd.conf` file in an `include` statement.

In a typical installation of Oracle Application Server, the `oradav.conf` file is located in the `$ORACLE_HOME/Apache/oradav/conf` directory.

After OracleAS Portal has been installed, the `oradav.conf` file is populated with a `<Location>` directive, which points to the portal schema. By default, the OracleAS Portal DAV URL is `http://host name:port/dav_portal/portal`. This URL enables WebDAV clients, such as Microsoft Web folders, to access portal data.

You can configure `mod_oradav` by using Application Server Control, which is the recommended way, or manually, by editing the `oradav.conf` file. Whenever you make changes to `dads.conf` or `oradav.conf`, to make the changes effective, you must restart Oracle HTTP Server and OC4J_Portal.

Configuring Language Support

OracleAS Portal supports 29 languages.

- **Use the `ptllang` tool:**
 - **To configure any additional languages that were not selected during initial OracleAS Portal installation**
 - **For each language that you want OracleAS Portal to support**

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

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Language Support

OracleAS Portal is configured with the languages that are selected in Oracle Universal Installer during the OracleAS Middle Tier installation. If you must install languages after you have installed OracleAS Portal, run the `ptllang` script. You must run the `ptllang` script for each language that you want OracleAS Portal to support.

Before you run the `ptllang` script:

- Set the `ORACLE_HOME` environment variable to the middle-tier Oracle Home in which OracleAS Portal is installed
- Run the `ptllang` script from `ORACLE_HOME` in which OracleAS Portal is installed (`Portal_ORACLE_HOME/assistants/opca`)

For example, to load the Dutch language strings into OracleAS Metadata Repository, execute the following:

```
ptllang.sh -s portal -sp portal -c  
host.domain.com:1521:dbServiceName -lang nl
```

Configuring Language Support (continued)

The parameters include the following:

- -s: The OracleAS Portal schema name
- -sp: The OracleAS Portal schema password
- -c: The connect string to the database hosting OracleAS Metadata Repository
- -lang: The abbreviation for the language to install
- -log: The directory that the log files write to

Note: If you want to install languages to a portal repository whose version is 10.1.4, then you must run the `ptllang.sh` script from the Metadata Repository Upgrade Assistant CD-ROM.

For more information, refer to the *Oracle Application Server Portal Configuration Guide 10g Release 2 (10.1.4)*.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Setting Language for a Portal Session

The Set Language portlet enables you to select:

- The language for the current portal session
- The territory for the selected language to determine localizations, such as date, currency, and decimal formats (only if enabled by the page designer)



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Setting Language for a Portal Session

After installation, the language can be set as a preferred language for a portal session. OracleAS Portal provides the Set Language portlet that displays a list of installed languages in the form of links. You set the language for the portal session by clicking the corresponding link. The current session language is shown as a highlighted link in the Set Language portlet. The Set Language portlet updates the `login_nls` cookie value. Therefore, the language is effectively tied to the session on the specific browser containing this cookie.

In addition to selecting a preferred language for a portal session, users may be able to choose the territory to use for their portal sessions. The territory selection is typically enabled by the page designer by setting the default properties of the Set Language portlet and, therefore, does not require additional configuration from the portal administrator. Choosing a territory determines localization settings, such as date, currency, and decimal formats. The territories are displayed as a list of links in a separate section of the portlet.

If you do not select the Enable Territory Selection check box while editing the Set Portlet Settings screen, then the territory defaults to the most common for the chosen language. The list of territories offered to users depends on the language they choose.

The database in which OracleAS Portal is installed should be created with Unicode (UTF8) as the character set to support multiple languages.

Configuring OracleAS Portal Dependencies

- OracleAS Portal stores its dependencies on the Oracle Application Server components in the Portal Dependency Settings file (the `iasconfig.xml` file).
- The Portal Dependency Settings file is located in the `$ORACLE_HOME/portal/conf` directory on the middle-tier machine.
- The Portal Dependency Settings tool (the `ptlconfig` script) updates OracleAS Metadata Repository with current settings in the `iasconfig.xml` file.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring OracleAS Portal Dependencies

OracleAS Portal is dependent on some of the Oracle Application Server components, such as OracleAS Web Cache and Oracle Internet Directory. It is important that you understand these dependencies because it may be necessary to fine-tune or configure these components after Oracle Application Server is installed.

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

You can use the Portal Dependency Settings file to:

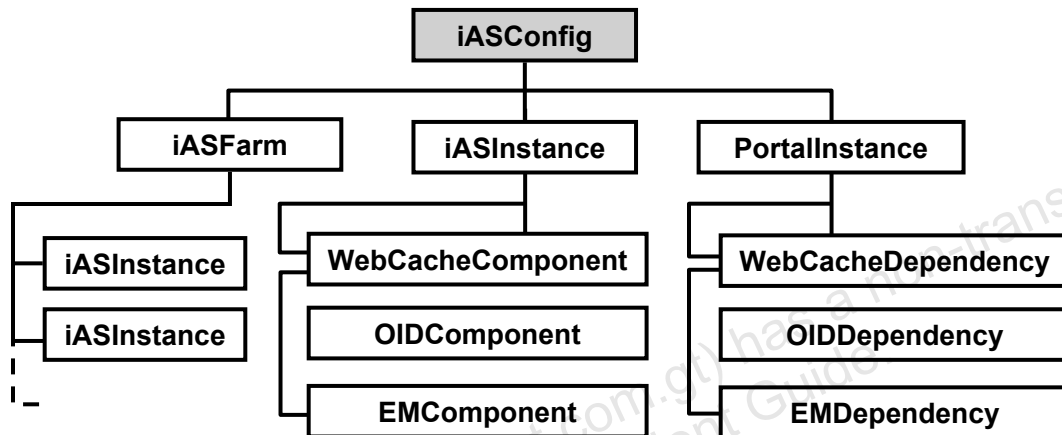
- Check the settings used by an OracleAS Portal instance
- Update the settings in OracleAS Metadata Repository

The name of the Portal Dependency Settings file is `iasconfig.xml`, and is located by default in `$ORACLE_HOME/portal/conf`, where `$ORACLE_HOME` is the OracleAS Middle Tier home directory.

To update OracleAS Metadata Repository with configuration settings in the `iasconfig.xml` file, you run the Portal Dependency Settings tool (the `ptlconfig` script) that is located in the same directory.

Portal Dependency Settings File

The `iasconfig.xml` file structure:



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Portal Dependency Settings File

The Portal Dependency Settings file is an XML file that is made up of a number of elements that describe the settings of specific Oracle Application Server components and the dependencies that OracleAS Portal instances have on them. The Portal Dependency Settings file definition is modeled in the `iasconfig.xsd` schema file that is located in the `$ORACLE_HOME/portal/conf` directory.

For the complete list of element descriptions used in the Portal Dependency Settings file, refer to the *OracleAS Portal Configuration Guide, Appendix A "Using the Portal Dependency Settings Tool and File."*

Portal Dependency Settings File (continued)

Example of the iasconfig.xml file:

```
<IASConfig XSDVersion="1.0">
  <IASInstance
    Name="portal.edrsr25p1.us.oracle.com"
    Host="edrsr25p1.us.oracle.com"
    <OIDComponent
      AdminPassword="@BWhcu41NOcaQUV4FzuW3+IE6Faf4hPsRMg=="
      AdminDN="cn=orcladmin"
      SSLEnabled="false"
      LDAPPort="3060"/>
    <WebCacheComponent
      AdminPort="4000"
      ListenPort="7778"
      InvalidationPort="4001"
      InvalidationUsername="invalidator"
      InvalidationPassword="@BVs6wUm3xqs/SMXYov29hXlCA=="
      SSLEnabled="false"/>
    <EMComponent
      ConsoleHTTPPort="1811"
      SSLEnabled="false"/>
  </IASInstance>
  <PortalInstance
    DADLocation="/pls/portal"
    SchemaUsername="portal"
    SchemaPassword="@BWs7Sze2lNTRJgiMW2l14Gkq42HgynbMWA=="
    ConnectString="cn=infra,cn=oraclecontext">
    <WebCacheDependency
      ContainerType="IASInstance"
      Name="portal.edrsr25p1.us.oracle.com"/>
    <OIDDependency
      ContainerType="IASInstance"
      Name="ias-1.edrsr25p1"/>
    <EMDependency
      ContainerType="IASInstance"
      Name="portal.edrsr25p1.us.oracle.com"/>
  </PortalInstance>
</IASConfig>
```

Portal Dependency Settings Tool

Run the `ptlconfig` script to:

- Update OracleAS Metadata Repository for a specific Portal instance defined in the Portal Dependency Settings file
- Encrypt all plain text passwords in the Portal Dependency Settings file
- Update OracleAS Web Cache, Oracle Internet Directory, Oracle Enterprise Manager, and OracleAS Portal data as defined in the Portal Dependency Settings file
- Create or delete the provisioning profiles in Oracle Internet Directory of an OracleAS Portal instance

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Portal Dependency Settings Tool

To update OracleAS Metadata Repository with configuration settings in the `iasconfig.xml` file, you must use the `ptlconfig` script. This script can:

- Update OracleAS Metadata Repository for a specific Portal instance defined in the Portal Dependency Settings file
- Encrypt all plain text passwords in the Portal Dependency Settings file
- Update OracleAS Web Cache, Oracle Internet Directory, Oracle Enterprise Manager, and OracleAS Portal site data, as defined in the Portal Dependency Settings file
- Create or delete provisioning profiles in Oracle Internet Directory of an OracleAS Portal instance

ptlconfig Modes

You can run `ptlconfig` in:

- **Configuration mode**

```
ptlconfig -dad portal -sso -host  
edrsr25p1.us.oracle.com -port 7778 -ssl
```

- **Encryption mode**

```
ptlconfig -encrypt
```

- **Load mode**

```
ptlconfig -load -schema portal30 -pw welcome1 -conn  
edrsr25p1.us.oracle.com:1521:infra -lp 4889
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

ptlconfig Modes

You can run `ptlconfig` in:

- Configuration mode to update a specific OracleAS Portal instance from the Portal Dependency Settings file:

```
ptlconfig -dad portal -sso -host edrsr25p1.us.oracle.com -  
port 7778 -ssl
```
- Encryption mode to encrypt plain text passwords in the Portal Dependency Settings file:

```
ptlconfig -encrypt
```
- Load mode to create and update entries in `iasconfig.xml` with the configuration settings of a specific Portal schema:

```
ptlconfig -load -schema portal30 -pw welcome1 -conn  
edrsr25p1.us.oracle.com:1521:infra -lp 4889
```

When you run this script, the `ptlconfig.log` log file is created in the `$ORACLE_HOME/portal/logs` directory, which records operations performed on OracleAS Metadata Repository.

ptlconfig Modes (continued)

Parameters of the ptlconfig script:

Parameter	Description	Example
-all	Updates all OracleAS Portal instances from the Portal Dependency Settings file	ptlconfig -all
-dad	Is the portal DAD name; used to update a specific OracleAS Portal instance from the Portal Dependency Settings file	ptlconfig -dad portal
-encrypt	Encrypts any plain text passwords in the Portal Dependency Settings file	ptlconfig -encrypt
-wc	Updates OracleAS Web Cache data as defined in the Portal Dependency Settings file	ptlconfig -dad portal -wc
-oid	Updates Oracle Internet Directory data as defined in the Portal Dependency Settings file	ptlconfig -all -oid
-site	Updates OracleAS Portal data (listening host and port) as defined in the Portal Dependency Settings file	ptlconfig -dad portal -site
-em	Updates Oracle Enterprise Manager data as defined in the Portal Dependency Settings file	ptlconfig -all -em

For additional information about ptlconfig parameters, refer to the *Oracle Application Server Portal Configuration Guide 10g Release 2 (10.1.2), Appendix A, Using the Portal Dependency Settings Tool and File*.

Configuring Portal Web Cache Settings

OracleAS Metadata Repository Used By Portal	
Status	Up
Name	orcl
Start Time	Oct 7, 2005 2:10:35 AM
Database Version	10.1.0.4.2
Repository Version	10.1.4.0.0

Administration	
Portal Web Cache Settings	
<div> <div>Cancel</div> <div>Revert</div> <div>Apply</div> </div>	
Specify the Oracle Web Cache settings that Portal should use.	
Published Host	EDRSR25P1
Listening Port	7778
Listening Port SSL Enabled	No <input type="checkbox"/>
Invalidation Host	EDRSR25P1
Invalidation Port	9401
Invalidation Username	invalidator <input type="checkbox"/>
Invalidation User Password	*****
Confirm Password	*****

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Portal Web Cache Settings

You can define the OracleAS Web Cache settings that OracleAS Portal should use from Application Server Control. When you set OracleAS Web Cache properties from Application Server Control, the Portal Dependency Settings file (`iasconfig.xml`) located on this middle tier is updated automatically, and the OracleAS Portal schema is also updated. To configure Portal Web Cache settings, perform the following steps:

1. Navigate to your Application Server Control Portal Home page.
2. Click the Portal Web Cache Settings link under Administration.
3. Modify the Web Cache information on the Portal Web Cache Settings page.

When you change the OracleAS Web Cache properties here, OracleAS Portal's perspective of these properties changes. However, the actual OracleAS Web Cache configuration properties do not change. When you change OracleAS Web Cache properties on the Portal Web Cache Settings page, the properties are saved to `iasconfig.xml`, but not to the `webcache.xml` file. You must navigate back to the Web Cache Administration page in Application Server Control Console to make the appropriate changes.

Configuring Virtual Hosts

To configure virtual hosts, perform the following steps:

- 1. Create virtual hosts.**
- 2. Configure OracleAS Web Cache.**
- 3. Register OracleAS Portal with OracleAS Single Sign-On.**
- 4. Verify the configuration.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Virtual Hosts

To configure virtual hosts with OracleAS Portal, you must perform the following steps:

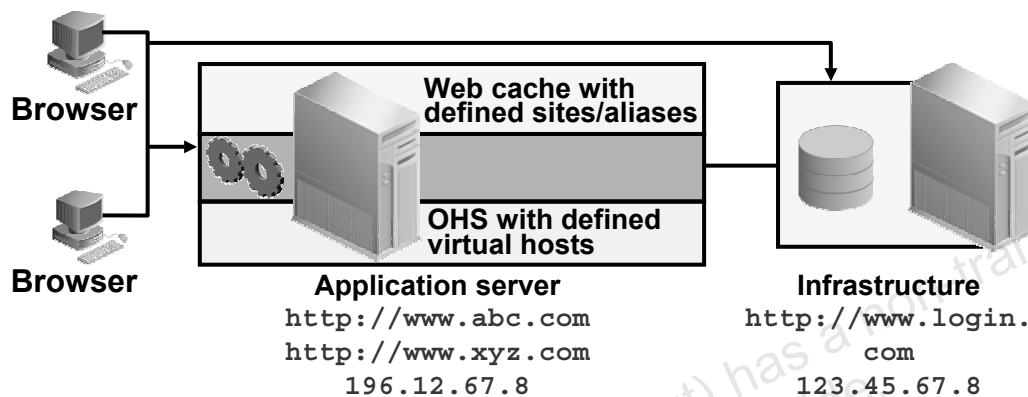
1. Create virtual hosts.
2. Configure OracleAS Web Cache.
3. Register OracleAS Portal with OracleAS Single Sign-On.
4. Verify the configuration.

For example, your server name is `www.server.com`, and you connect to OracleAS Portal at `http://www.server.com:7778/pls/portal`. In this example, port 7778 is the Oracle HTTP Server listening port and the OracleAS Web Cache listening port is 7779. You want to access OracleAS Portal using the actual server name, as well as using a virtual host name, `http://www.virtualhost.com`, where both URLs resolve to the same IP address. OracleAS Single Sign-On is installed on a different machine and accessed at `http://www.ssoserver.com:7777/pls/orasso`.

1. Create Virtual Hosts

Configure virtual hosts by using Oracle Enterprise Manager 10g Application Server Control. For additional details, refer to the lesson titled “Configuring Directives and Virtual Hosts.”

Configuring Virtual Hosts



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Virtual Hosts (continued)

2. Configure OracleAS Web Cache

The new virtual host that you configured in the previous step must be defined in OracleAS Web Cache. Additionally, you must create a site alias in OracleAS Web Cache to make the multiple virtual hosts apparent to OracleAS Metadata Repository. For additional details, refer to the lesson titled “Configuring and Managing OracleAS Web Cache.”

3. Register OracleAS Portal with OracleAS Single Sign-On

For OracleAS Portal in Oracle Application Server Single Sign-On to work properly, it must be referenced by any partner application with the same host name in the URL because cookies are sent back only to the host that generated them. So, in this example, OracleAS Single Sign-On must always be referenced as `http://www.ssoserver.com`. You must register `www.server.com` and `www.ssoserver.com` as partner applications.

Configuring Virtual Hosts (continued)

4. Verify the Configuration

To verify that the virtual hosts are set up correctly, connect to OracleAS Portal using either of the following URLs:

`http://server.com:7778/pls/portal`

`http://virtualhost.com:7778/pls/portal`

You should get a login screen at `http://www.server.com` on the first login and must be able to log in successfully. A subsequent login from the other virtual host should result in a successful single sign-on without a prompt for login credentials.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Configuring Multiple Middle Tiers

- You can set up OracleAS Portal in a multiple middle-tier environment, with a load-balancing router to access the same Oracle Application Server Metadata Repository.
- This solution is best:
 - For internal deployment
 - For environments where scalability and high availability are important

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Multiple Middle Tiers

You can set up OracleAS Portal in a multiple middle-tier environment, with a load-balancing router to access the same Oracle Application Server Metadata Repository. This solution is best for internal deployments because, as with all default Portal installations, it is not configured to use SSL. The load-balancing router provides a single published address to the client tier, and the client hosts a farm of servers that actually service the requests, based on the distribution of the requests done by the load-balancing router. This solution is also best for environments where scalability and high availability are important. For additional information about configuring multiple middle tiers with a load balancing router, refer to the *Oracle Application Server Portal Configuration Guide 10g Release 2 (10.1.2)*.

Configuring Multiple Middle Tiers

To configure OracleAS Portal in a multiple middle-tier environment, front-ended by a load-balancing router:

1. Install a single Portal and Wireless middle tier (M1)
2. Configure OracleAS Portal on M1 to be accessed through the load-balancing router
3. Confirm that OracleAS Portal is up and running
4. Install a new Portal and Wireless middle tier (M2)
5. Configure the new middle tier (M2) to run your existing portal
6. Configure Portal Tools and Web Providers (optional)
7. Enable session-binding on OracleAS Web Cache
8. Confirm the completed configuration

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring a Dedicated OracleAS Web Cache

To configure a dedicated OracleAS Web Cache installed on a different machine to communicate with OracleAS Portal middle tier:

- 1. Verify that the OracleAS Web Cache on the dedicated server is running**
- 2. Configure OracleAS Web Cache on the dedicated server**
- 3. Stop the unused OracleAS Web Cache on the middle-tier server (optional)**
- 4. Configure OracleAS Portal middle tier with OracleAS Web Cache settings**
- 5. Configure virtual host settings for Oracle HTTP Server**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring a Dedicated OracleAS Web Cache

When you install OracleAS Portal middle tier, OracleAS Web Cache is automatically configured on the same machine as the one hosting the middle tier. To configure a dedicated OracleAS Web Cache installed on a different computer to communicate with OracleAS Portal middle tier, you must disable the OracleAS Web Cache installed on the OracleAS Portal middle tier. To configure a dedicated OracleAS Web Cache, perform the following steps:

1. Verify that OracleAS Web Cache on the dedicated server is running.
2. Configure OracleAS Web Cache on the dedicated server. To properly configure OracleAS Web Cache, installed on the dedicated server, you need the origin server information from OracleAS Web Cache installed on the same computer as OracleAS Portal middle tier.
3. Using a copy of OracleAS Web Cache instance that was installed on the middle-tier computer, modify the Application Web Servers properties for the dedicated OracleAS Web Cache.
4. As an optional task, you can stop the unused OracleAS Web Cache on the middle-tier server.

Configuring a Dedicated OracleAS Web Cache (continued)

5. Configure OracleAS Portal middle tier with OracleAS Web Cache settings.
OracleAS Portal middle tier needs to know the OracleAS Web Cache listen ports, the invalidator username, invalidator password settings, and other information. Apply the new host name and port number of the dedicated OracleAS Web Cache to OracleAS Portal middle tier by modifying these settings on the Portal Web Cache Settings page.
6. Configure virtual host settings for Oracle HTTP Server.
Create virtual host entries in the `httpd.conf` file of Oracle HTTP Server that is part of the OracleAS Portal middle tier, with the dedicated OracleAS Web Cache settings.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Configuring OracleAS Portal Search Portlets

The screenshot shows the 'Global Settings' window with the 'Search' tab selected. The window has a title bar with tabs: Main, Configuration, SSO/OID, Cache, Mobile, Proxy, and Search. Below the title bar is a 'Global Settings' header with 'Apply', 'OK', and 'Cancel' buttons. The main content area is divided into two sections: 'Basic Search Portlets and Basic Search Box Items' and 'Internet Search Engine'.

Basic Search Portlets and Basic Search Box Items
 Select a page to display results from Basic Search portlets and items.
 Page Name: Basic Search Results

Advanced, Custom and Saved Search Portlets
 Select a page to display results from Search portlets. You can override the default page if desired.
 Page Name: Search Results Page

Search Properties
 Enter the number of search results to display on a page. If the number of results returned by a search exceeds this number, the search results pages include Next and Previous links that enable users to view all the results.
 Hits Per Page: 20

Internet Search Engine
 Enter the URL of the search engine that you want users to use to search the Internet if required. Enter the text that users click to access the specified search engine.
 URL:
 Link:
 Text: For searching the Internet, use

ORACLE
 Copyright © 2005, Oracle. All rights reserved.

Configuring OracleAS Portal Search Options

After a standard OracleAS Portal installation, you can start using the search feature in OracleAS Portal. Without any additional configuration, you can place the built-in OracleAS Portal search portlets on a page and use it to search the portal context. You can also configure certain aspects of the search feature that affect all OracleAS Portal search portlets. To configure OracleAS Portal search portlets, perform the following steps:

1. In the Services portlet, click Global Settings. Click the Search tab.
2. In the Search Results Pages section, for Basic Search Portlets and Basic Search Box Items, choose a search results page.
3. Similarly, for Advanced, Custom and Saved Search Portlets, choose a search results page. You can choose any portal page that contains a search portlet.
4. You can limit the number of search results that are displayed on all search result pages. This limit is applied to results from Basic, Advanced, and Custom Search portlets. Enter the number of search results to be displayed on a page in the Hits Per Page field.

Configuring OracleAS Portal Search Options (continued)

5. If users do not find the information they need when they search OracleAS Portal, they can extend their search by using an Internet search engine. When you set the URL of an Internet search engine and the link text that users click to access the specified Internet search engine, it applies to all new and existing Advanced/Custom Search portlet instances that display an Internet search link. In the Internet Search Engine section, for URL, enter the URL of an Internet search engine, such as <http://www.yahoo.com>. For Link Text, enter the text that is to be displayed as a link.
6. Click OK.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Administering Web Clipping

Provider Test Page: Web Clipping		
Portlet Information		
Your provider contains the following portlets:		
WebClippingPortlet		
Provider Initialization Parameters		
The following parameters are defined in the Web application configuration file (web.xml):		
<u>Name</u>	<u>Value</u>	
invalidation_caching	true	
Provider Configuration		
You can configure each of the following settings. For more information, see the "Configuring Web Clipping" section in the Configuring Portal Tools Providers appendix of the Oracle Application Server Portal Configuration Guide. Learn More...		
<u>Setting</u>	<u>Status</u>	<u>Actions</u>
Web Clipping Repository	Configured	Edit
HTTP Proxy	Not Configured	Edit
Portlet Caching	Use Portal Cache (Validation)	Edit

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Administering Web Clipping

Web Clippings are sections of the existing Web content, such as HTML code, applets, or JavaScript on a Web page, which can be reused in other Web pages. You can capture and display clipped Web content as portlet content through the Web Clipping portlet.

Before using Web Clipping, you must complete the following administrative tasks:

- Configure the Web Clipping Repository (required configuration)
- Configure the HTTP or HTTPS proxy settings (optional)
- Configure Caching (optional)

Web Clippings have definitions that are stored persistently in the Web Clipping Repository hosted by an Oracle database server. You can configure the Web Clipping Repository by using the Provider Test Page: Web Clipping page at:

<http://<host name>.<domain>:port/portalTools/webClipping/providers/webClipping>

The Provider Test Page: Web Clipping page automatically detects whether the Web Clipping provider is configured to access the database. If it is not, the Status column for the Web Clipping Repository displays Not Configured.

Administering Web Clipping (continued)

To configure repository settings, perform the following steps:

1. In the Provider Configuration section, the Setting column contains the Web Clipping Repository field. Click the corresponding Edit link in the Actions column.
2. In the Repository Settings section of the Edit Provider: webClipping page, you specify the database connection information for the Web Clipping provider. Select one of the following choices for the Repository Target database:
 - OracleAS Infrastructure Database (default): If you select this option, you do not need to specify any other connection parameters.
 - Other Oracle9i (or later) Database: If you select this option, specify the following connection parameters:
 - Server Host: The name of the database server
 - Listener Port: The listener port for the database server
 - SID: The database SID
 - Username: The database username
 - Password: The password for the database user
 - If you require a secure database connection, from the Advanced Security Option list, select enabled (secure database connection).
3. Click OK to save the settings and return to the Provider Test page.

Administering Web Clipping

Proxy Settings

Specify information about the proxy host name and port number. If HTTP Proxy Port is not specified, it will be set to default port number 80. You can also specify a list of host domains for which the proxy will be bypassed.

HTTP Proxy Port

No Proxy for

Requires Authentication ☐

Type

Realm

Login Scheme

Username

Password

Time out settings

These values are used as default values for the Web Clipping Portlet

Timeout (in seconds)

Caching Parameters

The Web Clipping Provider uses validation or invalidation cache. You may select one or the other. A value may also be set as the default expiration value for both caching scheme.

Cache Expires

Caching Scheme

Administering Web Clipping (continued)

The following are optional configurations for Web Clipping Provider. On the Web Clipping Provider Test page, click the links in the Action column next to a setting to configure it.

HTTP Proxy Server

1. Click Edit next to the status of the HTTP Proxy on the Web Clipping Provider Test Page.
2. The Edit Provider page is displayed, on which you can specify the HTTP proxy host and the HTTP proxy port.
3. For access to servers that are inside the firewall, you can specify a list of domain names that do not require a proxy in the No Proxy for field.
4. If your access to external Web sites is through a proxy server that requires authentication, select the Requires Authentication check box.
5. For proxies that require authentication, you must specify the method of authentication that is employed by the proxy server: Basic or Digest.
6. With the help of your proxy administrator, you should be able to specify the realm for the proxy server access.

Administering Web Clipping (continued)

7. Select one of the following login schemes:
 - **Use login below for all users:** Use this option if the login information you enter below is to be used for all users.
 - **Require login for all users:** Use this option if each user connects to the proxy server using his or her own login credentials.
 - **Require login for all users:** Use login below for public users: Use this option for specifying default username and password for OracleAS Portal public users.
8. Depending on the option selected above, specify the username and password to log in to the proxy server.

You do not need to restart OC4J for the new settings to take effect.

Web Cache

If you want your portlet content cached by using invalidation-based caching, an Oracle Web Cache instance must be configured in front of your provider.

After you set up and start the Web Cache instance, register the Web Clipping Provider as usual, but set the URL host name and port number to point to the Web Cache instance instead. For example,

`http://<cache_instance_name>:<cache_port>/portalTools/webClipping/providers/
webClipping`

When a Web Clipping portlet definition is altered through either an Edit Defaults or Customize page, the provider generates a request that invalidates and removes the portlet content from the cache. The invalidation request is sent to the invalidation port of the Web Cache instance.

Administering OmniPortlet

Provider Test Page: omniPortlet

Portlet Information
Your provider contains the following portlets:

- OmniPortlet
- Simple Parameter Form


Provider Initialization Parameters
The following parameters are defined in the Web application configuration file (web.xml):

Name	Value
invalidation_caching	true

Provider Configuration
You can configure each of the following settings. For more information, see the "Configuring OmniPortlet" section in the Configuring Portal Tools Providers appendix of the Oracle Application Server Portal Configuration Guide. [Learn more...](#)

Setting	Status	Actions
HTTP Proxy	Not configured	Edit
* Web Cache Invalidation	Configured	
* BI Graph Bean	Installed	
* DISPLAY Environment Variable	Configured	
*Requires restarting OC4J		

Repository Setting

Setting	Status	Actions
 Secured Data Repository	Configured	Edit

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Administering OmniPortlet

A page designer can modify the default settings of OmniPortlet Web Provider portlets to:

- Display data from multiple sources (SQL, XML, Web Services, Web page, and so on)
- Access secured data
- Format data by using a variety of layouts (chart, form, reports, and so on)
- Accept and pass page parameters

You can test whether the OmniPortlet Web Provider is functioning properly from the Provider Test Page: omniPortlet. You can access the test page by clicking the OmniPortlet Provider link on the Portal Tools Welcome Page located at:

`http://<host name>.<domain>:<Oracle HTTP Server port of Portal instance>/portalTools`
or, by directly accessing the following URL:

`http://<host name>.<domain>:portalTools/omniPortlet/providers/omniPortlet`

The test page displays:

- The portlets included in OmniPortlet Provider
- The provider initialization parameters and values

User Interface XML (UIX) is required for OmniPortlet Provider to function properly. UIX is installed when you install Portal Tools and, therefore, no additional configuration is necessary.

Oracle Application Server 10g R2: Administration I 14-32

Administering OmniPortlet (continued)

The following are the additional optional configurations for the OmniPortlet Provider:

- **HTTP Proxy:** Set up the HTTP Proxy if a proxy server is required for the provider to make a URL connection to a server outside the firewall.
- **Web Cache:** If you want your portlet content cached by using invalidation-based caching, then an Oracle Web Cache instance must be configured in front of your provider.
- **BIGraph Bean:** BIGraph Bean must be installed properly so that the portlet can produce graphical output such as charts.
- **DISPLAY Environment Variable (Linux/UNIX):** You must have a valid DISPLAY setting.
- **Secured Data Repository Setting:** OmniPortlet leverages Web Clipping Repository to store credentials needed to access secured data. This is a required configuration if you intend to use the Web Page data type or work with secured data (such as SQL database server or data source URL with HTTP Basic Authentication).

Note: Click the Learn more link for additional information about Provider Configuration Setting.

Summary

In this lesson, you should have learned how to:

- Describe OracleAS Portal configuration tasks
- Configure the Self-Registration feature to enable users to create their own portal accounts
- Configure OracleAS Portal for WebDAV
- List the configuration modes
- Configure language support
- Configure the OracleAS Portal instance dependencies by using the Portal Dependency Settings file

ORACLE

Copyright © 2005, Oracle. All rights reserved.

15

Administering the OracleAS Single Sign-On Server

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

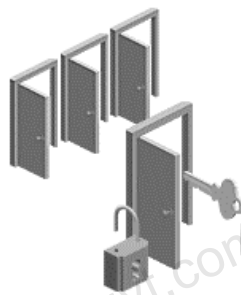
- **Discuss the components of the OracleAS Single Sign-On server**
- **Explain the OracleAS Single Sign-On server authentication flow**
- **Manage and configure the OracleAS Single Sign-On server**
- **Administer the partner and external applications**
- **Monitor the OracleAS Single Sign-On server**
- **Access the OracleAS Single Sign-On server from OracleAS Portal**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Single Sign-On Server: Overview

The OracleAS Single Sign-On server is a component of Oracle Application Server, which enables users to log in to the various components of Oracle Application Server by using a single username and password.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Single Sign-On Server: Overview

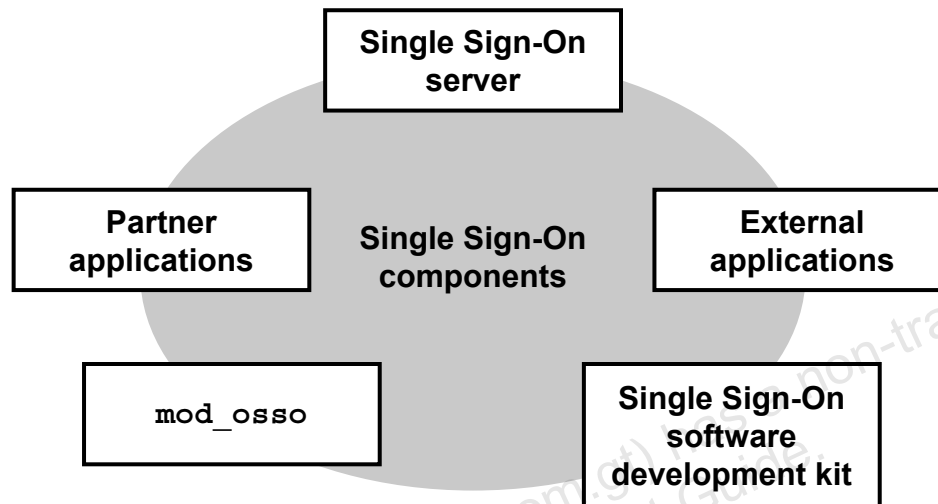
In the Internet environment, you must remember a username and password to authenticate and access any application. As the number of Web applications increases, the number of usernames and passwords also increases. It becomes very difficult to manage multiple usernames and passwords. The OracleAS Single Sign-On server solves this problem for you.

The Single Sign-On server enables you to gain access to multiple applications for which you have registered using a single username and password. After you are authenticated by the Single Sign-On server, you can access all the applications you had registered, without reentering the username and password.

The OracleAS Single Sign-On server provides the following benefits:

- Reduces administrative and management costs because you do not have to manage multiple user accounts
- Enables easier login for end users because they have to remember only one username and password
- Improves security because the users have to use only one username and password, therefore, they are less likely to use simple, easily exposed passwords or to write these passwords down

Single Sign-On Components



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Single Sign-On Components

OracleAS Single Sign-On is a component of OracleAS Identity Management Infrastructure. OracleAS Single Sign-On and other components enable OracleAS Identity Management Infrastructure to manage the security life cycle of users and network entities in an efficient and cost-effective way. The various components of the OracleAS Single Sign-On server are:

- **Single Sign-On server:** The Single Sign-On server is a set of programs stored in the metadata repository running on Oracle HTTP Server and OC4J server. The Single Sign-On server is accessed from the metadata repository using the `orasso` database access descriptor (DAD) created in OracleAS Infrastructure. The server enables the users to securely log in to single sign-on applications, such as e-mails, online file storages, and directories. These applications can be of two types:
 - Partner application
 - External application
- **Partner applications:** Users accessing an application on Oracle Application Server have to be authenticated using the OracleAS Single Sign-On server. The applications delegate the authentication function to the OracleAS Single Sign-On server. The OracleAS Single Sign-On server authenticates the user on behalf of the application. For this reason, these applications are known as partner applications. After the OracleAS Single Sign-On server authenticates the user, the partner application is responsible for determining whether a user is authenticated and has the privileges for using the application or not.

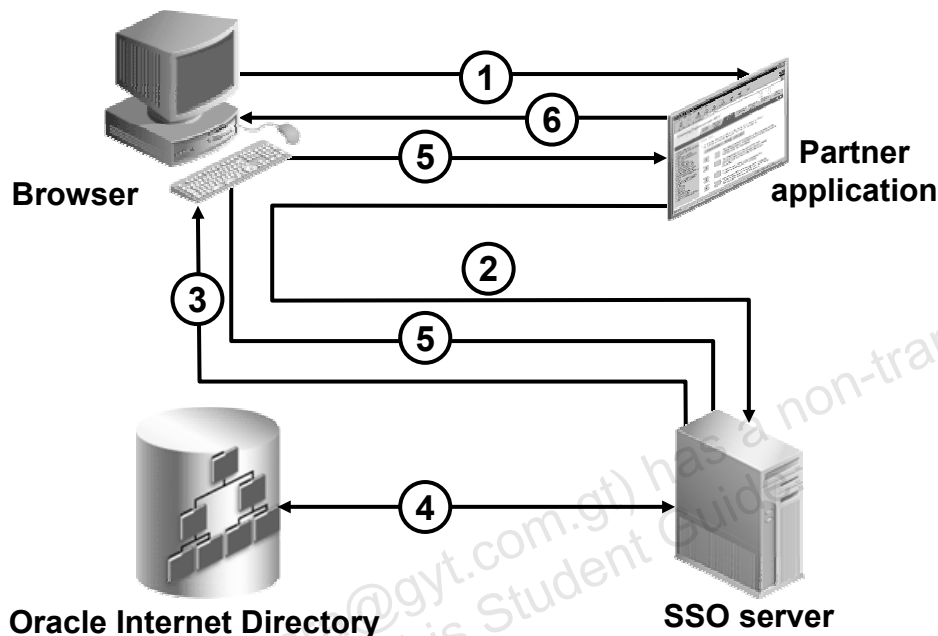
Single Sign-On Components (continued)

- **External Application:** The applications are not deployed on Oracle Application Server. These applications do not delegate authentication function to the OracleAS Single Sign-On server. These applications display their own HTML login forms to the user to enter the username and password. You can configure and store the external application username and password in the OracleAS Single Sign-On server. This enables the users to log in to the external application from the OracleAS Single Sign-On server, without authenticating to the external application separately. The external application username and password are stored in the metadata repository in an encrypted form.
- **mod_osso:** Unlike the previous versions of Oracle Application Server, the Single Sign-On administrator need not integrate the server with the partner applications using the Single Sign-On SDK to enable single sign-on functionality. This is taken care by mod_osso, which is an Oracle HTTP Server module that provides authentication to the Oracle Application Server applications. mod_osso acts as a sole partner application to the OracleAS Single Sign-On server and provides transparent authentication for the partner application. After authenticating the user, mod_osso transmits the simple header values to the Oracle Application Server applications that need to validate the user. The header values include:

- user name
- user DN
- user GUID
- language and territory

All the single sign-on usernames and passwords are stored in Oracle Internet Directory. The OracleAS Single Sign-On server authenticates the user against the user's entry in the Oracle Internet Directory server.

Authentication Flow for OracleAS Single Sign-On



Copyright © 2005, Oracle. All rights reserved.

Authentication Flow for OracleAS Single Sign-On

The `mod_osso` module in the HTTP listener enables the HTTP listener to be a partner application for the OracleAS Single Sign-On server. It effectively provides another application with the integration option because applications that can obtain user identities from `mod_osso` can also be OracleAS Single Sign-On enabled. What distinguishes SSO is that users authenticate their identities only once, and then are automatically granted access to a variety of permitted resources with no further identification required. The illustration in the slide displays the authentication flow for the OracleAS Single Sign-On server. The following steps are involved in the authentication process:

1. A user accesses a partner application. This application determines that the user is not authenticated because there is no partner application cookie in the user's browser.
2. The partner application redirects the user to the Single Sign-On server.
3. The Single Sign-On server displays a username and password page that prompts the user to supply this information. The Single Sign-On server verifies the password and sets an SSO cookie in the user's browser for authentication to the SSO server.
4. OracleAS Single Sign-On credentials (username and password) are verified in Oracle Internet Directory.
5. The Single Sign-On server redirects the user to the partner application with an encrypted token to authenticate the user to the application.

Oracle Application Server 10g R2: Administration I 15-6

Authentication Flow for OracleAS Single Sign-On (continued)

6. The partner application sets an application cookie in the browser of the requesting user. The Web server also creates a cookie for the user in the browser.

If the user requests a second time, the Web server uses the `mod_ossso` cookie to validate the user. During the session, if the user requests for any partner application, the login page is not displayed and the user is given access to the partner application.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Starting and Stopping OracleAS Single Sign-On Components

You can start and stop OracleAS Single Sign-On using:

- **Application Server Control Console**
- **Command-line tool: `opmnctl`**

System Components					
<div>Start Stop Restart Delete OC4J Instance</div> <div>Enable/Disable Components Create OC4J Instance</div>					
Select All Select None					
Select	Name	Status	Start Time	CPU Usage (%)	Memory Usage (MB)
<input type="checkbox"/>	HTTP Server	↑	Sep 6, 2005 6:02:59 AM	0.49	110.29
<input type="checkbox"/>	Internet Directory	↑	Sep 6, 2005 5:51:35 AM	0.03	8.81
<input type="checkbox"/>	OC4J SECURITY	↑	Sep 6, 2005 6:03:04 AM	0.00	20.14
<input type="checkbox"/>	oca	↑	Sep 6, 2005 6:04:02 AM	0.00	18.07
<input type="checkbox"/>	Single Sign-On:orasso	↑	N/A	N/A	N/A
<input type="checkbox"/>	Management	↑	Sep 6, 2005 5:44:47 AM	0.55	107.31

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting and Stopping OracleAS Single Sign-On Components

The OracleAS Single Sign-On server is an application that runs on Oracle HTTP Server; starting Oracle HTTP Server starts the OracleAS Single Sign-On server too. The SSO server also runs on OC4J instance. Starting, stopping, and restarting the OC4J instance also results in starting, stopping, and restarting the SSO server. You can use the Application Server Control Console to start, stop, and restart Oracle HTTP Server. To manage Oracle HTTP Server, perform the following steps:

1. Point your browser to `http://<host name>.<domain>:<Application Server Control port>` to log in to Application Server Control.
2. Click the name of the instance under Standalone Instances. A page that displays the details of that instance appears.
3. Select HTTP Server and OC4J SECURITY under System Components and perform the start, stop, or restart operation. This starts, stops, and restarts the Single Sign-On component too.

Starting and Stopping OracleAS Single Sign-On Components (continued)

You can also use the command-line tool to start, stop, and restart Oracle HTTP Server:

- **Starting Oracle HTTP Server:** `$ORACLE_HOME/opmn/bin/opmnctl startproc process-type=HTTP_Server`
- **Stopping Oracle HTTP Server:** `$ORACLE_HOME/opmn/bin/opmnctl stopproc process-type=HTTP_Server`
- **Restarting Oracle HTTP Server:** `$ORACLE_HOME/opmn/bin/opmnctl restartproc process-type=HTTP_Server`

To start, stop, and restart the OC4J_SECURITY instance, use `process-type=OC4J_SECURITY` with the `opmnctl` command.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

OracleAS Single Sign-On Administrator's Role

- **As a Single Sign-On administrator, you have full privileges for the OracleAS Single Sign-On server.**
- **You can perform the following tasks as a Single Sign-On administrator:**
 - **Configure OracleAS Single Sign-On server settings**
 - **Administer partner applications**
 - **Administer external applications**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Single Sign-On Administrator's Role

After installing OracleAS Infrastructure, there is only one Single Sign-On administrator (that is, `orcladmin`). You specify the password for `orcladmin` at the time of installation. The `orcladmin` user is used to create the OracleAS Single Sign-On administrators.

You can create users using OracleAS Self-Service Console or Oracle Directory Manager. You can create the OracleAS Single Sign-On administrator by making a user a member of the `iASAdmins` group using Oracle Directory Manager.

To add a user to the `iASAdmins` group, perform the following steps:

1. Start Oracle Directory Manager and log in as `orcladmin` (Oracle Internet Directory super user).
2. Using the navigation pane, navigate to `Entry Management > cn=OracleContext > cn=Groups > cn=iASAdmins`. All the properties of the group are displayed in the right pane.
3. Add the user entry DN to the `uniquemember` field, and then click **Apply**.

This adds the specific user to the `iASAdmins` group, and then the user can administer the OracleAS Single Sign-On server.

OracleAS Single Sign-On Administrator's Role (continued)

As an OracleAS Single Sign-On administrator, you can access the OracleAS Single Sign-On administrative pages. You can perform the following tasks as a Single Sign-On administrator:

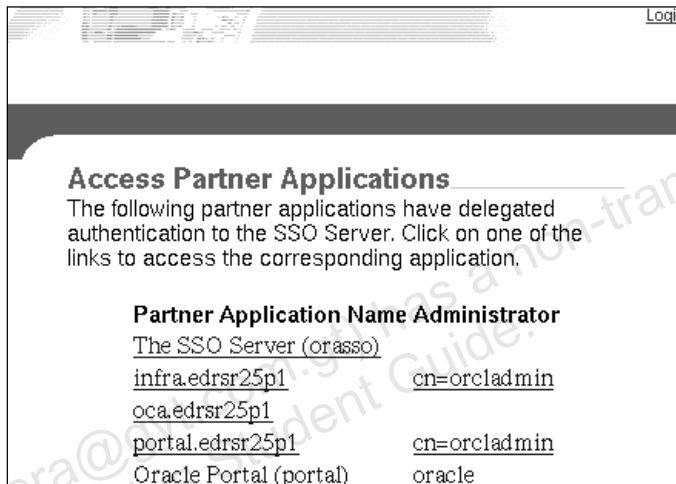
- Configure the Single Sign-On server
- Administer partner application
- Administer external application

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

OracleAS Single Sign-On Administration Pages

You can access the OracleAS Single Sign-On administrative pages at:

http://<host name>.<domain>:<Oracle HTTP Server port>/pls/orasso



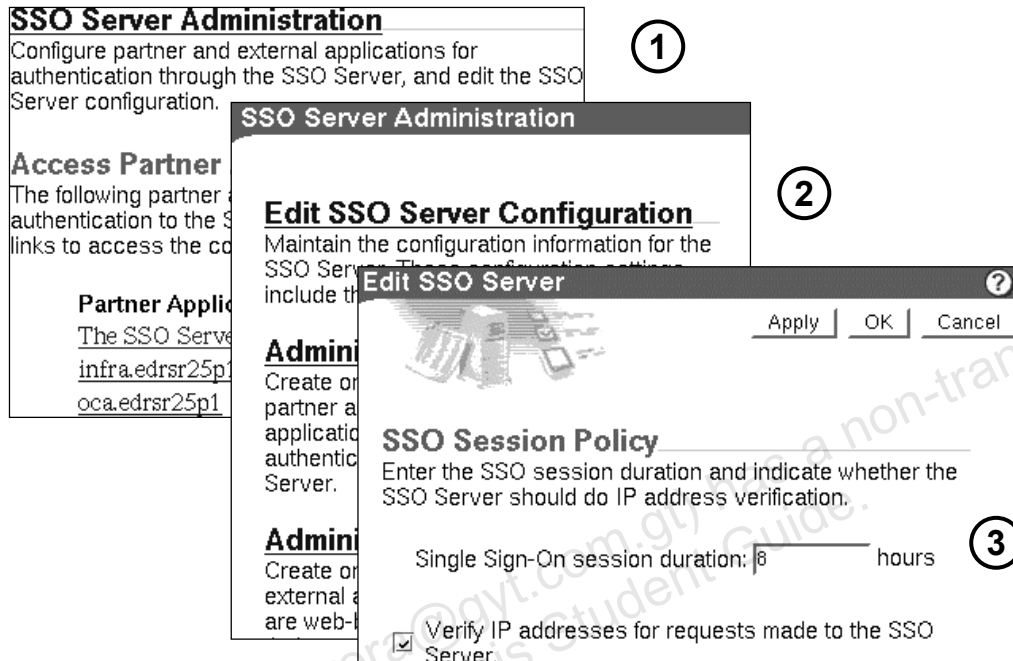
Copyright © 2005, Oracle. All rights reserved.

OracleAS Single Sign-On Administration Pages

To access the OracleAS Single Sign-On administration pages, perform the following steps:

1. Point your browser to ***http://<host name>.<domain>:<Oracle HTTP Server port>/pls/orasso***.
2. Click the Login link to log in as an administrator. The OracleAS Single Sign-On server login page appears.
3. Enter the username and password of the user with single sign-on administrative privileges. Then click Login.
4. The OracleAS Single Sign-On administration page appears. Click the SSO Server Administration link to administer the Single Sign-On server.

Configuring the OracleAS Single Sign-On Server



Configuring the OracleAS Single Sign-On Server

After you log in to the OracleAS Single Sign-On server, the image labeled 1 in the slide is displayed. Select SSO Server Administration to administer various components of the OracleAS Single Sign-On server. The image labeled 2 in the slide is displayed.

Click any of the links to configure and administer that component. Click the Edit SSO Server Configuration link to configure the OracleAS Single Sign-On server. The image labeled 3 is displayed.

On this page, you can configure the following:

- **Single Sign-On session duration:** Specify the number of hours a user can be logged in to the Single Sign-On server without having to time out and log in again.
- **Verify IP addresses for requests made to the SSO server:** Select to verify that the IP address of the browser is the same as the IP address in the authentication request to the Single Sign-On server.

Partner Application: Overview

- The applications on Oracle Application Server that delegate their authentication functionality to the OracleAS Single Sign-On server are known as partner applications.
- When you log in to any of the partner application through OracleAS Single Sign-On, you can access all the partner applications registered with OracleAS Single Sign-On.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Administering Partner Applications

Various applications on Oracle Application Server are registered to the OracleAS Single Sign-On server as partner applications at the time of installation. The Single Sign-On partner applications, if integrated with `mod_osso`, are registered either by the `ssoreg.sh` script or by the `ssoreg.bat` batch file, depending on the Linux, UNIX, or Windows platform. OracleAS Portal is registered by the `ptlconfig` script.

Registering mod_osso

You can use the `ssoreg.sh` script to register `mod_osso` in cases when the application is not registered.

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

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Registering mod_sso

Under certain circumstances, you must reregister `mod_osso` manually by using the single sign-on registration tool. These circumstances are as follows:

- The host name and port number of Oracle HTTP Server are changed after Oracle Application Server is installed.
- The `osso.conf` file is deleted or corrupted.
- SSL is enabled on the Single Sign-On server after Oracle Application Server is installed.

The parameters to the commands are described as follows:

- **oracle_home_path:** Absolute path to the Oracle home
- **site_name:** Name of the site, typically the host name and port (for example, the contiguous string `host:port`)
- **config_mod_osso:** If set to `TRUE`, this parameter indicates that the application being registered is `mod_osso`. You must include `config_mod_osso` for the `osso.conf` file to be generated.
- **mod_osso_url:** URL is `http://oracle_http_host.domain:port`

Registering mod_sso (continued)

- **virtualhost:** Optional. Use this parameter only if registering an Oracle HTTP virtual host with the Single Sign-On server.
- **update_mode:** Optional; creates, deletes, or modifies the partner registration record. CREATE, the default, generates a new record. DELETE removes the existing record. MODIFY deletes the existing record and then creates a new one.
- **config_file:** Optional; location of the `osso.conf` file for the virtual host if one is being configured. It might, for example, be `$ORACLE_HOME/Apache/Apache/conf/osso/virtual_host_name`. Note that the `osso.conf` file for the nonvirtual host is located at `$ORACLE_HOME/Apache/Apache/conf/osso`.
- **admin_info:** Optional; username of the `mod_osso` administrator
- **admin_id:** Optional; any additional information, such as e-mail address of the administrator

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Creating and Editing a Partner Application

You can add and edit partner applications from OracleAS Single Sign-On administration pages.

The screenshot displays the 'SSO Server Administration' window with the 'Administer Partner Applications' tab selected. The left sidebar contains links for 'Edit SSO Server Configuration', 'Administer Partner Applications', and 'Administer External Applications'. The main content area shows the 'Partner Application' section with a description, a 'Delete Partner Application' section, and a table of existing applications.

Delete	Application Name	Start Date
The SSO Server (orasso)		06-SEP-2005 05:53 AM

Creating and Editing a Partner Application

You can add partner applications from the OracleAS Single Sign-On Administrative pages. To create a partner application, perform the following steps:

1. Log in to the OracleAS Single Sign-On server and navigate to the SSO Server Administration page as shown in the slide.
2. Click the Administer Partner Applications link to add and edit partner applications.
3. Click Add Partner Application to add a partner application.

Creating and Editing a Partner Application

Partner Application Login

Enter the application name, the home URL and the success URL for this application. The home URL is the application's home page. The success URL refers to the URL to be redirected to upon successful login. It must correspond to the procedure that processes the user identification information from the SSO Server.

①

Name:
Home URL:
Success URL:
Logout URL:

Application Administrator

Enter the email address and descriptive information for the contact person or administrator of this partner application.

③

Administrator Email:

Administrator Information:

Apply OK Cancel

Valid Login Timeframe

Enter the dates between which logins will be allowed through the SSO Server. An empty field implies an indefinite login timeframe.

Start Date: 8-Sep-2005 Use Format DD-MON-YYYY
End Date: Use Format DD-MON-YYYY

②

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Creating and Editing a Partner Application (continued)

The Create Partner Application page is divided into three sections. Each section is described as follows:

- **Partner Application Login:** In this section, you specify the access information of the partner application. This section includes the following fields:
 - **Name:** Specify a unique name for the partner application.
 - **Home URL:** Specify the URL to access the application home page.
 - **Success URL:** Specify the URL to the routine responsible for establishing the partner application session and session cookies. This routine should redirect the browser to the URL that the user originally requested. The URL must point to a procedure that processes the user identification information from the Single Sign-On server.
 - **Logout URL:** Specify the URL for the logout routine of the application. The single sign-off page invokes this URL in parallel with others, enabling users to simultaneously log out of the server and active partner applications.

Creating and Editing a Partner Application (continued)

- **Valid Login Timeframe:** In this section, you specify the time period for which the application can be accessed by the user through the OracleAS Single Sign-On server. The two fields are:
 - **Start Date:** Enter the date when users are first able to access the partner application through the Single Sign-On server. Use the format shown next to the field label.
 - **End Date:** Enter the end date when users are last able to access the partner application through the Single Sign-On server. If you leave this field blank, users are able to log in to the partner application indefinitely.
- **Application Administrator:** In this section, you specify the details of an administrator for the partner application. The two fields are:
 - **Administrator Email:** Use this field to enter the e-mail address of the partner application administrator.
 - **Administrator Information:** Use this optional field to enter additional information about the partner application administrator.

Editing a Partner Application

On the Administer Partner Application page, all the partner applications are listed. To edit a partner application, click the pencil icon. This page includes the following fields in addition to the fields on the Add Partner Application page:

- **ID:** The ID is automatically set when a partner application is added. It is used by the Single Sign-On server to identify the partner application.
- **Token:** The token is automatically set when a partner application is added. It is used by the Single Sign-On server to identify the partner application. The partner application must use the application token to identify itself to the Single Sign-On server when requesting authentication.
- **Encryption Key:** The encryption key is automatically set when a partner application is added. When a user tries to log in, the Single Sign-On server generates a cookie that indicates a user's identity and whether the user has been authenticated. This key is used to encrypt the login cookie.

Click the OK button to confirm the changes made on this page.

Administering External Applications

- You can add external applications to the OracleAS Single Sign-On server and configure them.
- Use the OracleAS Single Sign-On administration pages to perform the following tasks:
 - Adding an external application
 - Editing an external application
 - Storing the external application credentials in the OracleAS Single Sign-On database


ORACLE

Copyright © 2005, Oracle. All rights reserved.

Administering External Applications

You can add external applications to the Single Sign-On server to enable the single sign-on functionality. OracleAS Single Sign-On server administration pages enable you to add external applications. These applications can be accessed using the External Application portlets of OracleAS Portal.

Adding External Applications



Add External Application

Add an external application that supports form-based authentication. The settings include the application login URL, field user name and password fields, and method.

External Application Login

Enter the application name, the login URL, and the user name and password HTML field names used by the application's login form. The login URL is typically the submit action of the application's login form. It will be used in conjunction with the user name and password field names to perform a single sign-on login into this application. The login URL as well as the user name and password field names should be determined by inspecting the source of the application's standard login form. User name/id, password, additional field etc. values are not required for Basic authentication and Login URL should be a url which requires authentication.

Authentication Method

Select the authentication method used by this application. The POST method submits the credentials with the body of the form. The GET method submits the login credentials as part of the login URL.

Type of Authentication Used:

Additional Fields

Type the names and values of any additional fields that are submitted with the login form of the external application.

Field Name	Field Value	Display to User
<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
<input type="text"/>	<input type="text"/>	<input type="checkbox"/>

Copyright © 2005, Oracle. All rights reserved.

Adding External Applications

To add an external application to the OracleAS Single Sign-On server, perform the following steps:

- Log in to OracleAS Single Sign-On administration pages, and navigate to the Administer External Applications page as shown in image 1 in the slide. Click the Add External Application link. The Create External Application page is displayed. The Create External Application page is divided into three main sections:
 - External Application Login (shown as image 2 in the slide)
 - Authentication Method (shown as image 3 in the slide)
 - Additional Fields (shown as image 4 in the slide)
- In the External Application Login section, enter the following details:
 - Application Name:** Enter a name that identifies the external application. This is the default name for the external application.
 - Login URL:** Enter the URL to which the HTML login page for the external application is submitted for authentication. This is the URL to which you are redirected after the username and password are submitted.
 - Username/ID Field Name:** Enter the name of the field on the external application login Web page where you enter the username. You can find the name of the field by viewing the HTML source of the Web page.

Oracle Application Server 10g R2: Administration I 15-21

Adding External Applications (continued)

- **Password Field Name:** Enter the name of the field on the external application login Web page where you enter the password. You can find the name of the field by viewing the HTML source of the Web page.
- 3. In the Authentication Method section, enter the following details:
 - Select an option from the Type of Authentication Used drop-down list. This is the authentication method that the browser uses to pass data to the specified URL. The options are:
 - POST:** Posts data to the Single Sign-On server and submits login credentials within the body of the form
 - GET:** Presents a page request to a server, submitting the login credentials as part of the login URL
 - BASIC AUTHENTICATION:** Submits the login credentials in the application URL, which is protected by HTTP basic authentication
- 4. In the Additional Fields section, enter the following details:
 - **Field Name:** Enter the name of any additional fields on the HTML login Web page that may require user input to log in to the application. This field is not applicable if you are using basic authentication.
 - **Field Value:** Enter a default value for a corresponding field name value, if applicable. This field is not applicable if you are using basic authentication.

For example, you can configure Yahoo! Mail as an external application on the OracleAS Single Sign-On server by using the following values:

- **Application Name:** OTN
- **Login URL:** http://otn.oracle.com
- **Username/ ID Field Name:** login
- **Password Field Name:** passwd
- **Type of Authentication:** POST
- **Field Name:** .persistent Y
- **Field Value:** [off]

Click OK to add the external application.

Accessing an External Application and Storing Its Credentials

You can access the external application created from Administer External Applications page.

The screenshot shows two overlapping windows. The background window is titled 'Administer External Applications' and contains sections for 'Add External Application' and 'Edit/Delete External App'. The foreground window is titled 'Login - Yahoo! Mail' and contains a login form. The form has fields for 'Application Name' (pre-filled with 'Yahoo! Mail'), 'User Name/ID', and 'Password'. There is a checkbox labeled 'Remember My Login Information For This Application' which is checked. The dialog box has 'Login' and 'Close' buttons in the top right corner.

Copyright © 2005, Oracle. All rights reserved.

Accessing an External Application and Storing Its Credentials

After you have added the external application to the OracleAS Single Sign-On server, you can access the external application, for example Yahoo!, by clicking the link on the Administer External Applications page.

Clicking the Yahoo! link displays a new window asking you to authenticate to the external application. Enter the username and password to authenticate to the external application. If you select the Remember My Login Information For This Application check box, then OracleAS Single Sign-On server stores your credentials for the external application. The next time when you log in to the OracleAS Single Sign-On server, you are automatically logged in to the external application too.

Monitoring the OracleAS Single Sign-On Server

You can monitor the OracleAS Single Sign-On server from the OracleAS Enterprise Manager Console.

System Components					
<div>Start Stop Restart Delete OC4J Instance</div> <div>Enable/Disable Components Create OC4J Instance</div>					
Select All Select None					
Select	Name	Status	Start Time	CPU Usage (%)	Memory Usage (MB)
<input type="checkbox"/>	HTTP Server	↑	Sep 6, 2005 6:02:59 AM	0.49	110.29
<input type="checkbox"/>	Internet Directory	↑	Sep 6, 2005 5:51:35 AM	0.03	8.81
<input type="checkbox"/>	OC4J SECURITY	↑	Sep 6, 2005 6:03:04 AM	0.00	20.14
<input type="checkbox"/>	oca	↑	Sep 6, 2005 6:04:02 AM	0.00	18.07
<input type="checkbox"/>	Single Sign-On:orasso	↑	N/A	N/A	N/A
<input type="checkbox"/>	Management	↑	Sep 6, 2005 5:44:47 AM	0.55	107.31

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Monitoring the OracleAS Single Sign-On Server

To find out the status of the OracleAS Single Sign-On server, you have to access the OracleAS Single Sign-On server monitoring page.

You can access the OracleAS Single Sign-On server monitoring page by performing the following steps:

1. Log in to the Application Server Control Console. Select the appropriate OracleAS stand-alone instance (usually the OracleAS Infrastructure instance).
2. In the System Components section, select Single Sign-On:orasso as shown in the image in the slide.

The OracleAS Single Sign-On server monitoring page is displayed. The page is divided into three sections:

- **General:** You can find the following details under this heading:
 - The status of the SSO server: How long it has been up or down. A green check mark means that the server is up. A red arrow means that the server is down.
 - The metadata repository database name and version
 - The name of the HTTP Server where the SSO server resides

Monitoring the OracleAS Single Sign-On Server (continued)

- **Last 24 Hours Status Details:** This section displays the login details for the previous 24 hours. It displays the following details:

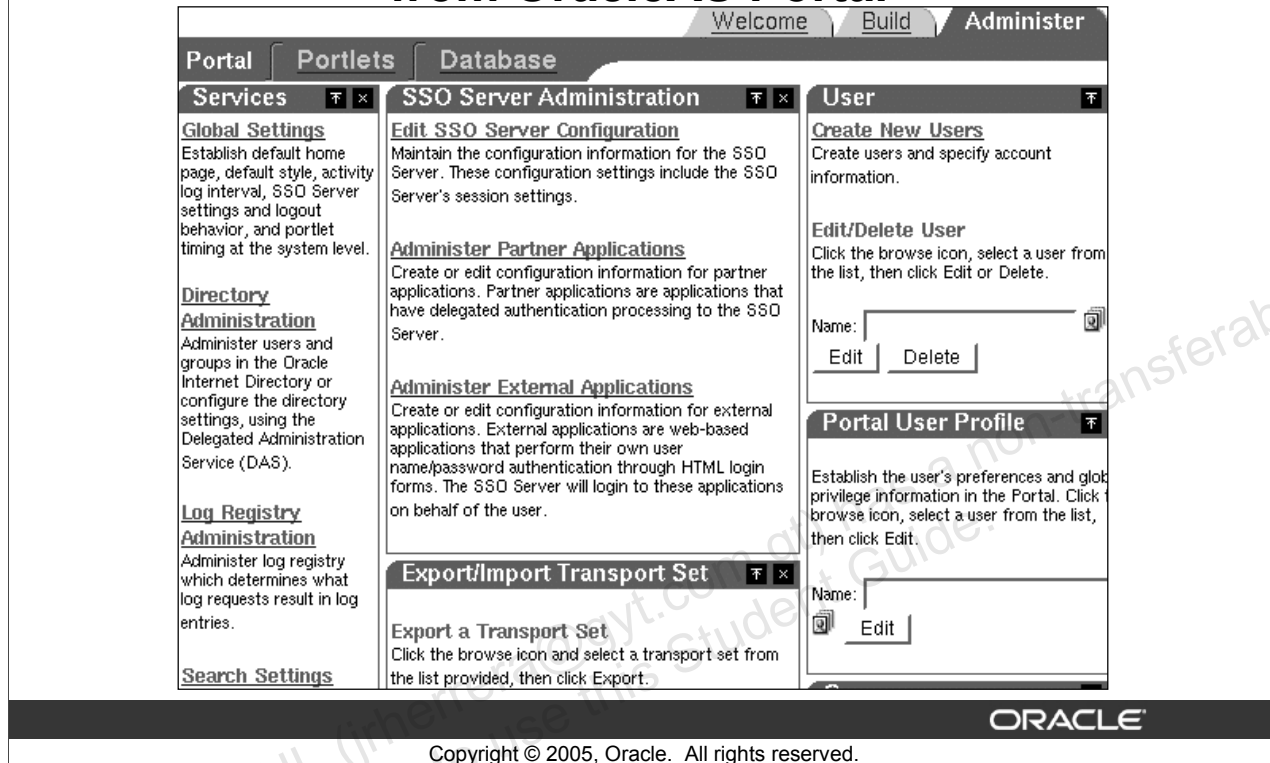
- Logins
- Successful logins
- Failed logins

All these values are in percentages and are for the previous 24 hours only.

- **Login Failure During Last 24 Hours:** This section displays details about login failures in the last 24 hours. The details are as follows:

Field	Description
Username	The user who is trying to log in
Failures	The number of login failures a user has logged during the previous 24 hours. Select this link to show details about each login failure.

Accessing the SSO Server from OracleAS Portal



Accessing the SSO Server from OracleAS Portal

When you install an OracleAS Middle Tier instance that includes OracleAS Portal, the Installer automatically adds OracleAS Portal as a partner application for OracleAS SSO. This allows you to access the portal from the OracleAS SSO Administer Partner Applications page. Similarly, you can access the SSO server from the portal:

1. Log in to the portal as `orcladmin`, and click the Administration tab.
Note: Portal administrator accounts, such as `portal` and `portal_admin`, do not have permissions to perform the administration of Oracle Internet Directory and OracleAS SSO. If you need to allow the OracleAS Portal account to perform OracleAS SSO administration, you need to explicitly give the user the privilege.
2. Click the Portal tab.
3. From the SSO Server Administration portlet, you can navigate to the relevant OracleAS SSO administration pages.

Accessing External Applications from OracleAS Portal

1. Add the External Applications portlet to a portal page.
2. Customize the External Applications portlet.

The screenshot shows the 'External Applications' portlet interface. At the top, there is a title bar with 'External Applications' and a 'Customize' button. Below the title bar, a list of applications is shown: 'My.Oracle.Com' and 'Yahoo!'. A modal dialog titled 'Select External Applications' is open, displaying instructions and a table for selecting applications.

Select External Applications
Use the check boxes to select the external applications for this portlet. For each application you select, click to supply your user name and password. You will then be logged in automatically each time you launch the application. If you wish to change the name of the application as it appears in this portlet, use the Preferred Name field to do so.

Display	Change Stored Password	Application Name	Preferred Name
<input checked="" type="checkbox"/>		My.Oracle.Com	<input type="text"/>
<input checked="" type="checkbox"/>		Yahoo!	<input type="text"/>

Accessing External Applications from OracleAS Portal

You can access external applications that are registered with OracleAS SSO from OracleAS Portal that has been added to the same SSO server as a partner application. To accomplish this, you must perform the following:

1. Add the External Application portlet to a portal page to which portal users have access and privilege to customize. This portlet provides access to the list of external applications that were added to OracleAS SSO.
2. After the External Application portlet is added to the portal page, portal users can customize the portlet by selecting which external applications they want to be displayed in the portlet and by providing their credentials to log in to the selected applications.

Summary

In this lesson, you should have learned how to:

- Discuss the components of the OracleAS Single Sign-On server
- Explain the OracleAS Single Sign-On server authentication flow
- Manage and configure the OracleAS Single Sign-On server
- Administer the partner and external applications
- Monitor the OracleAS Single Sign-On server
- Access the OracleAS Single Sign-On server from OracleAS Portal

ORACLE

Copyright © 2005, Oracle. All rights reserved.

16

Managing Access Using Oracle Delegated Administration Services

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Explain the Oracle Delegated Administration Services architecture**
- **Describe how Oracle Delegated Administration Services works**
- **Start and stop Oracle Delegated Administration Services**
- **Access the Oracle Delegated Administration Services home page**
- **Provide an overview of Oracle Internet Directory Self-Service Console**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

- **Manage user entries by using Oracle Delegated Administration Services**
- **Manage group entries using Oracle Delegated Administration Services**
- **Create Identity Management Realm**
- **Access Oracle Delegated Administration Services from OracleAS Portal**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Delegated Administration Services

- **Oracle Delegated Administration Services:**
 - Is a set of individual, predefined Web-based services
 - Is used to perform directory operations on behalf of users
 - Includes a Web application called Oracle Internet Directory Self-Service Console
- **You can use Oracle Delegated Administration Services to:**
 - Modify data that you are authorized to manage
 - Manage subscriber-level information
 - Manage site-level information

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Delegated Administration Services

Delegated Administrative Service Units

Delegated administration enables you to store all data for users, groups, and services in a central directory while distributing the administration of that data to other administrators and end users. Oracle Delegated Administration Services:

- Is a set of individual, predefined Web-based services
- Performs directory operations on behalf of a user
- Makes it easier to develop and deploy administration solutions for Oracle Internet Directory-enabled applications
- Can be used to delegate certain functions to an administrator or a user

Oracle Delegated Administration Services performs the following operations:

- **Maintains user entries:**
 - Search user entries
 - Create user entries
 - Modify user entries
 - Delete user entries
 - Change user passwords
 - Assign privileges to users to perform certain tasks

Oracle Delegated Administration Services (continued)

- **Maintains group entries:**
 - Search group entries
 - Create group entries
 - Modify group entries
 - Delete group entries
 - Assign privileges to groups

These operations are performed by Oracle Delegated Administration Services on behalf of the application. After the operation, Oracle Delegated Administration Services provides a user interface that displays the result of the operation.

Oracle Internet Directory Self-Service Console

Oracle Delegated Administration Services includes a prebuilt, Web application called Oracle Internet Directory Self-Service Console. You can access the directory data of an application by using this interface.

Oracle Internet Directory Self-Service Console enables you to perform the following operations:

- You can manage the directory data that you are authorized to manage, and change your personal information, such as telephone numbers, office locations, and any application preferences.
- As a subscriber administrator, you can perform the following operations:
 - Manage subscriber-level information
 - Provision a new user or group
 - Manage users and groups within a subscriber
 - Administer directory attributes that are not associated with a particular application
- As a site administrator, you can perform the following operations:
 - Manage site-level information, such as site configurations
 - Manage subscriber-level information, such as creating new subscribers and changing their privileges

Benefits of Oracle Internet Directory Self-Service Console

- **Faster development and deployment of directory-enabled applications**
- **Secure access to directory**
- **Easy to use for application users**
- **Ability for sites to delegate directory data administration**

ORACLE

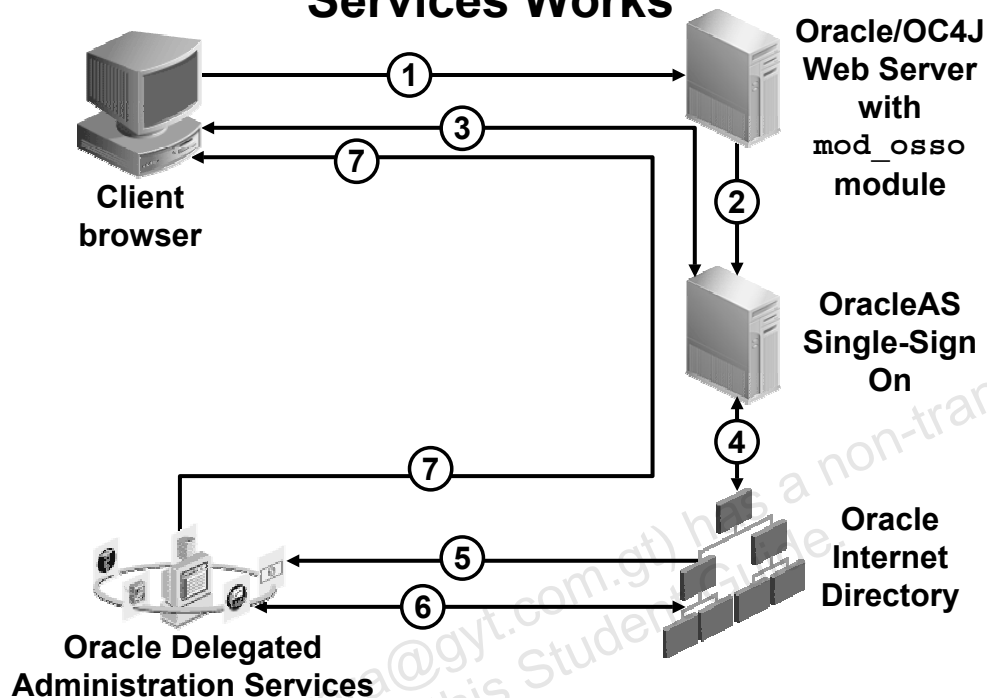
Copyright © 2005, Oracle. All rights reserved.

Benefits of Oracle Internet Directory Self-Service Console

Oracle Delegated Administration Services and Oracle Internet Directory Self-Service Console provide the following benefits:

- **Faster development and deployment of directory-enabled applications:** By using Oracle Delegated Administration Services, you can easily develop the tools required by your application to administer directory data.
- **Secure access to directory:** Oracle Delegated Administration Services uses the proxy-user feature of Oracle Internet Directory to perform various operations on behalf of users. This proxy access is the single-point access for all operations performed and it improves the directory security. You, as a directory administrator, need not provide super user access to various directory administration tools that applications require.
- **Ease of use:** Users of multiple directory-enabled applications interface with a single set of services for administering application-related directory data.
- **Ability for sites to delegate directory data administration:** Oracle Delegated Administration Services enables you to delegate the administration of defined directory data to subscriber administrators and application end users. This makes it easier for sites to manage directory data.

How Oracle Delegated Administration Services Works



Copyright © 2005, Oracle. All rights reserved.

How Oracle Delegated Administration Services Works

Oracle Delegated Administration Services can be enabled to work in conjunction with OracleAS SSO. The following occurs when you perform a search for another user or group using Oracle Delegated Administration Services when OracleAS SSO is enabled:

1. The user accesses Oracle Delegated Administration Services using Oracle HTTP Server with the `mod_osso` module.
2. If this is the first time during a session that the user is accessing Oracle Delegated Administration Services, then Oracle HTTP Server transparently redirects the user to the SSO server for authentication.
3. The SSO, by way of Oracle HTTP Server, prompts the user for the username and password.
4. The SSO verifies the user's credentials by comparing the values that the user entered with the corresponding values stored in Oracle Internet Directory.
5. If it successfully verifies the username and password, then SSO directs the user to Oracle Delegated Administration Services. It also sends an encrypted parameter that contains the user identifier to Oracle Delegated Administration Services. If the SSO authentication fails, the user is given an error message and asked to enter the credentials again.

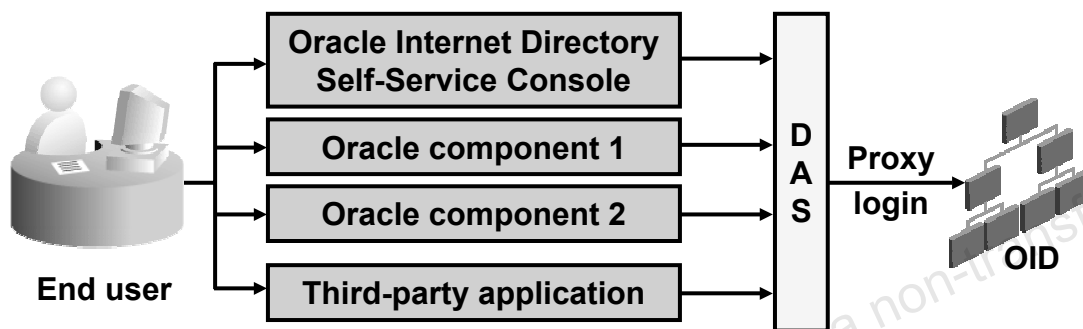
How Oracle Delegated Administration Services Works (continued)

6. To enable the user to access the directory, Oracle Delegated Administration Services:
 - Logs in to Oracle Internet Directory on the end user's behalf as proxy user that has the privileges to switch identities
 - Performs a second bind to the directory; this time using the distinguished name (DN) of the end user

When Oracle Delegated Administration Services logs in to the directory server by using the DN of the end user, the directory server:

- Recognizes the second bind as an attempt by the proxy sever to switch to the end user's identity
 - Trusts the authentication granted to the end user by Oracle Delegated Administration Services
 - Allows this second bind to succeed without requiring the end user's password
7. Oracle Delegated Administration Services compiles the LDAP result into an HTML page and sends it to the client Web browser.

Oracle Delegated Administration Services Proxy User



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Oracle Delegated Administration Services Proxy User

The Oracle components log in to the directory server to perform various operations on behalf of the end user. To perform these operations, the Oracle components log in to the Oracle Internet Directory server as a proxy user. The proxy user enables the Oracle component to switch its identity to that of the end user.

The proxy user is a kind of user typically employed in an environment with a middle tier, such as a firewall. In such an environment, the end user authenticates to the middle tier. The middle tier then logs in to the directory on the end user's behalf. A proxy user has the privilege to switch identities. After it has logged in to the directory, it switches to the end user's identity. Then, it performs operations on the end user's behalf, using the authorization that is appropriate to that particular end user.

Starting and Stopping

You can start and stop Oracle Delegated Administration Services by using the following commands:

- **To start Oracle Delegated Administration Services:**
 - `$ORACLE_HOME/opmn/bin/opmnctl startproc type=oc4j instancename=OC4J_SECURITY`
- **To stop Oracle Delegated Administration Services:**
 - `$ORACLE_HOME/opmn/bin/opmnctl stopproc type=oc4j instancename=OC4J_SECURITY`

You can also use the Application Server Control Console to start or stop Oracle Delegated Administration Services.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting and Stopping

- You can start Oracle Delegated Administration Services by using the following command:
 - `$ORACLE_HOME/opmn/bin/opmnctl startproc type=oc4j instancename=OC4J_SECURITY`
- You can stop Oracle Delegated Administration Services by using the following command:
 - `$ORACLE_HOME/opmn/bin/opmnctl stopproc type=oc4j instancename=OC4J_SECURITY`
- You can also use the Application Server Control Console to start or stop Oracle Delegated Administration Services. You can access the Application Server Control Console by using the URL `http://<host name>.<domain>:<Application Server Control port>`. Select the OracleAS Infrastructure instance whose components you want to access. Then, select the OC4J_SECURITY component and click the Start or Stop button to start or stop Oracle Delegated Administration Services, respectively.

Verifying That Oracle Delegated Administration Services Is Running

You can follow these steps to verify that Oracle Delegated Administration Services is running:

Verify that:

- 1. Oracle HTTP Server is running**
- 2. OC4J JVM is running**
- 3. The Oracle Delegated Administration Services Web site is running**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Verifying That Oracle Delegated Administration Services Is Running

You can verify that Oracle Delegated Administration Services is running by checking the following:

1. Oracle HTTP Server is running; run the following command:

```
ps -ef | grep http
```

This displays the name of the Oracle HTTP Server process, if it is running on the server.
2. OC4J JVM is running; run the following command:

```
ps -ef | grep java
```

This displays the name of the Java process, if it is running on the server.

You can verify the steps above in a Windows environment by checking the status of these components on the Application Server Control Console Web site. You can access the Application Server Control Console Web site by entering `http://hostname.domain:Application Server Control port` in any browser, and see whether the Oracle Delegated Administration Services component on that Oracle Application Server instance is running or not.

Verifying That Oracle Delegated Administration Services Is Running (continued)

3. To verify that the Oracle Delegated Administration Services Web site is running, point your browser to `http://<host name>.<domain>:<Oracle HTTP Server port>/oiddas`. The host name here is the name of the computer where Oracle HTTP Server is running. This takes you to the Oracle Delegated Administration Services home page from where you can manage user and group information.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Configuring the Default Identity Management Realm–Specific Context

1. Log in to Oracle Delegated Administration Services as the administrator.
2. Click the Configuration tab.
3. Enter values for the required fields in the:
 - Directory section
 - Logo Management section
4. Click Submit to save your changes.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring the Default Identity Management Realm–Specific Context

After installing Oracle Delegated Administration Services, configure the default subscriber context, that is, the root entry of the naming context that contains all entries for the default subscriber.

To configure the default subscriber, perform the following steps:

1. Log in to Oracle Delegated Administration Services as the administrator. The default administrator username is `orcladmin` and the default password is the same that was specified at the time of installation for `ias_admin`.
2. Click the Configuration tab:
 - In the Login Name field, enter the attribute by which you want users to identify themselves when they log in: `cn`, `UID`, `EmployeeNumber`, `SSN`.
 - In the User Search Base context field, enter the DN of the entry under which the user entries for this subscriber are located.
 - In the Group Search Base context field, enter the DN of the entry under which group entries for this subscriber are located.
 - In the Search Return Limit field, enter the number of entries that you want to display in the search results.

Configuring the Default Identity Management Realm–Specific Context (continued)

3. In the Logo Management section, perform the following:
 - Select the Enable Realm Logo check box to display the logo in the upper-left corner.
 - Select the Enable Product Logo check box to display the product name in the upper-left corner.
 - In the Update Realm Logo field, enter the path and file names of this realm's logo, or navigate to it by clicking Browse.
4. When you have entered the location of the corporate image logo file, click Submit to save your changes.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Configuring User Entries

1. Click User Entry on the Configuration tabbed page.
2. Add an object class for user entries.
3. Add attributes to user entries.
4. Configure the attributes of user entries.
5. Customize the way categories of attributes are displayed to a user.
6. Select the attributes to be displayed when a search is performed.
7. Enable role assignment in the user management interface.
8. Click Finish.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring User Entries

When a user creates or edits a user entry, the user interface displays various categories, for example, basic information, password, and photo, each with its own set of attributes. You can customize the way Oracle Delegated Administration Services displays these categories and the corresponding attributes by performing the following steps:

1. Click User Entry on the Configuration tabbed page. The Configure User Object Classes window is displayed, listing the existing object classes for user entries.
2. To add an object class for user entries, perform the following steps:
 - Click Add Object Class. The All Object Classes window is displayed.
 - Select an object class that you want to add, and click Add. This returns you to the Configure User Object Classes window.

To add more object classes, repeat the steps.

3. To add attributes to the user, perform the following steps:
 - In the Configure User Object Classes window, click Next to open the Configure User Attributes window.
 - Click Add New Attribute to open the Add New Attribute window.
 - From the Directory Attribute Name combo box, select the attribute that you want to add.

Configuring User Entries (continued)

3. To add attributes to the user, perform the following steps: (continued)
 - Enter values in the following fields:
 - UI Label: Friendly name of the attribute
 - Required: Specifies whether you want the attribute to be specified as mandatory
 - Viewable: Specifies whether you want the attribute to appear in search results
 - Self Editable: Specifies whether you want the attribute to be modifiable by the user
 - Password Reset Validation: Specifies whether this attribute can be used to validate the user if the user forgets his or her password
 - Searchable: Specifies if this attribute can be used to perform a search
 - UI Type: Specifies the type of interface for this field: Single Line Text, Multi Line Text, Predefined List, Date, Browse and Select, and Number
 - Click Done to return to the Configure User Attributes window. The attribute that you just chose is now listed in the Directory Attribute Name list.
4. To configure attributes of user entries, perform the following steps:
 - In the Directory Attribute Name list, select the attribute that you want to configure, and click Edit to open the Editing Attribute window.
 - Specify configurations for the selected attributes.
 - Click Done. This returns you to the Configure User Attributes window. The attribute configurations you just made are now reflected in the Directory Attribute Name list.

To configure more attributes, repeat these steps.
5. To customize the way categories of attributes are displayed to a user when adding or modifying user entries, perform the following steps:
 - Click Next in the Configure User Attributes window to open the Configure Attribute Categories window. This window contains a table listing the existing categories, the name displayed to the user, and the display order of each.
 - To add a new category, click Create to display a new row in the table.
 - To modify the display name of a category, in the UI Label column, edit the field for each attribute that you want to modify. To designate the display order, click Order Category. The Order Category window displays the various categories you specified. Use the up and down arrows to move them into the desired order.
 - To delete a category, select the category, and click Delete.
6. To set the attribute columns that are displayed when a search is performed, perform the following steps:
 - In the Configure Attribute Categories window, click Next. The Configure Search Table Columns window displays with the two list columns representing All Attributes and Selected Attributes. You can move items between the two lists—either one item at a time or all items at once. You can move a maximum of seven attributes under the Selected Attributes list.
 - Within the Selected Attributes list for each category, set the attribute display order by using the up and down arrow buttons in the list on the right.
7. To enable role assignment in the user interface, perform the following steps:
 - In the Configure Search Table Columns window, click Next to open the Configure Roles window.
 - Select the “Enable Role assignment in the user management interface” option to enable the role assignment in the user interface.
 - You can add new roles by clicking Add Role. The Search and Select: Role window is displayed. Search for a role in this window and then select it. The new role is displayed in the Name column.
 - To delete a role, select the role and then click Delete.
8. When you have finished configuring user entries, click Finish.

Managing Users, Groups, and Subscribers

You can use Oracle Delegated Administration Services to:

- **Search for user and group entries**
- **Maintain user entries**
 - **Create user entries**
 - **Modify user entries**
 - **Delete user entries**
- **Change passwords**
- **Create group entries**
- **Modify and delete group entries**
- **Assign privileges to users and groups**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing Users, Groups, and Subscribers

User data in the repository can consist of generic user data, as well as application data. Some user data can be read and modified by the user or may be read-only for others, while other portions are invisible to the user.

In Oracle Internet Directory, it is possible to restrict an application to add or delete only specific object classes from a user entry.

User credentials are managed by using Oracle Delegated Administration Services.

Searching for User and Group Entries

- **To search for user entries, perform the following:**
 1. Click the Directory tab, and click User.
 2. In the “Search for user” field, enter the first few characters of the name of the user.
 3. Click Go to display the search results.
- **To search for group entries, perform the following:**
 1. Click the Directory tab, and click Group.
 2. In the Search Group Name text box, enter the first few characters of the name of the group.
 3. Click Go to display the search results.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Searching for User and Group Entries

To search for users, on the Directory tabbed page of the Oracle Delegated Administration Services home page, perform the following steps:

1. Click User.
2. In the “Search for user” field, enter the first few characters of the name of the user. For example, if you are searching for “John Smith,” you can enter Joh.
3. Click Go to display the search results.

Searching for Group Entries Using DAS

To search for groups, on the Directory tabbed page, perform the following steps:

1. Click Group.
2. In the Search Group Name text box, enter the first few characters of the name of the group you are searching for.
3. Click Go to display the results that match the criteria you entered.

Maintaining User Entries

- **To create a user entry, perform the following:**
 1. On the Directory tabbed page, click User and click Create.
 2. Enter the user details and click Submit.
- **To modify a user entry, perform the following:**
 1. On the Directory tabbed page, search for the user whose entry you want to modify.
 2. Click Edit to modify the user entry.
- **To delete a user entry, perform the following:**
 1. On the Directory tabbed page, search for the user whose entry you want to delete.
 2. Click Delete to remove that user entry.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Maintaining User Entries

Creating User Entries Using the Oracle Delegated Administration Services

1. Click the Directory tab, and click User.
2. Click Create to open the Create User window.
3. Enter the values in the appropriate fields.
4. Verify that you have entered all information correctly, and click Submit.

Modifying User Entries Using the Oracle Delegated Administration Services

1. Click the Directory tab, and perform a search for the user whose entry you want to modify.
2. Select the user whose entry you want to modify, and click Edit to open the Edit User window.
3. Modify values in the required and other appropriate fields, and click Finish.

Deleting User Entries Using Oracle Delegated Administration Services

1. Click the Directory tab, and perform a search for the user whose entry you want to delete.
2. Select the user whose entry you want to delete, and click Delete.

Changing Passwords

To change your own password by using Oracle Delegated Administration Services, perform the following steps:

- 1. Log in to Oracle Delegated Administration Services.**
- 2. Click the My Profile tab.**
- 3. Click the Change My Password tab.**
- 4. Enter your old password.**
- 5. Enter and confirm the new password.**
- 6. Click Submit.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Changing Passwords

To change your own password using Oracle Delegated Administration Services, perform the following steps:

1. Log in to Oracle Delegated Administration Services.
2. Click the My Profile tab.
3. Click the Change My Password tab.
4. In the Old Password field, enter your current password.
5. In the New Password field, enter your new password and confirm the new password in the Confirm New Password field.
6. Click Submit.

Changing Another User's Password

To change the password of another user, perform the following steps:

1. Click the Directory tab.
2. Perform a search for the user.
3. Select the user entry, and click Edit.
4. In the Password Management section, enter and confirm the new password.
5. Click Submit.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Changing Another User's Password

To change another user's password by using Oracle Delegated Administration Services, perform the following steps:

1. Click the Directory tab.
2. Perform a search for the entry of the user whose password you want to change.
3. Select the user entry, and click Edit to open the Edit User window.
4. In the Password Management section, enter and confirm the password that you want to assign to the user.
5. Click Submit.

To change another user's password, you must have the necessary access rights.

Creating Group Entries

1. Click the Directory tab, click Group, and then click Create. The Create Group window appears.
2. Enter the name, friendly name, and description of the group.
3. In the User Members section, click Add User.
4. Search for the users whom you want as members of this group.
5. In the Group Members section, click Add Group.
6. Search for the group that you want to specify as the member of the group you just created.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Creating Group Entries

To create group entries by using Oracle Delegated Administration Services, perform the following steps:

1. Click the Directory tab, click Group, and then click Create. This displays the Create Group window.
2. In the Name field of the Basic Information section, enter the name for this group.
3. In the Display Name field, enter the friendly name.
4. In the Description field, enter a brief description of this group.
5. To hide this group entry from all users except its owners, in the Group Visibility field, select Private. The creator of the group is automatically a group owner. To specify an additional owner of this group, perform the following steps:
 - a. In the Owners section, click Add User to open the Search and Select: User window.
 - b. Perform a search for the entry of the user you want to specify as an owner of the group, and click Select. This returns you to the Create Group window. The user you specified is in the owner's list.
 - c. To remove an owner, in the Owners section, select the owner's name, and click Remove.

Creating Group Entries (continued)

To add a user as a member of this group, perform the following steps:

1. In the Members section, click Add User to open the Search and Select: User window.
2. Perform a search for the entry of the user you want to specify as a member of this group, and click Select. This returns you to the Create Group window. The user is specified in the Members section.
3. To remove a user from this group, in the Members section, select the user's name, and click Remove.

To add a group as a member of this group, perform the following steps:

1. In the Group Members section, click Add Group to open the Search and Select: Group window.
2. Perform a search for the entry of the group that you want to specify as a member of this group, and click Select. This returns you to the Create Group window. The group you specified is listed in the Members section.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Modifying and Deleting Group Entries

- **To modify group entries, perform the following:**
 1. Click the Directory tab.
 2. Perform a search for the group.
 3. Click Edit to modify the group.
- **To delete group entries, perform the following:**
 1. Click the Directory tab.
 2. Perform a search for the group.
 3. Click Delete to delete the group.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Modifying and Deleting Group Entries

To modify group entries using Oracle Delegated Administration Services, perform the following steps:

1. Click the Directory tab, and perform a search for the group entry that you want to modify.
2. Select the group entry you want to modify, and click Edit to open the Edit Group window.
3. Modify values in the required and other appropriate fields, and click Finish.

To delete group entries by using Oracle Delegated Administration Services, perform the following steps

1. Click the Directory tab, and perform a search for the group whose entry you want to delete.
2. Select the group whose entry you want to delete, and click Delete.

Assigning Privileges to Users and Groups

- **Users and groups can be granted the privilege to:**
 - Create and edit users and groups
 - Assign privileges to other users and other groups
- **You can also revoke privileges from users and groups.**
- **To assign privileges to a user, perform the following steps:**
 1. **On the Directory tabbed page, search for the user or group.**
 2. **Select the user or group, and click Assign Privileges to display a list of privileges.**
 3. **Select the privileges that you want to assign to this user or group.**
 4. **Click submit, or to assign privileges to another user or group, click Specify Other Users or Specify Other Groups.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Assigning Privileges to Users and Groups

Privileges that can be assigned to a user are:

- **Creation:** To create user entries
- **Editing:** To modify user entries
- **Deleting:** To delete user entries
- **Privilege assignment to users:** To assign access rights to users
- **Group creation:** To create group entries
- **Group editing:** To modify group entries
- **Group deletion:** To delete group entries
- **Privilege assignment to groups:** To assign access rights to groups
- **Service Management:** To manage services
- **Account Management:** To enable or disable user accounts
- **Allow Delegated Administration Service configuration:** To configure user and realm information

Managing Services

- **A service can be a single application or a bundle of applications that perform a coherent set of tasks.**
- **It is supplied by a service provider to either individuals or groups, called service recipients.**
- **To access a service, a service recipient must be subscribed to it. In the subscription process, an administrator for either an identity management realm or a service provider creates a subscription list. This list specifies which service recipient users can use the service and for how long.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing Services

The Services window lists the various services available in your domain, if they are configured. From this window, you can choose the appropriate button to:

- **Edit Services:** You can change the display name and network for each service.
- **Edit Subscriptions:** You can specify service recipients, the users on their respective subscription lists, and the time frame within which those users can access the services.

Managing Accounts

As an Oracle Internet Directory administrator, you can perform the following tasks on user accounts:

- **Unlock a user account.**
- **Enable or disable a user account.**

ORACLE Identity Management Provisioning Console

Logout Realm Manage

Home My Profile Directory Conf

Users | Groups | Services | Applications

Logged in as orcladm

Users

Search user1 Go Advanced Search Provisioning Search

Search is conducted over attributes listed below.

View Edit Privileges Delete Unlock Enable Disable Create Bulk

Select	User ID	Email Address	First Name	Last Name	Job Title	Work Phone	Locked	Enabled
<input type="checkbox"/>	user1	user1@oracle.com		user				✓

Search Attributes Email Address, First Name, Last Name, User ID

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing Accounts

If a user's account is locked for any reason (for example, for entering a wrong password for more than the specified number of tries), then you can unlock that user account from Oracle Internet Directory Self-Service Console, instead of resetting the user's password using Oracle Directory Manager or command-line tool.

To unlock a user's account, perform the following steps:

1. Search for the user account by using Self-Service Console.
2. Select the account that you want to unlock, and click Unlock.

If a user's account has been temporarily disabled, then you can enable it by performing the following steps:

1. Search for the user account by using Self-Service Console.
2. Select the account that you want to enable, and click Enable.

You can also temporarily suspend or disable a user's account. To disable a user's account, perform the following steps:

1. Search for the user account by using Self-Service Console.
2. Click Disable Accounts. This displays a list of disabled accounts.
3. Select the account that you want to disable, and click Disable.

Creating Identity Management Realms

The screenshot shows the 'Create Identity Management Realm' window in the Oracle Identity Management Provisioning Console. The window has a title bar with 'ORACLE Identity Management Provisioning Console' and navigation buttons for 'Home' and 'Help'. Below the title bar is a breadcrumb trail 'Identity Management Realms >' and 'Cancel'/'Submit' buttons. The main content area is divided into two sections: 'Basic Information' and 'Logo Management'. The 'Basic Information' section contains three text input fields: '* Realm Name', 'Realm Contact', and 'Description'. A note below these fields states '* indicates a Required Field.' The 'Logo Management' section has a sub-header 'Current Logo' and three options: 'Enable Realm Logo' with a checked checkbox, 'Enable Product Logo' with a checked checkbox, and 'Upload Realm Logo' with a 'Browse...' button next to it. A large, diagonal watermark reading 'HERBERT RAUL (hera@gyt.com.gt) has a non-transferable license to this Student Guide.' is overlaid on the right side of the form.

Copyright © 2005, Oracle. All rights reserved.

Creating Identity Management Realms

As an Oracle Internet Directory administrator, you can create an entry for an identity management realm that specifies:

- The name of the realm and that of the contact person for the realm
- The attribute by which you want users to identify themselves when they log in to the realm
- The root entries of the user search base and the group search base; that is, the locations in the directory information tree where entries for the realm users and groups are contained
- The display of realm and product logos

To create a new Identity Management Realm, perform the following:

1. Click the Realm Management icon at the top right of Oracle Internet Directory Self-Service Console. This displays the Identity Management Realm window.
2. Click Create. The Create Identity Management Realm window appears.
3. Enter values in the fields of the Create Identity Management Realm window.
4. Click Submit.

Creating Identity Management Realms (continued)

The Create Identity Management Realm page is divided into two sections:

- Basic Information
- Logo Management

The Basic Information section includes the following fields:

- **Realm Name:** Enter a relatively short version of the name of the realm for the realm for this subscriber. The name you enter is used to create the DN for this realm entry. This field is mandatory.
- **Realm Contact:** Enter the name of the person to contact for any issues regarding this realm.
- **Description:** Enter any additional information about this realm. This field is optional.

The Logo Management section includes the following fields:

- **Enable Realm Logo:** Select to display the realm logo on the Identity Management Realm Configuration page.
- **Enable Product Logo:** Select to display the product logo on the Identity Management Realm Configuration page.
- **Update Realm Logo:** Enter the path and file name of the logo for this realm or, alternatively, navigate to it by clicking Browse.

Accessing Oracle Delegated Administration Services from OracleAS Portal

OracleAS Portal provides links to Oracle Delegated Administration Services in:

- **The User portlet: To manage user information stored in Oracle Internet Directory**
- **The Group portlet: To manage group information stored in Oracle Internet Directory**
- **The Services portlet: To access Oracle Internet Directory Self-Service Console**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Accessing Oracle Delegated Administration Services from OracleAS Portal

In addition to querying the directory for user and group information, OracleAS Portal provides users with the means to add and modify user and group information. To change information in the directory, OracleAS Portal provides links to Oracle Delegated Administration Services in the following administrative portlets:

- The User portlet enables you to create and update users. Only a user who is a member of the OracleDASCreateUser, OracleDASEditUser, or OracleDASDeleteUser privilege groups can see the User portlet.
- The Group portlet enables you to create and update groups. The OracleDASCreateGroup privilege is granted to the AUTHENTICATED_USERS group. Therefore, every authenticated portal user has the ability to create a group. Users can only edit or delete a group if they are the group's owner or a member of the OracleDASEditGroup or OracleDASDeleteGroup privilege group, respectively.
- The Services portlet includes the Directory Administration link that takes you to Oracle Internet Directory Self-Service Console from which you can perform administrative tasks in the directory.

Granting Privileges to OracleAS Portal Users by Using Roles

Oracle Delegated Administration Services roles:

- **Provide a convenient mechanism to grant a set of privileges to OracleAS Portal users upon their creation**
- **Are based on an existing group**
- **Can include Oracle Internet Directory– and OracleAS Portal–specific privileges**
- **Can be created and managed by a portal administrator**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Granting Privileges to OracleAS Portal Users by Using Roles

In many cases, it is more efficient to use Oracle Delegated Administration Services roles to assign privileges rather than the more granular, per-user approach. When creating portal users using the Oracle Delegated Administration Services user interface, you may notice a section called Roles Assignment on the Create User page. Roles provide a very convenient mechanism by which users can be created and granted a set of privileges simultaneously.

When a check box is selected for a given user, the selected user is granted the designated role upon creation.

As a portal administrator, you can create your own Oracle Delegated Administration Services roles and preassign any combination of Oracle Internet Directory and OracleAS Portal privileges to them:

1. Create a group by using the Group portlet and assign Oracle Internet Directory privileges to the group. For example, if you are creating the user administrator group, then you can select the Allow user creation, the Allow user editing, and the Allow user deletion privileges.
2. Assign portal-specific privileges by using the Portal Group Profile portlet. For example, you can grant the Manage User Profiles portal privilege to the user administrator group.

Granting Privileges to OracleAS Portal Users by Using Roles (continued)

3. Make the group a role so that it appears in the list of roles on the Create User page. To accomplish this, you must add a new role on the Configure Roles page of the User Entry Wizard in Oracle Internet Directory Self-Service Console.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Disabling the Privilege Assignment Section

1. Log in to the portal product schema in SQL*Plus.
2. Set the `das_enable_pa` Oracle Internet Directory configuration entry in the Portal repository to no ('N').
3. Commit the change.
4. Invalidate the User portlet cache in OracleAS Web Cache.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Disabling the Privilege Assignment Section

To encourage the use of Oracle Delegated Administration Services roles rather than direct privilege assignment, you can turn off the detailed privilege assignment section of the Create Users page, when it is called from the User portlet. To implement this change, you must update a configuration entry in the OracleAS Portal repository:

1. Log in to the portal product schema (PORTAL) via SQL*Plus.
2. Enter the following commands:

```
SQL> exec wwsec_oid.set_preference_value('das_enable_pa',  
'N');  
SQL> commit;
```
3. Invalidate the User portlet in OracleAS Web Cache.

Summary

In this lesson, you should have learned how to:

- **Explain the Oracle Delegated Administration Services architecture**
- **Describe how Oracle Delegated Administration Services works**
- **Start and stop Oracle Delegated Administration Services**
- **Access the Oracle Delegated Administration Services home page**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Summary

- **Provide an overview of Oracle Internet Directory Self-Service Console**
- **Manage user entries using Oracle Delegated Administration Services**
- **Manage group entries using Oracle Delegated Administration Services**
- **Create Identity Management Realm**
- **Access Oracle Delegated Administration Services from OracleAS Portal**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

17

Managing and Configuring OracleAS Certificate Authority

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Discuss OracleAS PKI components**
- **Explain how SSL works**
- **Describe certificate provisioning using Oracle Application Server Certificate Authority (OCA)**
- **Explain the OCA architecture**
- **Access OCA administration pages**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS PKI Components

OracleAS PKI:

- **Implementation simplifies the process of implementing security**
- **Integrates authentication with user repository and applications**
- **Includes:**
 - **Secure sockets layer (SSL)**
 - **Oracle Internet Directory and OracleAS Single Sign-On**
 - **Oracle Application Server Certificate Authority**
 - **Containers, wallets, and Oracle Wallet Manager (OWM)**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS PKI Components

To implement public key infrastructure (PKI), an organization must acquire a certificate to use for authentication. This involves multiple steps, such as filling in the appropriate details and submitting it to a proper registration authority. After the authority validates and returns the approved form, you must deliver the approved form to a certificate authority. The certificate authority processes this form and issues a certificate.

Oracle Application Server implementation of PKI removes and replaces the steps, thereby reducing delays and reducing costs. This is achieved by integrating the authentication function, the user repository, and applications.

The components of OracleAS PKI include:

- **Secure Sockets Layer (SSL)**
Secure sockets layer (SSL) is a secure protocol that transmits any communication over the Internet in an encrypted form. SSL uses public key cryptography to enable authentication, encryption, and data integrity.
- **Oracle Internet Directory and Single Sign-On**
Oracle Internet Directory enables PKI-based single sign-on by providing the central repository for authentication credentials.

OracleAS PKI Components (continued)

- **Certificates and Oracle Application Server Certificate Authority**

A digital certificate is an electronic document that establishes credentials for any transactions on the Web. Certificates are issued by an authorized certificate authority. A certificate contains a username, expiration date, copy of the certificate holder's public key, and the digital signature of the certificate-issuing authority.

There are different types of certificates, each with different functions:

- **Root or authority certificates:** Root certificates are self-signed certificates that create the base or root of a certification authority.
- **Web server certificates:** Web server certificates (also called server-side certificates) are used to secure Web communications to and from servers.
- **Client certificates:** Client certificates (also called end-entity certificates) include personal or user certificates.

A certificate authority (CA) is a third party that verifies the credentials presented by an organization and issues security certificates used in SSL connections. The CA authenticates the certificate owner's identity and the services that the owner is authorized to use. An example of the CA is VeriSign.

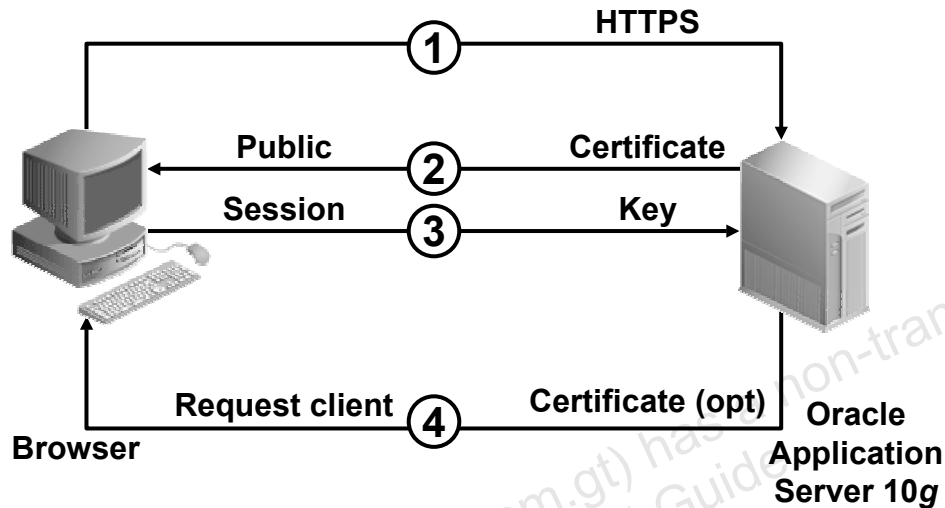
When you install Oracle Application Server, you can configure the installation to include Oracle Application Server Certificate Authority. You can use OracleAS Certificate Authority to administer and manage the complete certificate life cycle. This is typically useful for internal implementation and testing purposes. However, in a production environment, you can move from this internal implementation to requesting a certificate from a trusted third party.

- **Containers, Wallets, and Oracle Wallet Manager (OWM)**

After a certificate is available, it is necessary to provide a container for these certificates. Digital certificates can be stored in an LDAP-compliant directory or in a wallet. A wallet is a transparent database that can be used to manage authentication data, such as keys and certificates needed by SSL. The Personal Information Exchange Syntax (PKCS #12) standard provides specifications for these containers. Administrators use a tool, such as OWM, to manage security credentials on the server, and wallet owners use OWM to manage security credentials on clients.

OWM is a Java-based application that enables users or security administrators to manage public-key security credentials on both Oracle clients and database servers. It creates a wallet that can be opened using Oracle Enterprise Login Assistant. OWM creates a public-private key pair and manages credentials for a user. It issues PKCS #10 certificate requests to the certificate authority and installs the issued certificate in the wallet. OWM is shipped with trusted certificates from the well-respected root authorities VeriSign, RSA, and GTE CyberTrust, and can use a site's own in-house certificate authority. OWM can upload wallets to or download wallets from Oracle Internet Directory.

How SSL Works



ORACLE

Copyright © 2005, Oracle. All rights reserved.

How SSL Works

The image in the slide shows the steps that are involved in the authentication of a client by a server using SSL. The steps are described as follows:

1. The client initiates a connection to the server by requesting the SSL-enabled port on the server. The client uses HTTPS protocol instead of HTTP.
2. The server asserts its site identity by signing its server certificate and sending it to the client.
3. The client uses the server's public key to verify that the owner of the certificate is the same user who signed it. The client verifies the credentials of the CA. If the CA is unknown, the client program informs its user that this certificate was issued by an unknown CA. The user manually verifies that the site certificate was issued by a trusted third party for the exact site that the user is visiting. The client generates a premaster secret (a random string of bytes) and encrypts it using the server's public key. This encrypted premaster secret is then used as the basis for encryption keys and message authentication codes (MACs) or checksums. A message encryption algorithm and a hash function are negotiated for encryption and integrity verification, respectively.

How SSL Works (continued)

A typical example is encrypting data using Data Encryption Standard (DES), a symmetric encryption scheme, and generating authentication codes for integrity verification using MD5. MD5 is a hashing algorithm intended for use on 32-bit machines to create digital signatures. MD5 is a one-way hash function, that is, it converts a message into a fixed string of digits that form a message digest.

4. The server requests a client certificate to authenticate the client. If there are multiple certificates, the user chooses which personal certificate to present. A secure channel is established, with the client generating a session key and using the server public key to encode the key to send it securely over the Internet.

SSL Handshake

At the beginning of their communication, the client and directory server perform a handshake, which includes three important tasks:

- The client and server decide which cipher suite to use. A cipher suite is a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, the two nodes negotiate to see which cipher suite they use when transmitting messages back and forth.
- The server sends its certificate to the client. The client verifies that the server's certificate was signed by a trusted CA. Similarly, if client authentication is required, the client sends its own certificate to the directory server. The directory server verifies that the client's certificate was signed by a trusted CA.
- The client and server exchange key material using public key cryptography, and they generate a session key from this material. All subsequent communications between the client and the server are encrypted and decrypted by using this set of session keys and the negotiated cipher suite.

Cipher Suite

The secure sockets layer (SSL) technology supports a variety of alternate encryption protocols for communication. Each choice is called a cipher suite, which specifies the following:

- The type of signature-capable certificate (RSA or DSS)
- The type of symmetric encryption that should be used (RC4, RC2, IDEA, DES, or 3DES)
- The type of signature algorithm (secure hash) that should be used (MD5 or SHA)

Cipher Suite	Authentication	Encryption	Data Integrity
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	RSA	DES40	SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RSA	RC4_40	MD5
SSL_RSA_WITH_NULL_SHA	RSA	None	SHA
SSL_RSA_WITH_NULL_MD5	RSA	None	MD5

OracleAS Certificate Authority

OracleAS Certificate Authority provides:

- **A ready-to-use PKI solution**
- **Easy provisioning of X.509 version 3 digital certificates**
- **Seamless integration with the OracleAS Single Sign-On server**
- **Three methods of authentication:**
 - **OracleAS Single Sign-On server authentication**
 - **Secure sockets layer (SSL) using existing certificates issued by the CA**
 - **Traditional administrative review/approval**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Certificate Authority

You can get certificates from certificate authorities, such as VeriSign and Thawte. Oracle Application Server also has a certificate authority, that is, OracleAS Certificate Authority. You can use this to set up your own certificate authority.

OracleAS Certificate Authority completes the Oracle PKI solution by providing a certificate authority and registration authority combined with an easy-to-use, comprehensive Web interface. OracleAS Certificate Authority is highly integrated with Oracle products. Using the OracleAS Single Sign-On server, users can easily navigate to OracleAS Certificate Authority and request a certificate for future PKI-based OracleAS Single Sign-On authentication. The integration of OracleAS Certificate Authority with Oracle Internet Directory, OracleAS Single Sign-On server, and other Oracle products solves one of the deployment problems associated with PKI.

Authentication Methods

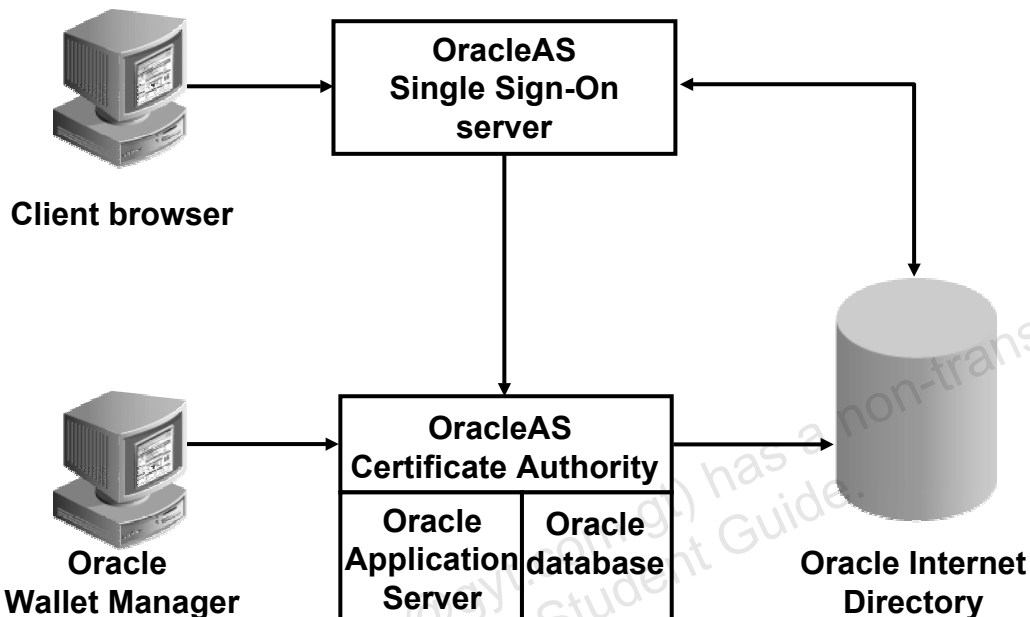
- Application users authenticating to the OracleAS Single Sign-On server can seamlessly obtain a certificate without having any technical education or understanding of PKI. The application can then use the newly issued certificates to transparently authenticate the application user, providing increased security.

OracleAS Certificate Authority (continued)

- If a user has previously been issued an X.509 version 3 certificate, he or she can submit the same certificate as a means of authenticating to OracleAS Certificate Authority over HTTPS.
- In addition, OracleAS Certificate Authority can enforce a manual certificate approval process. This is useful if an organization's security policy dictates that requests for certificates must be approved manually.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Oracle Application Server Certificate Provisioning



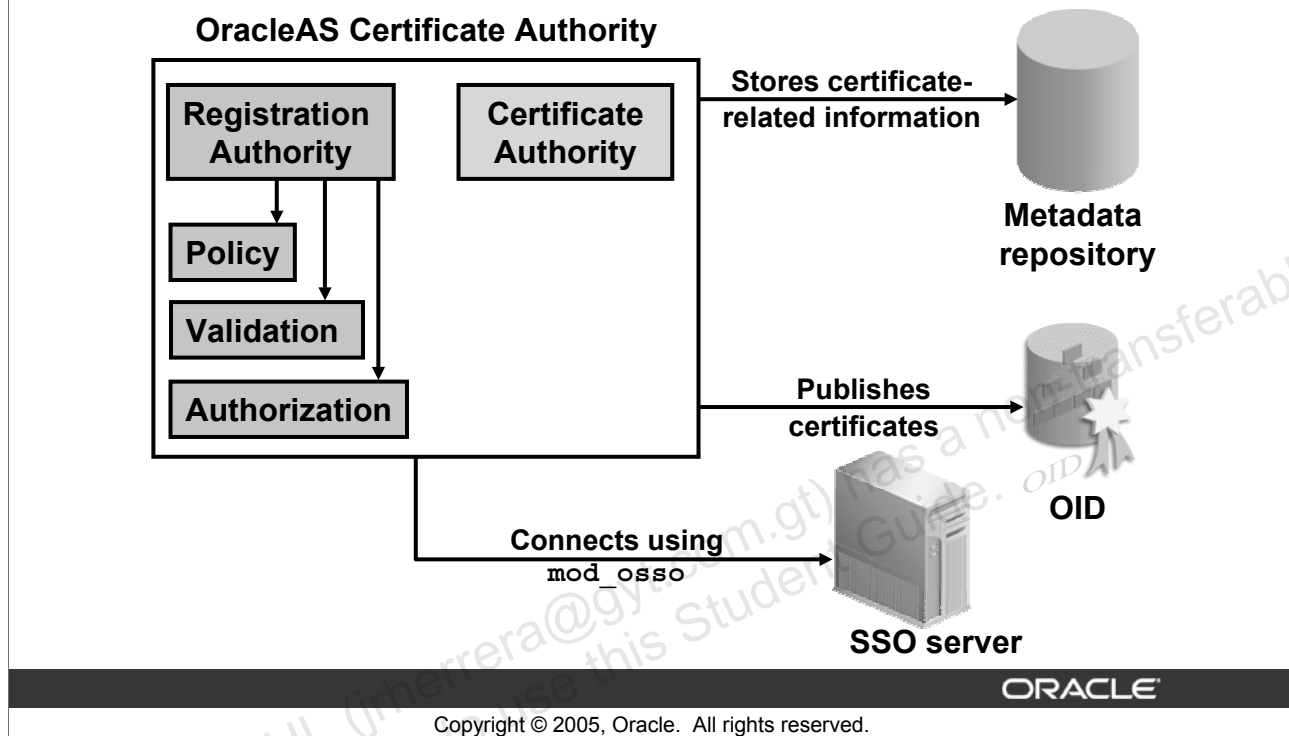
Copyright © 2005, Oracle. All rights reserved.

Oracle Application Server Certificate Provisioning

OracleAS Certificate Authority uses Oracle Internet Directory as the storage repository for certificates. This architecture provides centralized certificate management, simplifying the certificate provisioning and revocation tasks. After a user has been provisioned in Oracle Internet Directory and authenticated to the OracleAS Single Sign-On server, he or she can choose to request a digital certificate from OracleAS Certificate Authority. The certificate is automatically and immediately provisioned in Oracle Internet Directory. This method of certificate provisioning leverages OracleAS Single Sign-On to identify the user and populate required fields in the certificate request. Likewise, the OracleAS Certificate Authority administrator or certificate owner can revoke the certificate in real time, causing future attempts to use the certificate for OracleAS Single Sign-On authentication to fail.

OWM is the interface for all Oracle server-side components. It generates PKCS #10 certificate requests that can be submitted to OracleAS Certificate Authority or external CA services. OracleAS Certificate Authority is designed for customers who want an easy-to-deploy, one-stop Oracle PKI solution for their enterprises. Oracle's PKI is interoperable with leading PKI vendors, such as RSA, Baltimore, and Entrust.

OCA Functional Structure



Copyright © 2005, Oracle. All rights reserved.

OCA Functional Structure

After a user has an entry in Oracle Internet Directory, SSO can authenticate that user. The CA can issue certificates to the users authenticated by SSO. OCA must be linked to the SSO server, and relies on the `mod_osso` component of Oracle Application Server for that connection. OCA automatically publishes the certificates that it issues to Oracle Internet Directory. When a certificate is revoked, OCA deletes the certificate entry in Oracle Internet Directory.

OCA uses the Metadata Repository, which is part of OracleAS Infrastructure, as its internal repository for storing certificate requests, issued certificates, and related information.

OCA consists of the Registration Authority (RA) and the Certification Authority (CA). The three important modules of RA are Authorization, Validation, and Policy.

The Authorization module enforces the requirement that the user has privileges appropriate to making the request received by OCA. Any user authenticated by SSO and SSL automatically has the privilege to get more certificates or revoke existing certificates, but only for the same distinguished name (DN).

A user who is not authenticated automatically and must be approved manually has only the privilege to list or request certificates.

OCA Functional Structure (continued)

The policy restrictions can be created by the OCA administrator by editing the OCA configuration file with any editor. The packages within the Policy module enforce these restrictions.

The certificate authority stores certificate requests, as well as issued or revoked certificates, in the database.

Oracle Internet Directory publishes certificates that are issued and deletes those that are revoked.

The Oracle HTTP Server module `mod_ossso` is responsible to interact with the Single Sign-On server.

In Oracle Application Server, components such as OracleAS Web Cache and Oracle HTTP Server communicate with other components as well as external clients, such as browsers. To secure these communications, you can configure Oracle Application Server to use SSL. The slide depicts communication paths between Oracle Application Server components and the protocols that they use. Though Oracle HTTP Server communicates with OC4J using Apache JServ Protocol (AJP), and browsers use HTTP to communicate with OracleAS Web Cache, all these protocols can work with SSL.

None of the components in the Oracle Application Server installation are configured for SSL by default. Depending on the paths that you want to secure, you can configure SSL only for specific paths. For example, secure only OracleAS Web Cache and Oracle HTTP Server because these are accessible by the public.

OCA Single Sign-On Authentication



ORACLE

Copyright © 2005, Oracle. All rights reserved.

OCA Single Sign-On Authentication

OracleAS Certificate Authority supports authentication using OracleAS Single Sign-On server, using an existing X.509 version 3 certificate over SSL, and the traditional manual, administrative approval process.

After a user has been authenticated by using one of the authentication methods, he or she can be provisioned a certificate. Using the OracleAS Single Sign-On server over SSL results in immediate certificate provisioning. The traditional manual, administrative approval process requires administrator intervention to review and approve a certificate request.

Users can overcome the tremendous difficulties that are historically associated with PKI by using the OracleAS Single Sign-On server or SSL. The time delay and administrative overheads between requests and provisioning are eliminated.

OracleAS Certificate Authority does not assume any understanding of PKI by the end user. As a result, the amount of input required by the end user has been minimized.

OCA Configuration Elements

- **The OCA configuration file is located at**
`$ORACLE_HOME/oca/conf/oca.xml`.
- **OCA relies on:**
 - **Wallets to store various PKI credentials**
 - **A password store to hold various required passwords**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OCA Configuration Elements

The OCA configuration file is located at `$ORACLE_HOME/oca/conf/oca.xml`. The parameters in this configuration file are used by the command-line tool (`ocactl`) and by the policy administration modules. You can use any XML editor to edit it.

In addition to correct configuration specifications, OCA relies on wallets to store various PKI credentials and a password store to hold various required passwords.

OCA Wallets

OCA needs the following wallets and certificates to operate:

- **CA Signing Wallet:** Automatically created by the Installer. This wallet contains the signing key and signing certificate of OracleAS Certificate Authority. Such a wallet can also be imported from another Certificate Authority to set up a new and different hierarchical CA structure, replacing the existing one. This new wallet, signing key, and signing certificate are signed by the new CA from which you imported it.
- **CA SSL Wallet:** Also automatically created by the Installer. This wallet contains the SSL certificate and the private key of the SSL server hosting the OCA. Such a wallet can also be imported from another Certificate Authority or can be manipulated by OWM.

OCA Configuration Elements (continued)

Password Store

- Most of the passwords that OracleAS Certificate Authority uses are automatically generated and kept in a password store, which is encrypted using the OCA administrator password. All of the following passwords are kept in that store and can be changed by the OCA command-line tool (`ocactl`):
 - Several passwords needed to protect the interactions that OracleAS Certificate Authority has with various entities, such as for Oracle Internet Directory and Oracle database, and for SSL
 - Passwords protecting various sensitive data structures that OracleAS Certificate Authority uses, such as for the CA signing wallet and CA SSL wallet. The only password not automatically generated and kept in the password store is that of the CA administrator. This can be changed by the OCA `ocactl` command-line tool.
- Note:** In case the OCA administrator password is forgotten, you cannot recover it from any tool such as `ocactl`.

Starting and Stopping OCA

- **These operations can be performed only by using the `ocactl` command-line tool.**
 - **To start OCA, use the command:**
`$ORACLE_HOME/oca/bin/ocactl start`
 - **To stop OCA, use the command:**
`$ORACLE_HOME/oca/bin/ocactl stop`
 - **To obtain the status of OCA, use the command:**
`$ORACLE_HOME/oca/bin/ocactl status`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Starting and Stopping OCA

For security reasons, these operations can be performed only by using the `ocactl` command-line tool, which requires the OCA administrator's password.

Before OracleAS Certificate Authority can be started, the following components must be operational or available:

- Oracle HTTP Server (must be on the same machine as OCA)
- OC4J for OCA (must be on the same machine as OCA)
- Infrastructure metadata repository
- Oracle Internet Directory and, optionally, OracleAS Single Sign-On server (SSO)

In the default deployment, all these components are in the same Oracle home. For OracleAS Certificate Authority security reasons, you should not use Enterprise Manager for starting or stopping these components.

Accessing the OCA Interface

Use the following URL to access the OCA home page:

https://<your_server>:<ssl_port>/oca/admin



Accessing the OCA Interface

After OracleAS Infrastructure is installed successfully, the OracleAS Certificate Authority server becomes functional. To access the OracleAS Certificate Authority administration interface, launch your Web browser and enter the URL and port number of the administration server as they were displayed at the end of the OracleAS Infrastructure installation; for example, `https://<your_server>:<ssl_port>/oca/admin`.

The port information is available from the `$ORACLE_HOME/install/portlist.ini` file in the OracleAS Certificate Authority SSL port. The OracleAS Certificate Authority home page appears, displaying three additional subtabs.

If you are accessing the administration interface for the first time, then you must request a certificate to obtain authentication before you can perform the tasks on the Certificate Management tabbed page.

Details Required to Obtain a Certificate

Common name	Name that you want on the certificate
E-mail address	E-mail address of the OCA administrator
Organization unit	Name of the organization unit or division to which the OCA administrator belongs
Organization	Name of the company or organization to which the administrator belongs
Location	City location of the administrator
State/Province	State or province of the administrator
Country	Two-letter code for the country
Password	Password specified for the administrator

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Details Required to Obtain a Certificate

As stated earlier, to access one of the tabs, you need to obtain a certificate. In contrast to other systems where it is a complex task to request, acquire, and install your Web administrator PKI certificate, this process is simple and easy when using OracleAS Certificate Authority.

To request the Web administrator certificate for your authentication, you fill in the details and submit a brief form. You must be working at the computer that you intend to use as the Web administrator, and OracleAS Certificate Authority must be running. If not, you get an error message indicating that you have to start OracleAS Certificate Authority first.

When the Web administrator certificate is issued, you import it into your browser so that you can access the facilities of the OracleAS Certificate Authority administration interface.

This easy process—easy importation after filling in a simple request form—replaces all the operations otherwise required for PKI certificate acquisition and use before OCA.

Requesting the Web Administrator Certificate

Web Administrator Enrollment

DN Information [Advanced DN](#)

*Common Name

E-Mail Address

Organizational Unit

*Organization

City/Locality

State

Country

OracleAS Certificate Authority administrator password

*Password

Certificate Information

Certificate Key Size Select the size of the certificate key to generate. The

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Requesting the Web Administrator Certificate

To request for your certificate, perform the following steps:

1. Enter data in the fields on the Web Administrator Enrollment page to request your certificate. The Common Name, Organization, and Password fields are mandatory. Scroll down to display the entire fields on this page.
2. Depending on the browser you are using, you are asked which key size to use for encryption. You should use Microsoft Enhanced Crypto Provider for the Web Administrator Certificate. If smart cards (such as Gemplus) are available, they should be used. Provide a value for Validity period to determine how long (in number of days) the certificate should be valid.
3. Click Submit.
4. Follow the instructions of your browser as it generates the key pair.

After the key generation process finishes, click Import Certificate. Now you have a client authentication certificate in the common name you specified.

Managing Certificates

As an OCA administrator, you can perform the following actions on the certificates:

- **Search the master certificate list by name or number**
- **Examine the details of a specific certificate**
- **Approve or reject any individual certificate request**
- **Revoke certificates**

Home Certificate Management

Search Certificate Request on All Pending Requests value

Certificate Management

Use this form to approve certificate requests, renew or revoke certificates and update certificate revocation lists.

Request ID	User DN	Request Type	Request Date	Status	Serial Num
No requests retrieved.					

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing Certificates

You can perform specific tasks in managing certificates or the certificate authority from the OracleAS Certificate Authority Administration interface. From the Certificate Management tab, you can:

- Approve certificate requests
- Reject certificate requests
- Search and list issued certificates
- Search and list certificate requests
- Revoke certificates
- Update CRLs

OracleAS Certificate Authority maintains a master list of all certificate requests and their current statuses: pending, rejected, or certified. When you click the Certificate Management tab, all certificate requests that need action are displayed. The OCA Web administrator is responsible for approving or rejecting such requests, and for managing the Certificate Revocation List (CRL).

In performing these tasks as the OCA Web administrator, you can search the master certificate list by name or number, and then examine specific certificates of interest. You can then approve or reject any individual certificate request. You can also revoke specific certificates, if they have been compromised or are no longer appropriate, such as those owned by someone who has left the company.

Viewing, Approving, and Rejecting Certificates

The screenshot displays the Oracle Application Server Certificate Authority web interface. The main page is titled "Certificate Management" and includes a search bar with a dropdown menu set to "Certificate Request" and a "Go" button. A table lists certificate requests with columns for "Serial Number" and "User DN". A pop-up window titled "Certificate Request Details" is open, showing information for a specific request. The pop-up includes buttons for "Approve", "Reject", and "OK". The "Approve" button is circled in red. The "Certificate Request Details" window also contains a checkbox for "Apply policy check while approving a certificate request" and sections for "Contact Information" and "Certificate Request Information".

Select	Serial Number	User DN
<input type="radio"/>	4	CN=administrator,Email=adm

Certificate Request Details

Use this screen to approve or reject certificate requests.

☒ Apply policy check while approving a certificate request

Contact Information

Name : Heike Hundt
E-Mail ID: Heike.Hundt@oracle.com

Certificate Request Information

Status: Certificate Type: PENDING

Viewing, Approving, and Rejecting Certificates

Using the OracleAS Certificate Authority administration home page, you can display a specific certificate (issued or requested) or a list of all certificates.

To find a specific certificate or display a list of certificates, perform the following steps:

1. From the OracleAS Certificate Authority administration home page, click the Certificate Management tab to display the Search for Certificates form.
2. Use the Search drop-down list to find a specific issued or requested certificate:
 - a. Select Certificate to display issued certificates.
 - b. Select Certificate Request to display requested certificates.
3. Enter the appropriate value in the Search criteria field to search:
 - a. For All Pending Requests, no further specification is needed.
 - b. For ID/Serial, enter the serial number or the Request ID of the desired certificate or request.
 - c. For Common Name, enter the desired common name.
4. Click Go.

Viewing, Approving, and Rejecting Certificates (continued)

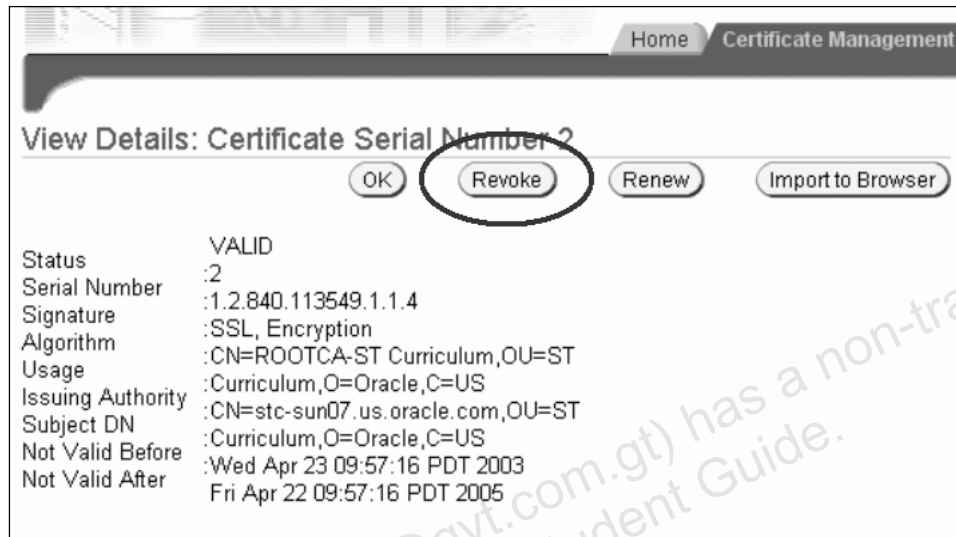
To view the details of a certificate, select the certificate that you want to review, and click View Details. The Certificate page appears, displaying the certificate's detailed contents.

Using this page, you can also revoke, renew, or import the selected certificate. To approve a certificate, perform the following steps:

- Use the contact information to authenticate the requestor of the certificate.
- Check the validity period and change it, if necessary.
- For subordinate CA certificate issuance, a default path length of 2 is displayed. A length of 2 means that OracleAS Certificate Authority can have two CAs below it in the chain. You can set a different path length, if required.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Revoking and Renewing Certificates



Revoking and Renewing Certificates

The OCA administrator can revoke certificates, and should do so if one of the following situations occurs:

- The owner of the certificate has changed status and no longer has the right to use the certificate.
- The private key of a certificate owner has been compromised.
- The root CA has ceased operations, or its private key has been compromised.

Being the OCA Web administrator, you can renew a user certificate 10 days before or after it expires, enabling it to continue to be used without interruption. (You can alter the number of days allowed before and after expiration.) Expired certificates can be renewed during the number of days specified for the period after the expiration date. After a certificate expires, and is not renewed during this permitted period, it becomes unusable and must be replaced by submitting a new certificate request and having it approved.

To renew a certificate, select the certificate (see the sections on listing and searching certificates), click View Details to display the Certificate page, and then click Renew. If the date is within the established window around the certificate's expiration date (default: 10 days before or after), then the certificate can be renewed. Otherwise, an error message appears regarding the established window.

Updating the Certificate Revocation List (CRL)

The screenshot shows the Oracle Application Server Certificate Authority interface. The title bar reads 'Oracle Application Server Certificate Authority'. Below it, there are tabs for 'Home' and 'Certificate Management'. The main heading is 'Update Certificate Revocation List'. The form contains two main fields: '*CRL Validity' with a text input box containing the number '5' and a label 'Enter the validity of the CRL in number of days.', and 'Signature Algorithm' with a dropdown menu currently set to 'SHA1 with RSA'. At the bottom right of the form are 'Cancel' and 'OK' buttons. A footer bar contains navigation links: 'Home', 'Certificate Management' (which is highlighted), 'Configuration Management', 'View Logs', 'Practice Statement', and 'Help'. Below the footer bar, it says 'Copyright (c) 2003, 2004, Oracle Corporation. All rights reserved.'

Updating the Certificate Revocation List (CRL)

Revoking a certificate makes it unusable in your environment. Making the fact of revocation publicly available ensures that revoked certificates are not misused. You do so by publishing the list of revoked certificates, called the certificate revocation list (CRL). All the applications in your trusted environment can use the CRL to prevent authentication by a revoked certificate. If an application is using OracleAS Certificate Authority and OracleAS Single Sign-On, then the user is prevented from authenticating immediately.

To generate an updated CRL, perform the following steps:

1. Access the OracleAS Certificate Authority administration home page, and click the Certificate Management tab.
2. Click the Update Certificate Revocation List (CRL) button. The Update Certificate Revocation List form appears.
3. In the CRL Validity field, specify a number, which represents the number of days until the next update.
4. For Signature Algorithm, choose from the drop-down list, such as MD5 with RSA or SHA1 with RSA.

Updating the Certificate Revocation List (CRL) (continued)

5. After entering data in the fields on this page, click OK. This action generates the certificate revocation list and stores it in the file system.

You can retrieve it for review or save by choosing Download CRL then Import to Browser or Download to your local disk.

Oracle HTTP Server uses this list to check the validity of the SSL certificates it receives, rejecting an SSL connection with any end entity whose certificate is on the CRL. If your system uses such multiple servers, you must copy the certificate revocation list file to the appropriate path and file names used by those servers as their CRLs. Follow the steps established for each server in setting up its CRL.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Configuring the OCA Server

You can configure the OCA server to:

- **Set notifications**
- **Manage policies**
- **Use logging and tracing**
- **Use the database and directory**

The screenshot shows the OCA administrator interface. At the top, there are tabs for 'Home', 'Certificate Management', and 'Configuration Management'. The 'Configuration Management' tab is active, and within it, the 'Notification' sub-tab is selected. The 'Notification' section contains a tip: 'Please note that the changes made to configuration parameters will take effect only when OracleAS Certificate Authority is restarted.' Below this, the 'Mail Details' section is visible, with a label 'Parameters to be set to enable email alerts or notification.' and a text input field for 'SMTP Server'.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring the OCA Server

You can use the OCA administrator interface to configure the OCA server. On the OCA administration page, all the configuration details are grouped and listed on the Configuration Management tabbed page. The Configuration Management tabbed page is further divided into three more tabs, each representing certain configuration details. The tabs are as follows:

- **Notification:** You can use this tab to set and configure different notification properties for the OCA server.
- **General:** You can use this tab to set and configure different general properties for the OCA server, such as logging and tracing, certificate publishing, and database and directory settings.
- **Policy:** You can use this tab to manage the policies of the OCA server.

Summary

In this lesson, you should have learned how to:

- **Discuss OracleAS PKI components**
- **Explain SSL and digital certificates**
- **Describe OracleAS Certificate Authority (OCA)**
- **Explain the OCA architecture**
- **Access OCA administration pages**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

18

Securing OracleAS Components by Using SSL

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

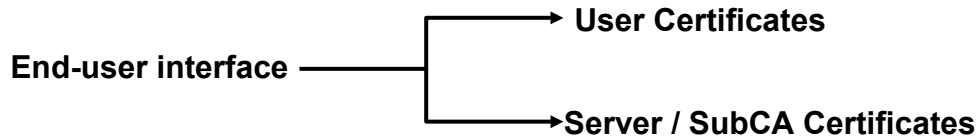
After completing this lesson, you should be able to do the following:

- **Access the OracleAS Certificate Authority (OCA) user pages**
- **Explain the Oracle Wallet Manager (OWM) functionality**
- **Manage user and trusted certificates**
- **Enable Oracle HTTP Server, Web Cache, and Portal to use SSL**

ORACLE

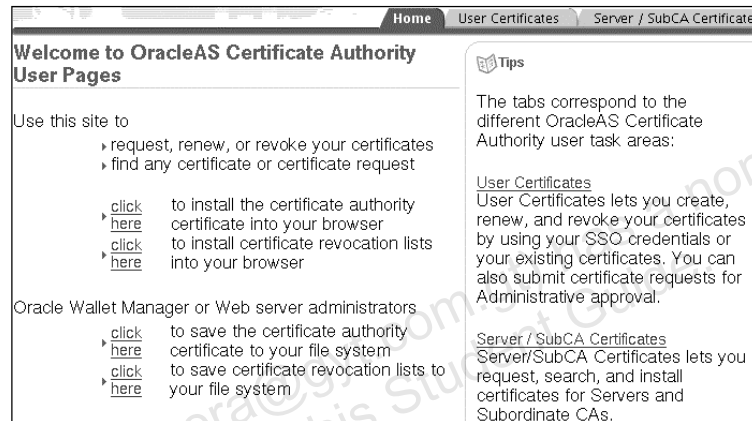
Copyright © 2005, Oracle. All rights reserved.

Accessing the End-User Interface



Enter the following URL:

`https://<your_server>:<ssl_port>/oca/user`



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Accessing the End-User Interface

The end-user interface of OracleAS Certificate Authority can be used by individuals, as well as server entities, to acquire certificates. To access the OracleAS Certificate Authority user home page, the user must perform the following:

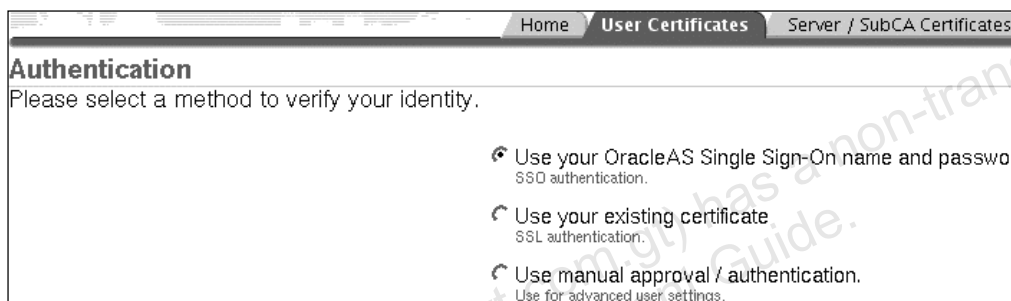
1. Launch a Web browser.
2. Enter the URL and port number of the OracleAS Certificate Authority server.

The OracleAS Certificate Authority user home page appears with the following tabs:

- User Certificates
- Server / SubCA Certificates

User Certificates

- You may need a user certificate to authenticate to an SSL-enabled Web site.
- The OCA user pages enable you to get a certificate.



User Certificates

To access any secure sockets layer (SSL)–enabled Web site (application) in your enterprise, you need a user certificate. This certificate authenticates you to the Web site. As a user of the enterprise Web site or applications, you can get this certificate by using the OracleAS Certificate Authority (OCA) user pages. These pages enable you to get a certificate in various ways. After getting the certificate, you can import it to your browser. After importing the certificate, whenever you access an SSL-enabled Web site or application, the browser presents the certificate for authentication.

When you click the User Certificates tab, the Authentication page is displayed. This page enables you to select how you intend to authenticate yourself to OracleAS Certificate Authority. The Authentication page enables you to select the type of authentication to be used for certificate management. The following authentication methods are available:

- OracleAS Single Sign-On, where the authentication is automated, based on the SSO password of the user
- Secure sockets layer (SSL), where the authentication is automated, based on the preissued SSL certificate of the user
- Manual, where the authentication is not automated. It requires the user to obtain the Certificate Request form, submit it, and then wait for approval from the OCA administrator.

Single Sign-On Authentication

The screenshot displays the 'User Certificates - SSO Authentication' web page. At the top, there are tabs for 'Home' and 'User Certificates'. Below the title, there is a 'Get Certificate' button. A table lists certificate details with columns: Select, Serial Number, User DN, Not Valid Before, Not Valid After, Status, Usage, and Revocation Reason. The table currently shows 'No certificates retrieved.' Below the table, a tip states: 'TIP To obtain a new certificate click "Get Certificate"'. There are 'Save CRL' and 'Change Authentication' buttons. A modal window titled 'User Certificate Information' is open, showing the 'User DN' field with the value 'cn=orcladmin,cn=users,dc=edrsr25p1,dc=com' and a description: 'This string is your fully-qualified Distinguished Name (DN) entry'. The 'Certificate Information' section includes a 'Certificate Key Size' dropdown set to '2048 (High Grade)' and a 'Certificate Usage' dropdown set to 'Authentication, Signing, Encryption'. The Oracle logo and copyright notice 'Copyright © 2005, Oracle. All rights reserved.' are at the bottom.

Single Sign-On Authentication

You can get a certificate automatically by identifying yourself to the OCA server by using your single sign-on (SSO) username and password. After you are authenticated to the OCA server, it generates your certificate and allows you to import it to the browser. This process saves the administration time and cost involved in provisioning a certificate. The following steps enable a user to acquire a certificate for authentication by supplying the required SSO information, such as username and password:

1. In the Authentication form, select the Use Your OracleAS Single Sign-On Name and Password option, and click Submit.
2. Enter the SSO username and password.
The User Certificates – SSO form appears. This form shows a list of valid certificates and enables the user to get a certificate, view details of a selected certificate, and renew or revoke a certificate.
3. On the User Certificates – SSO Authentication page, click Get Certificate to display the Certificate Request form.

Single Sign-On Authentication (continued)

4. The Certificate Request Form – SSO Authentication page appears. This page displays the distinguished name (DN) of your entry in the Oracle Internet Directory server. You cannot modify this value. Enter appropriate values for:
 - **Certificate Key Size:** Specify the size in bits of the key pair to be generated
 - **Certificate Usage:** Specify the purpose for which you are requesting the certificate.
5. After you submit the certificate request form, the OCA server generates your certificate and stores it in the Oracle Internet Directory server with your user DN. The Approved Certificate Information page displays the certificate information. You can import the certificate to your browser by clicking Import to Browser.

After you import the certificate to the browser, you can use it to authenticate to any SSL-enabled Web site or application in your enterprise.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Requesting Other User Certificates

The screenshot shows a web-based authentication interface. At the top, a section titled "Authentication" asks the user to select a method to verify their identity. Three options are listed: "Use your OracleAS Single Sign-On authentication." (selected), "Use your existing certificate SSL authentication.", and "Use manual approval / authentication." (Use for advanced user settings.). Below this, a "User Identification Request" dialog box is open. It contains a message: "This site has requested that you identify yourself with a certificate: edrsr25p1.us.oracle.com Organization: 'Oracle Corporation' Issued Under: 'Oracle Corporation'". Below the message is a dropdown menu labeled "Choose a certificate to present as identification:" with the selected option "users's Oracle Corporation ID [06]". To the right of the dialog box, there is a "DN Information" section with five input fields: "*Common Name", "E-Mail Address", "Organizational Unit", "*Organization", and "City/Locality".

Requesting Other User Certificates

SSL Authentication

You can set the authentication type to SSL when requesting for a user certificate. In SSL authentication, authentication is automated, based on a preissued SSL certificate. In this method of authentication too, as in SSO authentication, a certificate is provisioned to you immediately.

Manual Approval or Authentication

Enterprise applications are used by both employees and partners. Every employee in the enterprise has a unique identity created in the directory server. The employee uses this identity to connect to different applications in the enterprise. Employees can also request a certificate from the OCA server by using this SSO identity. The partners may or may not have identities created in the enterprise. If they do not have an identity, it becomes difficult for them to get a certificate by identifying themselves to the OCA server. In such cases, a user can use the manual approval method to request a certificate. As the name suggests, authentication is not automated. You must fill the Certificate Request form, submit it, and wait for approval from the administrator.

Managing User Certificates

- **After a user's certificate request is approved, the user can perform certain operations on the certificate.**
- **The operations that the user can perform are:**
 - **Review and retrieve a certificate**
 - **Renew a certificate**
 - **Revoke a certificate**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing User Certificates

Users use the OCA user pages to request for a certificate. After the request is approved, they can manage their certificates by using the same interface. The operations that a user can perform are:

- **Reviewing and retrieving a certificate:** You notify the user that his or her manual certificate request is approved. After this notification, the user can search for his or her certificate request by using the request ID assigned at the time of the request. The user can check the status of the certificate request, if the status displayed is Valid, and then click View Details. This displays the content of the approved certificate. The user must review the certificate details, and then click Import to Browser to import the certificate to the browser.
- **Renewing a certificate:** The users can use the OCA user pages to renew his or her certificate. To renew a certificate, the user must first search for his or her certificate, and then click the Renew button to renew the certificate. But this operation must be performed only within a certain period before or after the certificate expiry date—that is, within the renewal period.
- **Revoking a certificate:** To revoke a certificate, the user must first search for his or her certificate, and then click the Revoke button to revoke the certificate. Revoking a certificate marks it as revoked in OCA repositories, and it is added to the Certificate Revocation List (CRL) the next time the CRL is generated.

However, revoked certificates are not removed automatically from the user's browser database. The user should remove them manually.

Obtaining a Server Certificate

- You can make the access to a server in the enterprise secure by enabling SSL.
- To enable the server security, you require a PKCS#10 certificate request.
- Use OWM to generate the server request.
- You can get a server certificate from the OCA server or a trusted CA after submitting the PKCS#10 request.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Obtaining a Server Certificate

An administrator for any server can obtain a server certificate. This server certificate enables PKI authentication for that server with other servers or users. To enable the server security, you require a PKCS#10 certificate request. This request can be generated by using OWM or any other third-party tool, such as OpenSSL `reqtool`.

After generating a server certificate request by using OWM, you can request for a server certificate by using the OCA user pages.

Note: OWM is discussed in the next few slides.

What Is Oracle Wallet Manager?

- **Oracle Wallet Manager (OWM) is a stand-alone Java application that wallet owners use to manage and edit security credentials in their wallets.**
- **As a security administrator, you can use OWM to manage public-key security credentials on Oracle Application Server.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is Oracle Wallet Manager?

Oracle Wallet Manager (OWM) is a stand-alone Java application that wallet owners use to manage and edit security credentials in their wallets. It is used to manage user certificates and trusted certificates.

As a security administrator, you can use OWM to manage public-key security credentials on Oracle Application Server and other Oracle clients and servers. It can be used to request, store, and manage certificates.

OWM is used to create private keys and save them in the file system. These private keys are associated with X.509 certificates and require strong encryption. The 3-key Triple-DES is a substantially stronger encryption algorithm that is supported.

OWM Functions

- **Generating a public/private key pair**
- **Creating a certificate request**
- **Installing a certificate for the entity**
- **Configuring trusted certificates for the entity**
- **Creating a wallet that can be accessed by OWM**
- **Uploading a wallet to an LDAP directory, such as Oracle Internet Directory**
- **Downloading a wallet from an LDAP directory, such as Oracle Internet Directory**
- **Importing and exporting wallets**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OWM Functions

You can perform the following tasks by using OWM:

- Generate a public/private key pair and create a certificate request for submission to a CA.
- Install a certificate for the entity.
- Configure trusted certificates for the entity.
- Open a wallet to enable access to PKI-based services.
- Create a wallet that can be accessed by OWM.
- Upload a wallet to an LDAP directory, such as Oracle Internet Directory.
- Download a wallet from an LDAP directory, such as Oracle Internet Directory.
- Import wallets.
- Export wallets.

Creating a New Wallet

- You can create a new empty wallet by using the OWM tool.
- The password that you provide for the new wallet must:
 - Have at least eight characters
 - Contain alphabetic characters
 - Contain numbers or special characters

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Creating a New Wallet

You can create a new empty wallet by using the OWM tool.

The password that you provide for the new wallet must:

- Have at least eight characters
- Contain alphabetic characters
- Contain numbers or special characters

A wallet contains user credentials that can be used to authenticate the user to multiple databases or application servers. This makes it important to choose a strong wallet password. A malicious user who guesses the wallet password can access all the databases to which the wallet owner has access.

It is strongly recommended that users avoid choosing easy passwords based on their usernames or other personal information. A prudent security practice is that users change their passwords periodically, such as once per month.

Managing User Certificates

- **OWM uses two kinds of certificates:**
 - User certificates
 - Trusted certificates
- **You must install a trusted certificate from the CA before you can install a user certificate issued by that CA.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Managing User Certificates

OWM uses two kinds of certificates: user certificates and trusted certificates.

Trusted certificates are any certificates that you can trust, for example, a certificate issued by a CA. A wallet comes with some common trusted certificates. You also have the ability to add certificates.

User certificates are used by end entities, such as an end user, a database, a client, or a server. You must first install a trusted certificate from the CA before you can install a user certificate issued by that authority.

Managing user certificates involves the following tasks:

- Adding a certificate request
- Submitting the certificate request to a CA
- Importing the user certificate to a wallet
- Removing a user certificate from a wallet
- Removing a certificate request
- Exporting a user certificate
- Exporting a user certificate request

Adding a Certificate Request

- You must first create a certificate request to obtain a user certificate.
- You can add multiple certificate requests to a wallet.

Create Certificate Request

Please enter the following information to create an identity.

Common Name:

Organizational Unit:

Organization:

Locality/City:

State/Province:

Country: Key Size:

DN:

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Adding a Certificate Request

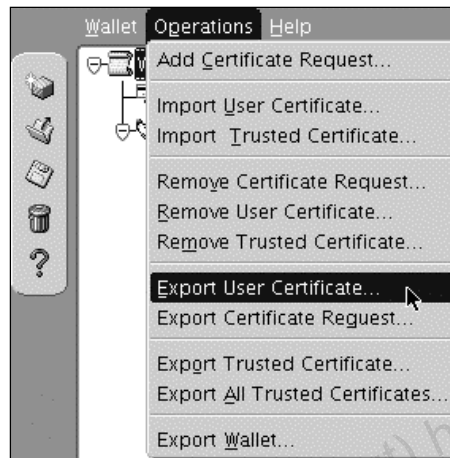
To obtain a user certificate, you must first create a certificate request. You can add multiple certificate requests to a wallet. When creating multiple certificate requests, OWM automatically populates each subsequent request dialog box with the content of the initial request, which you can then edit.

The actual certificate request becomes a part of the wallet. You can reuse any certificate request to obtain a new certificate. However, you cannot edit an existing certificate request.

To create a certificate request, perform the following steps:

1. Select Operations > Add Certificate Request.
2. The Create Certificate Request dialog box appears.
3. Enter the required information in the fields.
4. Click OK. An OWM dialog box appears confirming that a certificate request was successfully created. You can either copy a certificate request text from this dialog box and paste it into an e-mail message to send to a certificate authority, or you can export the certificate request to a file.
5. Click OK. You are returned to the OWM main window and the status of the certificate is changed to Requested.

Exporting a User Certificate Request



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Exporting a User Certificate Request

To save a certificate request in a file system directory, you must export the certificate request.

The advantages of exporting a user certificate request to the file system include the following:

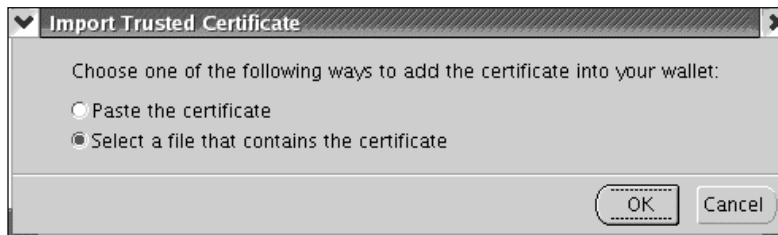
- Separates the act of generating a user certificate request and the act of going to a CA and requesting the certificate
- Enables users to create their private keys and export their certificate requests, which can then be sent to an administrator who may submit all requests to a CA collectively

To export a certificate request, perform the following steps:

1. Select Operations > Export Certificate Request from the menu bar. The Export Certificate Request dialog box appears.
2. Select the file system directory to which you want to save your certificate.
3. Enter the file name by which you want to save your certificate request.
4. Click OK.

A message at the bottom of the window confirms that the certificate request has been successfully exported to the file.

Importing the User Certificate to the Wallet



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Importing the User Certificate to the Wallet

After you have successfully generated a certificate request, you must send the request by e-mail to a certifying authority requesting for a certificate. You receive an e-mail notification from the certifying authority informing you that your certificate request has been fulfilled. Import the certificate in either of the following two ways (as shown in the slide):

- Copy and paste the certificate from the e-mail that you receive from the CA.
- Import the user certificate from a file.

If you have a certificate provisioned in the browser through other means, then, in order to use this certificate to authenticate to Oracle, you can use the PKCS#12 format of the certificate. After you have a PKCS#12 format, you can import it to your wallet.

Managing Trusted Certificates

Managing trusted certificates includes the following tasks:

- **Importing a trusted certificate**
- **Removing a trusted certificate**
- **Exporting a trusted certificate**
- **Exporting all trusted certificates**
- **Exporting a wallet**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

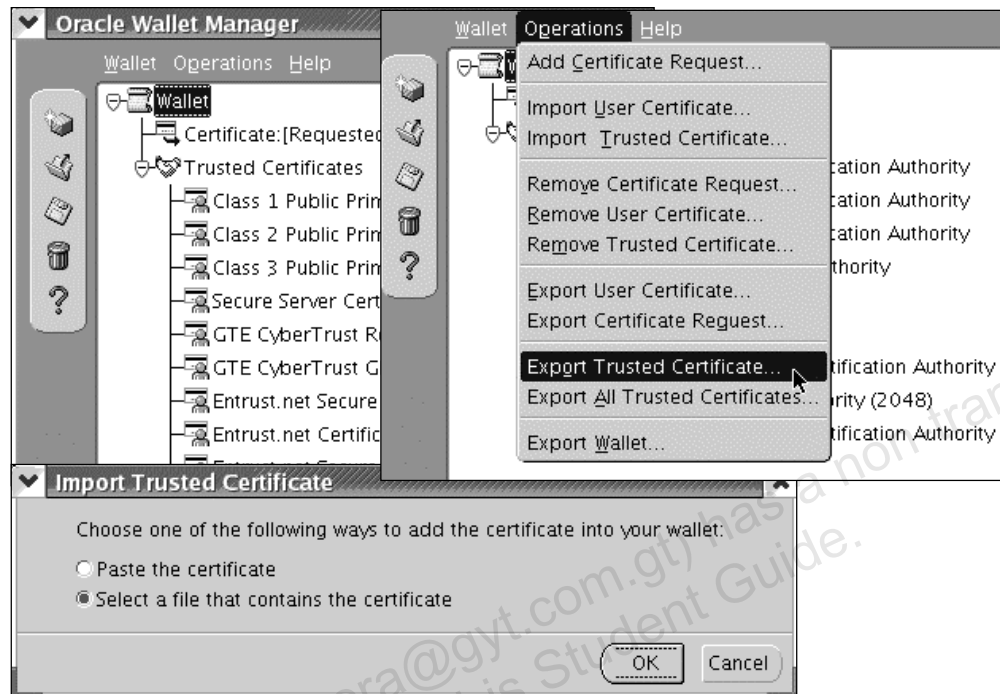
Managing Trusted Certificates

Authentication during run time involves verifying the certificate with the issuing CA. This process requires the root certificate of the CA (and its sub-CAs if required) to be imported to the wallet.

Managing trusted certificates includes the following tasks:

- Importing a trusted certificate
- Removing a trusted certificate
- Exporting a trusted certificate
- Exporting all trusted certificates
- Exporting a wallet

Importing/Exporting a Trusted Certificate



Copyright © 2005, Oracle. All rights reserved.

Importing/Exporting a Trusted Certificate

OWM automatically installs trusted certificates from VeriSign, RSA, Entrust, and GTE Cyber Trust when you create a new wallet. The graphic in the slide shows a list of trusted certificates that are installed when you create a new wallet.

You can import a trusted certificate into a wallet in either of the following two ways (as shown in the slide):

1. Paste the certificate from an e-mail that you receive from the certifying authority.
2. Import the trusted certificate from a file.

You must export a trusted certificate to save it to another file system location. To export a trusted certificate, perform the following steps:

1. Select Operations > Export Trusted Certificate. The Export Trusted Certificate dialog box appears.
2. Select a file system directory to which you want to save your trusted certificate.
3. Enter the file name by which to save your trusted certificate.
4. Click Save.

To export all your trusted certificates to another file system location, select Operations > Export All Trusted Certificate. The Export Trusted Certificate dialog box appears. The rest of the steps are similar to exporting a trusted certificate.

Exporting a Wallet

You can export a wallet to text-based PKI formats.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Exporting a Wallet

You can export a wallet to text-based PKI formats. Individual components are formatted according to the following standards:

Component	Encoding Standard
Certificate chains	X509v3
Trusted certificates	X509v3
Private keys	PKCS #8

To export a wallet, perform the following steps:

1. Select Operations > Export Wallet. The Export Wallet dialog box appears.
2. Select a file system directory to which you want to save your wallet.
3. Enter the file name by which to save your wallet.
4. Click Save.

Uploading Wallets

- **To upload a wallet to an LDAP directory, OWM uses:**
 - **SSL, if the specified wallet contains an SSL certificate**
 - **The directory password**
- **OWM does not permit executing the upload option unless the target wallet is currently open and has at least one user certificate.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Uploading Wallets

To upload a wallet to an LDAP directory, OWM uses SSL if the specified wallet contains an SSL certificate. Otherwise, it lets you enter the directory password.

To prevent accidental destruction of your wallet, OWM does not permit you to execute the upload option unless the target wallet is currently open and contains at least one user certificate.

Uploading Wallets (continued)

To upload a wallet, perform the following steps:

- Choose Wallet > Upload Into The Directory Service. If the currently open wallet has not been saved, a dialog box appears with the following message: Wallet needs to be saved before uploading.
- Choose Yes to proceed. Wallet certificates are checked for SSL key usage. Depending on whether a certificate with SSL key usage is found in the wallet, one of the following results occurs:
 - **If at least one certificate has SSL key usage:** When prompted, enter the LDAP directory server host name and port information, and then click OK. OWM attempts a connection to the LDAP directory server by using SSL. A message appears indicating whether the wallet was uploaded successfully or whether it failed.
 - **If no certificates have SSL key usage:** When prompted, enter the user's distinguished name (DN), the LDAP server host name, and port information. Click OK. OWM attempts a connection to the LDAP directory server using simple password-authentication mode, assuming that the wallet password is the same as the directory password.

Downloading Wallets

- **When a wallet is downloaded from an LDAP directory, it is resident in working memory.**
- **A downloaded wallet needs to be explicitly saved using any of the available save options:**
 - **Save:** Saves changes to the current open wallet
 - **Save As:** Saves open wallets to a new location
 - **Save in System Default:** Saves wallets in the default directory location

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Downloading Wallets

To download a wallet from an LDAP directory:

- Select Wallet > Download From The Directory Service.
- A dialog box prompts you for the user's distinguished name (DN), and the LDAP directory password, host name, and port information. OWM uses simple password authentication to connect to the LDAP directory. Depending on whether the downloading operation succeeds or not, you must perform one of the following:
 - **If the download operation fails:** Check whether you have correctly entered the user's DN, and the LDAP server host name and port information.
 - **If the download is successful:** Click OK to open the downloaded wallet. OWM attempts to open that wallet using the directory password. If the operation fails after using the directory password, then a dialog box prompts you for the wallet password.

If OWM cannot open the target wallet by using the wallet password, then check whether you have entered the correct password. Otherwise, a message is displayed at the bottom of the window indicating that the wallet has been downloaded successfully.

Requesting a Server Certificate

Home User Certificates **Server / SubCA Certificates**

Search on value [Advance Search](#)

Server / SubCA Certificates

Use this form to search certificate, certificate requests or to request a Server / SubCA certificate.

Select	Serial Number	User DN	Not Valid Before	Not Valid After	Status	Usage	Revocation Reason
No certificates retrieved.							

Requesting a Server Certificate

After generating a server certificate request by using Oracle Waller Manager, you can request for a server certificate by using the OCA user pages.

To request for a server certificate, perform the following steps:

1. Open the OCA user pages, and click the Server / SubCA Certificates tab.
2. The Server / SubCA Certificate tabbed page appears. Click Request a Certificate.
3. The Server / SubCA Certificate Request form appears. In this form, enter the following details:
 - **PKCS#10 Request:** Paste the PKCS#10 request that you generated by using OWM.
 - **Name:** Specify your name.
 - **E-mail or Phone Number:** Specify your phone number or e-mail address.
 - **Additional Comments:** Specify if you have any special instructions for the OCA administrator.
 - **Certificate Usage:** Specify the purpose for which the certificate is used.
 - **Validity Period:** Specify the duration for which you require the certificate.
4. Click Submit to request for the server certificate.

You can use this certificate only after it is approved by the OCA administrator.

Requesting a Subordinate CA Certificate

Certificate Request	
*PKCS#10 Request	<pre>MIIBbzCB2QIBADAwwMQswCQYDVQQGEwJVUzEPMA0GA1UEChQGb3Jl ZXIwMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCkRrNy/scT71C dNVRAv061OPosCce7v7TOyJzoDsRRrCP8P0UUR7gaN+HPWkR9xa6nqPhv LiF4bcs3bJePKBe8TpzdPDq6tnAHL0zepGM/Kka/yGGEWt6Og4HPAB+BL KoZlhvcNAQEEBQADgYEAYUijVdiho5GMI77scpwkIWwa62gH1I6IQJ/ArufZx cxaRDbUwKndjCCqPssiMNCItZQzI5/+QD+gCoT4+ALJ8Rx5E9bVR2+ecU0f +Z1QlqjhrSuS8KrnzhFRzKpwASdnmN9znZimN18=</pre> <p>Please paste the PKCS#10 certificate request into this text area.</p>
Contact Information	
*Name	<input type="text" value="server1"/>
E-Mail ID	<input type="text" value="server1@oracle.com"/> <small>E-Mail ID or Phone number is required.</small>
Phone Number	<input type="text"/>
Additional Comments	<div><div></div><div><small>If you have any comments for the administrator, please mention them here.</small></div></div>

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Requesting a Subordinate CA Certificate

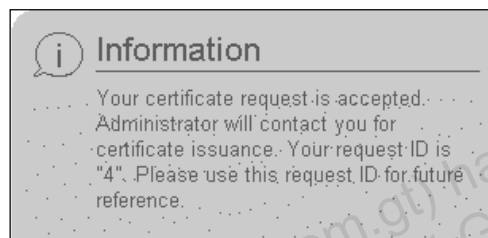
Enterprises with thousands of employees and several partners prefer to manage certificate issuance by using multiple CAs. This creates a hierarchical PKI structure with the root CA at the top and subordinate CAs below. A subordinate CA may, in turn, issue certificates to even lower-level CAs, creating what is called a certificate chain. An individual certificate signed by one of the subordinate CAs must present the certificates of all CAs up to the root. Because each authority's certificate is signed by a higher CA, a user can verify the validity of a particular certificate by tracing the certificate authority path back to the root CA.

You can obtain a subordinate CA certificate by using the OCA user pages. Perform the following steps to request a subordinate CA certificate:

1. Open OCA user pages, and click the Server / SubCA Certificates tab.
2. The Server / SubCA Certificate tabbed page appears. Click Request a Certificate.

Requesting a Subordinate CA Certificate

Certificate Information	
Certificate Usage	CA Signing
Validity Period	1 year



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Requesting a Subordinate CA Certificate (continued)

3. The Server / SubCA Certificate Request form appears. In this form, enter the following details:
 - **PKCS#10 Request:** Paste the PKCS#10 request that you generated by using OWM.
 - **Name:** Specify your name.
 - **Email or Phone Number:** Specify your phone number or e-mail address.
 - **Additional Comments:** Specify if you have any special instructions for the OCA administrator.
 - **Certificate Usage:** Specify the purpose for which the certificate is used. For the CA certificate, select CA Signing.
 - **Validity Period:** Specify the duration for which you require the certificate.
4. Click Submit to request for the SubCA certificate.

Importing and Downloading a CRL

Certificate Revocation List	
Issuer	:CN=ocaadmin,O=oracle,C=US
Last Update Date:	Fri Jan 02 07:56:09 PST 2004
Next Update Date:	Sat Jan 03 07:56:09 PST 2004
Serial Number	Revocation Date
No revoked certificates retrieved.	
<input type="button" value="OK"/> <input type="button" value="Import CRL into Browser"/> <input type="button" value="Download CRL in Binary"/> <input type="button" value="Download CRL in BASE64 format"/>	

Importing and Downloading a CRL

You can import or download the latest CRL to enable your browser and other programs to detect a revoked or expired certificate. You must avoid the use of such certificates; otherwise your resources and applications can be used by inappropriate or unauthorized users or applications.

To import a CRL, perform the following steps:

1. Go to OCA user pages and click the User Certificate tab.
2. The Authentication page appears. Select an appropriate method to identify yourself to the OCA server. For example, select Using your existing certificate. Then, click Submit.
3. The User Certificates – SSL Authentication page appears displaying the user's certificate. On this page, click the Download CRL button to download the latest CRL.
4. The CRL page appears. This page displays the list of certificates revoked with their serial numbers and revocation dates. You can download the CRL by using one of the three buttons: Import CRL into Browser, Download CRL in Binary, and Download CRL in BASE64 format. The first option stores the CRL in your browser, whereas the other two options save the CRL in a directory in the operating system.

Configuring Browser to Trust OCA



Copyright © 2005, Oracle. All rights reserved.

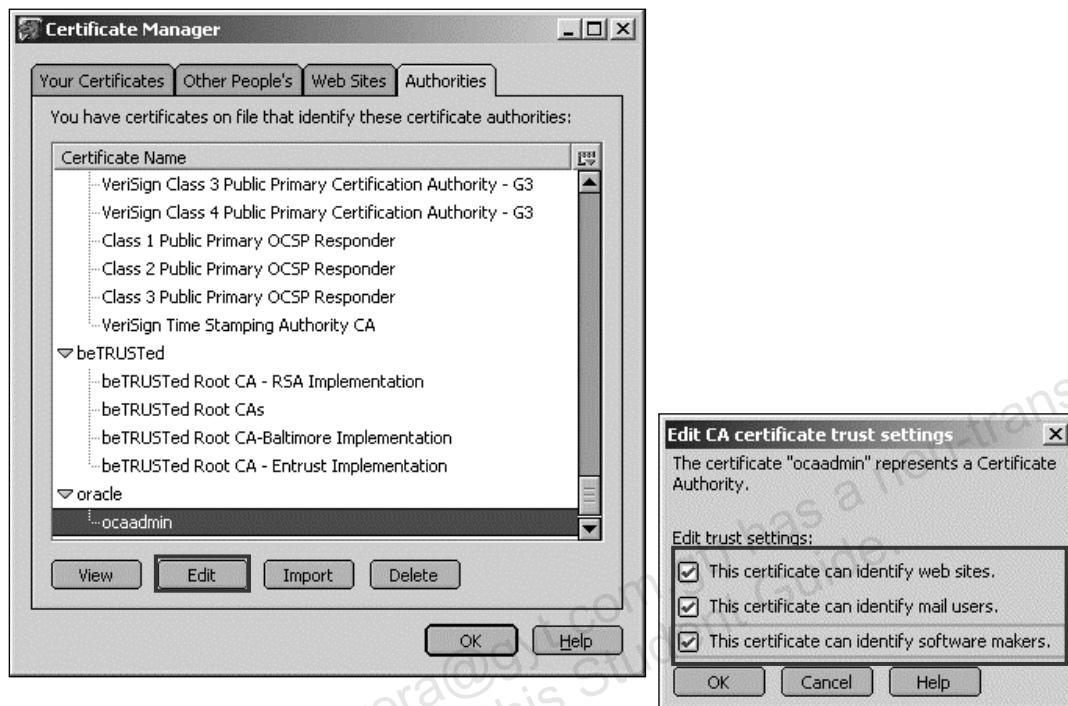
Configuring Browser to Trust OCA

When you import a certificate to your browser, it imports both the certificate that you requested and the certificate representing the CA that signed and issued your new CA certificate. Because OCA is not trusted by default, you must configure your browser for those activities for which you want to trust the CA. This configuration depends on the browser that you are using.

To trust a CA in Mozilla, perform the following steps:

1. From the Edit menu, select Preferences.
2. The preferences of the browser are displayed to be set. In the left pane, navigate to Privacy and Security, and expand it. From this menu option, select Certificates. In the right pane, the different options to manage your browser certificates are displayed.
3. Click the Manage Certificates button. This opens the Manage Certificates dialog box.

Configuring Browser to Trust OCA



Copyright © 2005, Oracle. All rights reserved.

Configuring Browser to Trust OCA (continued)

4. Click the Authorities tab. This displays the list of CAs from which you have received a certificate. Scroll down and find the OCA certificate. Select the certificate, and click Edit.
5. The “Edit CA certificate trust settings” dialog box appears. Select the options listed to trust the CA certificate.

Enabling Oracle HTTP Server to Use SSL

- One common use of SSL is to secure HTTP communication between a browser and a Web server.
- `mod_oss1` is Oracle's secure sockets layer (SSL) implementation.
- `mod_oss1` supports SSL v. 3.0.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Enabling Oracle HTTP Server to Use SSL

SSL is an encrypted communication protocol that is designed to securely send messages across the Internet. It resides between Oracle HTTP Server on the application layer and the TCP/IP layer, transparently handling encryption and decryption when a secure connection is made by a client.

`mod_oss1` supports SSL v. 3.0 and provides:

- Encrypted communication between client and server, using RSA or DES encryption standards
- Integrity checking of client/server communication using MD5 or SHA checksum algorithms
- Certificate management with wallets
- Authorization of clients with multiple access checks, exactly as performed in `mod_oss1`

SSL Configuration Tool

The SSL configuration tool:

- **Automates the manual steps for securing HTTP**
- **Must be installed for infrastructure first and then middle tier**
- **Creates a log file for verifying changes**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

SSL Configuration Tool

The SSL configuration tool is designed to automate the manual steps required for securing HTTP. If you design a topology where both an infrastructure and middle tier are present, you need to run the SSL configuration tool for the infrastructure first and then the middle tier. If you install Oracle Application Server and configure some changes, you need to run this tool and then refer to the SSL configuration tool log files to verify the changes. This tool creates log files in the directory from where the tool is run. In addition, a new log file is created each time the tool is run.

The SSLConfigTool executable is located in the \$ORACLE_HOME/bin directory.

SSL Configuration Tool (continued)

The syntax for the SSLConfigTool command is as follows:

```
SSLConfigTool ( -config_w_prompt  
                | -config_w_file <input_file_name>  
                | -config_w_default  
                | -rollback )  
[-dry_run]  
[-wc_for_infra]  
[-secure_admin]  
[-opwd <orcladmin_pwd>]  
[-ptl_dad <dad_name>]  
[-ptl_inv_pwd <ptl_inv_pwd>]
```

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Configuring Oracle HTTP Server for SSL Certificates

- You can configure Oracle HTTP Server for SSL by configuring the `http.conf` file.
- The `httpd.conf` file is located at `ORACLE_HOME/Apache/Apache/conf/httpd.conf`.
- You can enable SSL by adding valid parameters to the SSL Virtual Host Context in the `httpd.conf` file.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring Oracle HTTP Server for SSL Certificates

You can enable SSL for Oracle HTTP Server by including valid parameters in the `httpd.conf` file. The parameters that you can include are as follows:

- **ServerName:** Is the name of the server to be enabled for SSL
- **SSLEngine [on | off]:** Setting of the `SSLEngine` parameter to `on` enables the server for SSL.
- **SSLWallet File:** The location, or path, of the server wallet
- **SSLVerifyClient:** The verification type for client certificates. The options are as follows:
 - **None:** SSL without certificates
 - **Optional:** Server certificate only
 - **Require:** Server and client certificates

After the SSL is configured in the `httpd.conf` file, you can access Oracle HTTP Server with the HTTPS protocol by using the URL `https://host.domain:4443`. The default SSL port for Oracle HTTP Server is 4443. The server displays a certificate and it waits for your response to reject or accept the certificate.

Configuring Oracle HTTP Server for SSL Certificates (continued)

A sample of SSL Virtual Host Context in the `httpd.conf` file is as follows:

```
##      SSL Virtual Host Context
##
# file otherwise your virtual host will not respond to SSL
requests.
#
<VirtualHost _default_:443>
#   General setup for the virtual host
DocumentRoot
"/ade/lkethana_iasdemo/oracle/work/Apache/Apache/htdocs"
ServerName stc-sun07.us.oracle.com
ServerAdmin you@your.address
ErrorLog /private/oracle/work/Apache/Apache/logs/error_log
TransferLog /private/oracle/work/Apache/Apache/logs/access_log

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   Server Wallet:
#   The server wallet contains the server's certificate, private
key
#   and trusted certificates. Set SSLWallet at the wallet
directory
#   using the syntax: file:<path-to-wallet-directory>

SSLWallet file:/private/ias/wallet

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#   to point to the certificate files. Use the provided
#   Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /private/oracle/Apache/Apache/conf/ssl.crl
#SSLCARevocationFile /private/oracle/Apache/Apache/conf/ssl.crl/ca-
bundle.crl

#   Client Authentication (Type):
#   Client certificate verification type and depth. Types are
#   none, optional, require and optional_no_ca. Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
SSLVerifyClient optional
</VirtualHost>
```

Adding User Certificates to Oracle Internet Directory

- To enable a successful, certificate-based authentication, user certificates must be stored in the Oracle Internet Directory server.
- You can add the user certificate to the Oracle Internet Directory server by loading an `ldif` file.
- You can use the `ldapmodify` command to load the file.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Adding User Certificates to Oracle Internet Directory

After the SSO server is enabled to use SSL certificates, the user certificates must be stored in the Oracle Internet Directory server. You can upload user certificates to the Oracle Internet Directory server by using an `ldif` file. The `ldapmodify` command is used to upload the certificate to the Oracle Internet Directory server.

Syntax

```
ldapmodify -h ldaphost -p ldapport -D "cn=orcladmin" -w  
password -f file_name.ldif
```

In the sample `ldif` file shown below, the certificate of the user `ktaylor` is represented as an attribute of his entry in Oracle Internet Directory. The attribute type is `usercertificate`. The attribute value is the long string that follows.

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

The `ldif` file:

```
dn: cn=ken,o=oracle,dc=com  
changetype: modify  
replace: usercertificate
```

Oracle Application Server 10g R2: Administration I 18-34

Adding User Certificates to Oracle Internet Directory (continued)

```
usercertificate: :MIIC3TCCAkYCAgP3MA0GCSqGSIb3DQEBBAUAMIG8MQsw  
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcn5pYTEXMBUGA1UEBxMOUmVkd  
29vZCBTaG9yZXMxGzAZBgNVBAoTEk9yYWNsZSBDb3Jwb3JhdGlvbjEfMB0GA1  
UECxMWV2ViIFNpbmdsZSBTaWduLU9uLCBTVDEeMBwGA1UEAxMVQ2VydGlmaWN  
hYoEHmF4gomt c4mxSKh/zAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAkwXoCLDR  
qmK1Y9LQtIjLnCaIJKUZmS1Qj+bhu/IHeZLGHg4TJg3O2XVA5u/VxwjLeGBqL  
Xy2z7o3RujNKx2CVx6p/0Hk jnw4w6KVau2hcBgC9m4kzUGhHJ9b65v/zx7dIU  
KyJr4RF+lJhJg4/oYXxLrYHp5NAkHP4htT0gqCXiI=
```

Adding this user certificate to the Oracle Internet Directory server enables the SSO server to have a secure authentication on behalf of the partner applications.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Configuring OracleAS Web Cache to Use SSL

To configure HTTPS support for OracleAS Web Cache, perform the following tasks:

1. Create wallets.
2. Configure HTTPS ports and wallet location.
3. Request Client-Side Certificates (optional).
4. Permit only HTTPS requests for a URL or a set of URLs (optional).

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring OracleAS Web Cache to Use SSL

To provide more security for your Web site, you can configure OracleAS Web Cache to receive HTTPS protocol browser requests and send HTTPS requests to the origin server. HTTPS uses the secure sockets layer to encrypt and decrypt user page requests, as well as the pages that are returned by OracleAS Web Cache and origin servers.

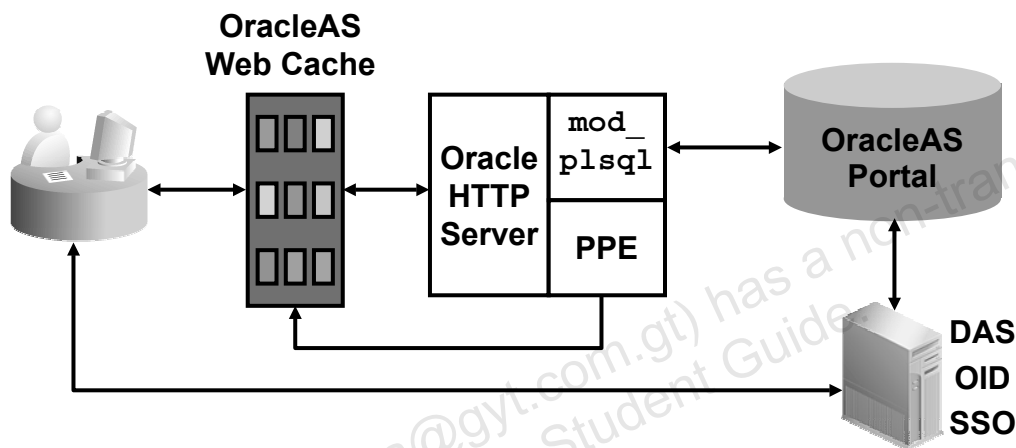
Typically, HTTP requests use port 80 and HTTPS requests use port 443. You can also configure OracleAS Web Cache to send traffic to the application Web server through an HTTP or HTTPS listening port.

To configure HTTPS support, perform the following tasks:

1. Create wallets.
2. Configure HTTPS ports and wallet location.
3. Request Client-Side Certificates (optional).
4. Permit only HTTPS requests for a URL or a set of URLs (optional).

Securing OracleAS Portal

Each Oracle Application Server component that communicates with OracleAS Portal must support HTTPS.



Copyright © 2005, Oracle. All rights reserved.

Securing OracleAS Portal

OracleAS Portal communicates with a number of components, each of which may act as a client or a server in the HTTP communication. As a result, to secure the user interaction with OracleAS Portal, each of these components may be configured individually to support HTTPS. After the components are secured, you need to perform the following tasks to secure your portal:

1. Secure the Parallel Page Engine (PPE), which is a multithreaded Java servlet that runs in the middle tier and assembles portal pages on the user's request. The PPE is called from Oracle HTTP Server and loops back to OracleAS Web Cache. Both connections must be secured.
2. Add OracleAS Portal to OracleAS SSO as a secured partner application. In this task, you reassociate the portal instance with the SSO server by running Oracle Portal Configuration Assistant (OPCA).
3. Secure calls from OracleAS Portal to Delegated Administration Services. This task ensures that the calls to the Infrastructure tier for user and group information from Oracle Internet Directory are secure.

Securing the Parallel Page Engine

Specify HTTPS ports in the `web.xml` file associated with the `OC4J_Portal` instance on the middle tier:

```
<servlet>
  <servlet-name>page</servlet-name>
  . . .
  <init-param>
    <param-name>httpsports</param-name>
    <param-value>4443</param-value>
  </init-param>
</servlet>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Securing the Parallel Page Engine

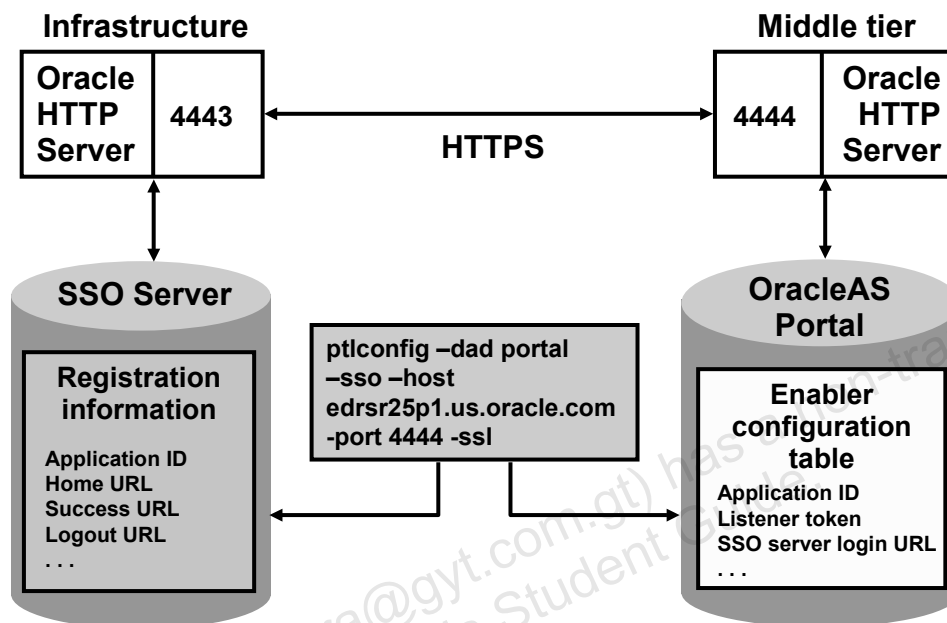
The PPE uses the protocol assigned to it by Apache (via `mod_oc4j`). Therefore, if the site (Web Cache) is using a protocol that is different from the one at the Apache level, then the PPE must be instructed to use the protocol of the site for generated URLs and loopback requests. If OracleAS Web Cache is implemented with HTTPS, connections to the PPE are automatically secured by SSL. Because the PPE creates URLs that perform loopbacks to OracleAS Web Cache, it is necessary to specify which ports use HTTPS:

1. Open the `web.xml` file for the `OC4J_Portal` instance on the middle tier. The file is located in the following directory:
`$ORACLE_HOME/j2ee/OC4J_Portal/applications/portal/portal/WEB-INF`
2. Add an additional `<init-param>` block to the file to indicate the ports that are using HTTPS. This should point to the OracleAS Web Cache HTTPS listening port. If multiple ports implement SSL-based connections, indicate the separation of the ports with a colon (:) in the block. For example:

```
<init-param>
<param-name>httpsports</param-name>
<param-value>4443:4445</param-value>
</init-param>
```

Oracle Application Server 10g R2: Administration I 18-38

Associating OracleAS Portal with OracleAS SSO in SSL Mode



Copyright © 2005, Oracle. All rights reserved.

Associating OracleAS Portal with OracleAS SSO in SSL Mode

Because the SSO configuration parameters have been changed to support HTTPS, you must reassociate the portal instance to the SSO for using SSL. OracleAS Portal is a partner application for OracleAS SSO. The SSO server stores registration information about partner applications in its repository. The information includes the partner application's ID, home URL, success URL, logout URL, and other parameters.

OracleAS Portal stores information about the SSO server in the form of enabler configuration information. It stores data (such as the application ID), listener token that consists of the host name and port used in a URL for the current request, encryption key to encrypt the login cookie, the SSO sever login URL, and some other data.

To associate the portal instance and the SSO server, you need to invoke `ptlconfig` on the OracleAS Portal middle tier.

Associating OracleAS Portal with OracleAS SSO

Run the `ptlconfig` script with the `-sso`, `-host`, and `-port` properties.

```
ptlconfig -dad portal -sso -host edrsr25p1.us.oracle.com -port 4444 -ssl
-dad: Portal DAD name
-sso: Creates partner application entries in OracleAS Single Sign-On
-host: Name of the host that you want to register as a partner application with OracleAS Single Sign-On
-port: Port used for registration
-ssl: Indicates that the port is https
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Associating OracleAS Portal with OracleAS SSO

In the example shown in the slide, the `ptlconfig` script associates the portal to the secured SSO server. The `ptlconfig` script is located in the `$ORACLE_HOME/portal/conf` directory on the middle-tier machine. The following parameters should be defined to run the script:

- `-dad`: Is the Portal DAD name
- `-sso`: Creates partner application entries in OracleAS Single Sign-On. When run without any additional parameters, partner application details are updated using the details from `iasconfig.xml`.
- `-host`: Is the name of the host that you want to register as a partner application with OracleAS Single Sign-On
- `-port`: Is the port that is used for registration
- `-ssl`: Indicates that the port is HTTPS

After you run the OPCA successfully, you should be able to access the Welcome page of your portal using HTTPS—for example, `https://edrsr25p1.us.oracle.com:4444/pls/portal`.

Securing Calls to DAS from OracleAS Portal

1. Log in to Oracle Internet Directory and update the base URL to Delegated Administration Services in Oracle Internet Directory:
cn=OracleContext, cn=Products, cn=DAS, cn=OperationURLs, orclidasurlbase=https://infra.mycompany.com:4443/oiddas/
2. Log in to the portal as administrator and refresh portal cache for Oracle Internet Directory parameters:
Services portlet > General Settings > SSO/OID tab
3. Invalidate the content of Web Cache.
4. Test DAS by accessing it from the User and Group portlets.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Securing Calls to DAS from OracleAS Portal

OracleAS Portal sends calls to the OracleAS Infrastructure tier for user and group information from Oracle Internet Directory using Delegated Administration Service (DAS). To secure these calls, it is necessary to update the base DAS URL within Oracle Internet Directory to reflect the use of HTTPS as the protocol. This base DAS URL is then used by the portal environment for generating subsequent DAS URLs. After updating the base URL in Oracle Internet Directory, it is necessary to force a refresh of the local cache of the portal, which holds the Oracle Internet Directory parameters. To accomplish that, you need to log in to the portal as a portal administrator, open the Global Settings in the Services portlet, select the Refresh Cache for Oracle Internet Directory Parameters check box, and click Apply. The page should refresh with the appropriate DAS host information.

You can test the new secure connection to DAS from the User and Group portlets. Because the portlets may have been cached in Web Cache, you may also need to invalidate Web Cache by using the OracleAS Web Cache administration interface.

Summary

In this lesson, you should have learned how to:

- **Access the OCA user pages**
- **Explain the OWM functionality**
- **Manage user and trusted certificates**
- **Enable Oracle HTTP Server, Web Cache, and Portal to use SSL**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

19

Backing Up and Restoring Oracle Application Server

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Install the Oracle Application Server Backup and Recovery Tool**
- **Configure the Backup and Recovery Tool by using Oracle Application Server Control**
- **Configure Oracle Application Server for a full backup**
- **Perform a complete Oracle Application Server backup**
- **Perform an incremental backup**
- **Restore a middle tier from a backup**
- **Recover OracleAS Infrastructure from a backup**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Backup and Recovery Features

- **The OracleAS Backup and Recovery Tool**
- **Three parts to a complete backup strategy:**
 - Perform a complete Oracle Application Server environment backup.
 - Perform incremental online backups on an ongoing basis.
 - Perform a new, complete Oracle Application Server environment backup after a major change.
- **Two types of recovery procedures:**
 - Recovery for data loss, host failure, or media failure (critical)
 - Recovery for process crashes or system outages (noncritical)

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Backup and Recovery Features

Oracle Application Server 10g offers complete backup and recovery procedures for your Oracle Application Server environment, along with the OracleAS Backup and Recovery Tool.

The OracleAS Backup and Recovery Tool consists of PERL scripts and associated configuration files that you can use to back up and recover configuration files and Metadata Repository.

There are three parts to a complete backup strategy:

- Perform a complete Oracle Application Server environment backup.
- Perform incremental online backups on an ongoing basis.
- Perform a complete Oracle Application Server environment backup after a major change.

Backup and Recovery Features (continued)

There are two types of recovery procedures:

- **Recovery for data loss, host failure, or media failure (critical):** These procedures enable you to recover from critical failures that involve actual data loss. In all cases, these procedures involve making sure that your state is consistent across all installations. Depending on the type of loss, they can involve recovering any combination of the following types of files:
 - Oracle software files
 - Configuration files
 - Metadata Repository files
 - Oracle system files
- **Recovery for process crashes or system outages (noncritical):** These procedures involve restarting the processes that have stopped or failed. These procedures generally do not involve restoring data. However, restoring data would be necessary in cases where a process crashes during a write operation to a configuration file, corrupting that file.

Roadmap for Backup and Recovery

- **Learn about database backup and recovery.**
- **Install and configure the OracleAS Backup and Recovery Tool.**
- **Test the backup and recovery strategy.**
- **Implement the backup strategy.**
- **Recover as necessary.**

ORACLE

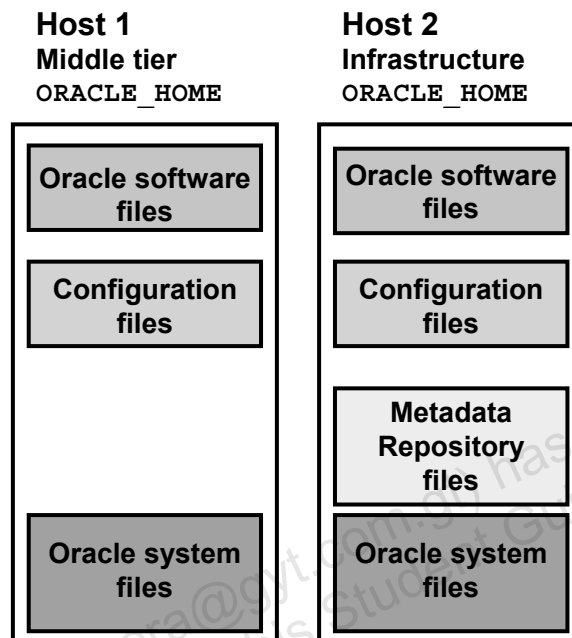
Copyright © 2005, Oracle. All rights reserved.

Roadmap for Backup and Recovery

This roadmap helps in getting started with backup and recovery of Oracle Application Server:

- **Learn about database backup and recovery:** Generally, OracleAS Infrastructure is an integral component of an Oracle Application Server environment. OracleAS Infrastructure has an Oracle database (Oracle Database 10g Release 10.1.0.3.1). The backup and recovery operations in an Oracle Application Server environment include performing backup and recovery of a database. Therefore, it is important for application server administrators to understand database backup and recovery.
- **Install and configure the OracleAS Backup and Recovery Tool:** You must install and configure the tool and familiarize yourself with its features. Even if you do not use the tool in the long run, it helps you get started with backup and recovery. You must install the tool in each of your infrastructure and middle-tier installations. This is because you will customize the tool for each installation.
- **Implement the backup strategy:** Following a standard backup strategy ensures that you are able to plan for recovery operations and avoid confusion and delay in recovery after a failure.
- **Perform recovery as necessary.**

Concept of Oracle Application Server Backup and Recovery



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Concept of Oracle Application Server Backup and Recovery

For the purposes of backup and recovery, you can divide your Oracle Application Server into different types of files:

- **Oracle software files:** These are static files such as binaries and libraries. They reside in the middle-tier and Infrastructure home directories. They are created during installation.
- **Configuration files:** These files contain configuration information and deployed applications. They reside in the middle-tier and Infrastructure home directories. They are created during installation and are updated during the normal operation of your application server. The configuration files are component specific files, such as `formsweb.cfg` and `default.env`.
- **Metadata Repository files:** These are the data files and control files that make up your Metadata Repository. They reside in the Infrastructure home directory. They are created during installation and are updated during the normal operation of your application server.

Concept of Oracle Application Server Backup and Recovery (continued)

- **Oracle system files:** These files include the `/var/opt/oracle` or `/etc` directory and the `oraInventory` directory. They exist on each host in your Oracle Application Server environment. They usually reside outside your Oracle Application Server installations, although the `oraInventory` directory may be in an installation home directory. They are created and updated by Oracle Universal Installer during installation and contain information about your installations.

Note: Your Oracle Application Server environment contains additional files to those mentioned in this section, such as log files; database configuration files, such as `tnsnames.ora`, `listener.ora`, `sqlnet.ora`, `orapwd`, and `spfile/pfile`; and files that you may deploy in the home directory, such as static HTML files and CGI scripts. You need to protect yourself from the loss of these files using your routine file system backup procedures.

A typical Oracle Application Server environment contains:

- An Infrastructure installation that contains Identity Management and Metadata Repository
- One or more middle-tier installations (J2EE and Web Cache, or Portal and Wireless) that use the Infrastructure

The installations in an Oracle Application Server environment are interdependent in that they contain configuration information, applications, and data that are kept in sync. For example, when you perform a configuration change, you may update configuration files in the middle-tier installation and Infrastructure; when you deploy an application, you may deploy it to all middle-tier installations; and when you perform an administrative change on a middle-tier installation, you may update data in Metadata Repository.

It is, therefore, important to consider your entire Oracle Application Server environment when performing backup and recovery. For example, you should not back up your middle-tier installation on Monday and your Infrastructure on Tuesday. If you lose files in your middle-tier installation, you restore it to Monday's state. However, your Infrastructure would be in its current state—out of sync with the middle tier. And, because you backed up the Infrastructure on Tuesday, you have no means of restoring it to a state in sync with Monday's middle-tier installation. You would not be able to restore your environment to a consistent state.

Instead, you should back up your entire Oracle Application Server environment at once. Then, if a loss occurs, you can restore your entire environment to a consistent state.

Note: For user-defined files or files that are not a part of regular backup (for example, `*.ora` files), there is a process to allow users to include them for backup using the OracleAS Backup and Recovery Tool. These files need to be entered in the `config_misc_files.inp` file for the tool to back up.

Terminology

- **A complete Oracle Application Server environment backup is performed:**
 - Just after an Oracle Application Server installation
 - Just after some major administrative change
- **A complete backup (hot or cold) is a full backup of a particular node (either the middle tier or Infrastructure).**
- **An incremental backup (hot or cold) is a backup that includes any changes since the last full or incremental backup.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Terminology

A **complete Oracle Application Server environment backup** is performed only after either of the following cases:

- Just after an Oracle Application Server installation
- Just after some major administrative change, for example, a new patch, operating system change, and topological change (the addition or removal of nodes from a farm)

A **complete backup** (hot or cold) is a full backup of a particular node (either the middle tier or Infrastructure). This can be performed while the system is up (hot) or down (cold).

For the middle tier, a complete backup includes a backup of all configuration files (bkp_restore.pl -m backup_config).

For the Infrastructure, a complete backup includes:

- Backup of all configuration files (bkp_restore.pl -m backup_config)
- Backup of all metadata (bkp_restore.pl -m backup_cold/backup_online)

An **incremental backup** (hot or cold) is a backup that includes any changes since the last full or incremental backup. This can be done while the system is up (hot) or down (cold).

For the middle tier, an incremental backup includes a backup of configuration files (bkp_restore.pl -m backup_config_incr).

Terminology (continued)

For Infrastructure, an incremental backup includes:

- Backup of configuration files (`bkp_restore.pl -m backup_config_incr`)
- Backup of metadata (`bkp_restore.pl -m backup_cold_incr -l <level>/backup_online_incr -l <level>`)

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Performing a Complete Oracle Application Server Environment Backup

A complete Oracle Application Server environment backup includes:

- A full backup of files in the middle-tier home directory
- A full backup of all files in the Infrastructure
- A complete cold backup of Metadata Repository
- A backup of the DCM file-based repository
- A full backup of the Oracle system files on each host

Host 1
Middle tier
ORACLE_HOME

	Oracle software files
	Configuration files
	Oracle system files

Host 2
Infrastructure
ORACLE_HOME

	Oracle software files
	Configuration files
	Metadata repository files
	Oracle system files

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Performing a Complete Oracle Application Server Environment Backup

A complete Oracle Application Server environment backup includes:

- A full backup of all files in the middle-tier home directory (including Oracle software files and configuration files)
- A full backup of all files in the Infrastructure home directory (including Oracle software files and configuration files)
- A complete cold backup of Metadata Repository
- A backup of the DCM file-based repository (if used)
- A full backup of the Oracle system files on each host in your environment

Performing Online Backups

Online backups include:

- An incremental backup of the configuration files in the middle-tier home directory
- An incremental backup of configuration files in the Infrastructure home directory
- A backup of the DCM file-based repository (if used)
- An online backup of Metadata Repository

Host 1
Middle tier
ORACLE_HOME

	Oracle software files
	Configuration files
	Oracle system files

Host 2
Infrastructure
ORACLE_HOME

	Oracle software files
	Configuration files
	Metadata repository files
	Oracle system files

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Performing Online Backups

Online backups should be performed on an ongoing basis. An online backup includes:

- An incremental backup of the configuration files in the middle-tier home directory
- An incremental backup of the configuration files in the Infrastructure home directory
- A backup of the DCM file-based repository (if used)
- An online backup of Metadata Repository

Note: When performing a restore, every incremental backup must also be restored.

Performing a Backup After a Major Change

- **Major changes:**
 - Operating system software upgrade
 - Oracle Application Server software upgrade or application patch
 - Restoration of Metadata Repository to a particular point in time
- **Backup procedure:**
 - Perform a new, complete Oracle Application Server environment backup.
 - Perform regular online backups.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Performing a Backup After a Major Change

If you make any of the following major changes to your host or the Oracle Application Server environment, perform a new complete Oracle Application Server environment backup, and then perform regular online backups:

- Operating system software upgrade
- Oracle Application Server software upgrade or application patch
- Restoration of Metadata Repository to a particular point in time, or a metadata restoration using a control file (using the `-u` or `-c` options with `"bkp_restore.pl -m restore_db"`). This type of restoration invalidates your previous backups, so you must perform a new, complete cold backup.

OracleAS Backup and Recovery Tool

- The OracleAS Backup and Recovery Tool is written in PERL.
- The tool is installed as part of the Oracle Application Server installation, and is located in the `Oracle_Home/backup_restore` directory.
- The tool can be used:
 - To perform backup and recovery operations
 - As a guidance in backup and recovery operations
- The tool assumes that OracleAS Metadata Repository is located in a single database.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Backup and Recovery Tool

The OracleAS Backup and Recovery Tool consists of PERL scripts and associated configuration files that you can use to perform backup and recovery procedures in the Oracle Application Server environment.

The tool can be used in different ways, depending on your level of experience and requirements:

- All users can refer to the tool for the list of Oracle Application Server configuration files that must be backed up.
- You can use the tool to automatically perform configuration file and Metadata Repository backup and recovery.
- You can refer to the tool for guidance when setting up your own backup and recovery.

This tool is installed as part of an Oracle Application Server installation.

Note: There are two areas where the OracleAS Backup and Recovery Tool cannot be used; instead, you need to use an existing backup and recovery strategy:

- When the Oracle Application Server metadata is loaded into an existing database using the RepCA utility
- If the DCM file-based repository is used

Preparing to Configure the Tool

1. Log in as the user that installed Oracle Application Server.
2. Create a script to set the ORACLE_HOME, ORACLE_SID, and PATH environment variables appropriately.
3. Create a directory under Oracle home.
4. Copy the OracleAS Backup and Recovery Tool files to the newly created directory, and extract the TAR file in the new directory.
5. Ensure that the bkp_restore.pl script has the execute permission.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Preparing to Configure the Tool

Before installing the OracleAS Backup and Recovery Tool, consider the following:

- You should install the tool for each Oracle Application Server installation in your environment so that you can customize the tool for each installation.
- You must install the tool in an empty directory on the same host as the corresponding Oracle Application Server installation.
- Make sure that the directory has read and write permissions for the user that installed Oracle Application Server.

The following steps assume that the OracleAS Backup and Recovery Tool is configured for an installation that has Oracle Application Server installed in \$HOME/infra, with the database SID of infra:

1. Log in as the user that installed Oracle Application Server.
2. Create a script to set the ORACLE_HOME and PATH environment variables.

```
prompt> cat "export ORACLE_HOME=$HOME/infra"
>>set_infr_env.sh
prompt> cat "export PATH=$PATH:$ORACLE_HOME/bin"
>>set_infr_env.sh
prompt> cat "export ORACLE_SID=infra" >>set_infr_env.sh
prompt> chmod u+x set_infr_env.sh
```

Oracle Application Server 10g R2: Administration I 19-14

Preparing to Configure the Tool (continued)

3. Create a directory for the OracleAS Backup and Recovery Tool in your \$ORACLE_HOME directory:
prompt> cd \$HOME/infra
prompt> mkdir backup_tool
4. Copy and extract the backup_restore.tar file:
prompt> cd /CD-ROM/utilities/backup/
prompt> cp * \$HOME/infra/backup_tool
prompt> cd \$HOME/infra/backup_tool
prompt> tar -xvf backup_restore.tar
5. Ensure that the bkp_restore.pl script has execute permission:
prompt> cd \$HOME/infra/backup_tool/backup_restore
prompt> chmod 755 bkp_restore.pl

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Configuring the OracleAS Backup and Recovery Tool

To configure the OracleAS Backup and Recovery Tool, perform the following steps:

1. Create backup and log directories.
2. Edit the `config.inp` file and provide the values for file variables depending on the type of installation.
3. Set the `ORACLE_HOME` environment variable.
4. Set the `ORACLE_SID` environment variable for the OracleAS Infrastructure installation.
5. Execute `./bkp_restore.sh -m configure`.

You are now ready to use the OracleAS Backup and Recovery Tool for this installation.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring the OracleAS Backup and Recovery Tool

The following steps describe the process to configure the OracleAS Backup and Recovery Tool:

1. Create directories to hold the backup and log files. To have better manageability, create these directories on separate disks:
 - **Log file directory (OracleAS Infrastructure and middle tier):** This directory holds log files that are created by the tool. The size of each log file depends on the number of components enabled for backup in your Oracle Application Server installation, and the number of applications that are deployed. Usually, the size is a few megabytes.
 - **Configuration backup directory (OracleAS Infrastructure and middle tier):** This directory holds the backup of configuration files. The size of the directory depends mainly on the components enabled, the number of applications deployed, the size of application archives (WAR, EAR files), and the frequency of the update to the applications. This directory should have several hundred megabytes of space.

Configuring the OracleAS Backup and Recovery Tool (continued)

- **Database backup directory (OracleAS Infrastructure only):** This directory holds backup of the data files and control files of OracleAS Infrastructure. This directory should have several gigabytes of space. To create the directories mentioned, enter:

```
prompt> mkdir -p /private/backups/log_files
prompt> mkdir -p /private/backups/config_files
prompt> mkdir -p /private/backups/db_files
```

2. The config.inp file is operating system dependent. Edit the config.inp file and enter values in the specific format of the operating system. The backup and recovery of configuration files depends on the setting of:

- config_files_list: Specifies the list of files to be backed up and recovered
- oracle_home: Specifies the base directory for the location of the files specified in config_files_list
- config_backup_path: Specifies the location where the configuration files should be backed up and is also used in restore_config mode to provide the user with a list of configurations that can be restored

For a middle-tier installation, these variables are required:

- oracle_home: Specifies the full path of \$ORACLE_HOME
- log_path: Defines the full path of the log file directory
- config_backup_path: Determines the full path of the configuration backup directory

For an OracleAS Infrastructure installation, set these variables in addition to the above:

- database_backup_path: Specifies the full path of the database backup directory

For example, the following is a list of variables set in config.inp of an OracleAS Infrastructure installation on Linux:

```
oracle_home=/home/oracle/infra
log_path=/home/oracle/infra/backup_tool/logbackup
config_backup_path=/home/oracle/infra/backup_tool/configbac
kup
database_backup_path=/home/oracle/infra/backup_tool/dbback
up
```

Note: As mentioned earlier, these files need to be entered in the config_misc_files.inp file for the tool to back up. When referring to these files, make sure that they reflect the operating environment.

3. Set the ORACLE_HOME environment variable to the home directory of the Oracle Application Server installation and include \$ORACLE_HOME/bin in the PATH variable:

```
export ORACLE_HOME=$HOME/infra
export PATH=$PATH:$ORACLE_HOME/bin
```

4. In an OracleAS Infrastructure installation, set the ORACLE_SID environment variable to Metadata Repository SID. Make sure that Metadata Repository is started.

```
export ORACLE_SID=infra
```

Configuring the OracleAS Backup and Recovery Tool (continued)

5. Execute the script to configure backup parameters as follows:

```
./bkp_restore.sh -m configure
```

This updates parameters in `config.inp` and, in the case of an Infrastructure, creates customized `.dat` files, which are used to perform backup, restore, and recovery on the database.

This configure option also creates a `backup_cold.dat` file, which contains the steps for a full cold backup of Metadata Repository. The following is a list of `backup_cold.dat`:

```
cat backup_cold.dat
shutdown immediate;
startup mount ;
configure controlfile autobackup on;
configure controlfile autobackup format for device type
disk to '/home/oracle/infra/backup_tool/dbbackup/%F';
run {
allocate channel dev1 device type disk format
'/home/oracle/infra/backup_tool/db
backup/%U';
backup database plus archivelog;
release channel dev1;
alter database open;
}
```

Using `bkp_restore.sh`

`bkp_restore.sh [-defmsv] filename`

- d: To print a trace without executing the command
- e: To specify an environment file
(default=config.inp)
- f: To force creation of files required by the command
- m: To specify the mode for running the script
- s: To run in silent mode
- v: To run in verbose mode

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using `bkp_restore.sh`

The following are some tips when running the `bkp_restore.sh` script:

- Run the `-configure` option once after installation, and again after you change the values in the `config.inp` environment configuration file.
- Use the `-d` option to print a trace without actually executing the command.
- Use the `-e` option to specify an environment file (default is `config.inp`).
- Use the `-f` option to force log file, database backup, and configuration file directories to be created if they are required by the current command and do not exist.
- Use the `-m` option to specify the mode for running the script.
- Use the `-s` option to run in silent mode.
- Use the `-v` option to run in verbose mode.
- Make sure that recovery is done using the same environment `config` file that was used during backup.

Using bkp_restore.sh: Examples

- **Cold backup:**
`./bkp_restore.sh [-dsv] -m backup_cold`
- **Incremental cold backup:**
`./bkp_restore.sh [-dsv] -m
backup_cold_incr -l incr_backup_level`
- **Online backup:**
`./bkp_restore.sh [-dsv] -m backup_online`
- **Incremental online backup:**
`./bkp_restore.sh [-dsv] -m
backup_online_incr -l incr_backup_level`
- **Restore database:**
`./bkp_restore.sh [-dsv] -m restore_db`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using bkp_restore.sh: Examples

The slide shows some usage examples.

Using bkp_restore.sh: Examples

- **Restore database along with control file:**
`./bkp_restore.sh [-dsv] -m restore_db -c`
- **Restore database to a specific point in time:**
`./bkp_restore.sh [-dsv] -m restore_db -u
07/04/2003_15:34:38`
- **Restore database without prompting for user input:**
`./bkp_restore.sh [-dsv] -m restore_db -n`
- **Backup configuration:**
`./bkp_restore.sh [-dsv] -m backup_config`
- **Backup configuration (incremental):**
`./bkp_restore.sh [-dsv] -m
backup_config_incr`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using bkp_restore.sh: Examples (continued)

The slide shows some usage examples.

Note: The `restore_db` option does not restore database control files. Unless you perform an explicit restore with “-c” or this is a point-in-time recovery, the control file will not get restored.

Using bkp_restore.sh: Examples

- **Restore configuration:**
`./bkp_restore.sh [-dsv] -m
restore_config -t timestamp_to_use`
- **List configuration files that changed since the last backup:**
`./bkp_restore.sh [-dsv] -m
list_changed_config`
- **Configure:**
`./bkp_restore.sh [-dsv] -m configure`
- **Configure the component configuration files:**
`./bkp_restore.sh [-dsv] -m
configure_nodb`
- **Configure and use the specified dbid:**
`./bkp_restore.sh [-dsv] -m configure -i
dbid_to_use`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Using bkp_restore.sh: Examples (continued)

The slide shows some usage examples.

- Configure and use the specified dbid instead of querying it from the database:
`./bkp_restore.sh [-dsv] -m configure -i dbid_to_use`

Note: For stand-alone installations, the “-I” option is not to be used. The “-i <user_dbid>” option is used only in cases where the metadata is being moved from one node to another, for example, in Disaster Recovery or Metadata Repository reassociation.

Using Oracle Application Server Control for Backup and Recovery

Farm > Application Server: portal.edrsr16p1

Home J2EE Applications Ports Infrastructure Backup/Recovery

Page Refreshed Sep 8, 2005 10:20:00 AM

General **CPU Usage** **Memory Usage**

Stop All Restart All

Status Up
Host edrsr16p1
Version 10.1.2.0.2
Installation Type Portal and Wireless
Oracle Home /home/oracle/portal
Farm infra.us.oracle.com

Application Server (2%)
Idle (90%)
Other (8%)

Application Server (44% 442MB)
Free (2% 18MB)
Other (54% 542MB)

Warning
The Backup & Recovery Tool is not configured. Click Configure Backup/Recovery Settings to proceed.

Application Server: portal.edrsr16p1

Home J2EE Applications Ports Infrastructure Backup/Recovery

What is Backup and Recovery?
The Backup/Recovery page provides a set of features and options that help you back up and protect your application server data and configuration files.
To back up your application server instance:
1) Click **Configure Backup/Recovery Settings** to specify a set of directories for your backup data and log files.
2) Click **Perform Backup** to select a type and mode of backup and then perform the backup.
3) Click **Perform Recovery** to recover your backup and restore your backed up configuration and data files.
For more information, see [Introduction to Backup and Recovery](#).

Home J2EE Applications Ports Infrastructure Backup/Recovery

Configuring the Backup and Recovery Tool by Using Oracle Application Server Control

You can use Application Server Control to configure the OracleAS Backup and Recovery Tool. By using this GUI tool, you can easily manage the backup and recovery configuration, depending upon the type of middle-tier installation performed on the host machine.

Starting the OracleAS Backup and Recovery Tool

You need to perform the following steps to start the OracleAS Backup and Recovery Tool:

- From the Application Server home page, click the Backup/Recovery tab.
- A warning screen displays that you need to configure the OracleAS Backup and Recovery Tool if it has not been configured previously.

Configuring Backup/Recovery Settings

Farm > Application Server: portal.edrsr16p1 >

Configure Backup/Recovery Settings

Before you perform a backup or recovery operation, you must first specify a directory for the generated log files and specify a directory for the backed up data. You must also make sure that the operating system user account that was used to install Oracle Application Server has write access to the backup directories. Cancel OK

Oracle Home /home/oracle/portal

* Log File Location /home/oracle/portal/backup_restore/logs
Enter the full directory path. Select a directory with several megabytes of available disk space. If the directory does not exist, Enterprise Manager can create it for you.

* Configuration Files Backup Location /home/oracle/portal/backup
Enter the full directory path. Select a directory with several hundred megabytes of available disk space. If the directory does not exist, Enterprise Manager can create it for you.

☒ **TIP** For the best protection against loss of data due to hardware failure, do not create your backup directories on the same disk where you installed the Oracle Application Server Oracle home. Instead, use a different disk, and if possible, a different disk controller. Cancel OK

Confirmation

The following directories do not exist. Do you want Enterprise Manager to create them?

Log File Location /home/oracle/portal/backup_restore/logs
Configuration Files Backup Location /home/oracle/portal/backup

No Yes

Confirmation

The Backup and Recovery configuration has been successfully updated.

Application Server: portal.edrsr16p1

Home J2EE Applications Ports Infrastructure Backup/Recovery

Configuring Backup/Recovery Settings

You need to specify the settings for configuring the backup and recovery on the host machine. Depending upon the type of installation, you can specify the following configuration:

- **Log File Location:** You need to specify the path where the log file for the backup and recovery needs to be stored.
- **Configuration Files Backup Location:** You must specify the path where the backup for the configuration files needs to be stored.

A confirmation screen appears if the specified directories do not exist. On clicking Yes, a final confirmation screen appears that highlights the successful updating of the Backup and Recovery configuration.

Backup Procedures

1. **Create a record of the Application Server configuration.**
2. **Perform a complete Application Server environment backup.**
3. **Perform online backups after every administrative change.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Backup Procedures

Before you perform your first backup, create a record of your environment. In the event you need to reconstruct your environment, you can refer to this record.

The first backup you perform should be a complete Oracle Application Server environment backup, which includes all the files in your environment. The procedure contains the following steps:

- Step 1. Shut down your Oracle Application Server environment.
- Step 2. Back up the Infrastructure.
- Step 3. Back up the middle-tier installations.
- Step 4. Back up the DCM file-based repository (if required).
- Step 5. Back up the Oracle system files.
- Step 6. Start your Oracle Application Server environment.

After every administrative change, or, if this is not possible, on a regular basis, perform an online backup of your Oracle Application Server environment. It contains the following steps:

- Step 1. Back up the Infrastructure.
- Step 2. Back up the middle-tier installations.
- Step 3. Back up the DCM file-based repository (if required).

Oracle Application Server 10g R2: Administration I 19-25

Creating a Record of the Configuration

Create a record for your Oracle Application Server configuration. The record must contain:

- **For each host:**
 - **Host name, virtual host name, domain name, IP address, hardware platform, and operating system information**
- **For each installation:**
 - **Installation type, host, owner name and number, group name and number, environment profile and type of shell, directory structure, mount points, full path for Oracle home, and port numbers used by the installation**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Creating a Record of the Configuration

You should maintain an up-to-date record of your Oracle Application Server environment in hard copy and in electronic form. You need this information in the event you must restore and recover your Oracle Application Server environment on a new disk or host. The electronic form should be stored on a host or e-mail system that is completely separate from your Oracle Application Server environment. Your Oracle Application Server hardware and software configuration record should include:

- The following information for each host in your environment:
 - Host name, virtual host name (if any), domain name, IP address, hardware platform, and operating system release level and patch information
- The following information for each Oracle Application Server installation in your environment:
 - Installation type (for example, Infrastructure, or J2EE and Web Cache), host on which the installation resides, username, user ID number, group name, group ID number, environment profile, and type of shell for the operating system user that owns the Oracle home (/etc/passwd and /etc/group entries), directory structure, mount points, and full path for Oracle home, and port numbers used by the installation.
 - For Metadata Repository (for example, the database version, patch level, base language, character set, global database name, and SID)

Oracle Application Server 10g R2: Administration I 19-26

Performing a Complete Backup

1. Enable `archive log` mode for your OracleAS Infrastructure.
2. Back up the OracleAS Infrastructure database.
3. Back up the OracleAS Infrastructure home and configuration files.
4. Back up the middle tiers.
5. Back up your Oracle system files.
6. Start your Oracle Application Server environment.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Performing a Complete Backup

To perform the first complete backup of your Oracle Application Server environment, you must perform the following steps.

- Shut down your Oracle Application Server environment:
 - Shut down your middle-tier instances:

```
prompt> cd $HOME/portal/opmn/bin
prompt> ./opmnctl stopall
```
 - Shut down your OracleAS Infrastructure instance:

```
prompt> cd $HOME/infra/opmn/bin
prompt> ./opmnctl stopall
prompt> export ORACLE_SID=infra
prompt> export ORACLE_HOME=$HOME/infra
prompt> export PATH=$PATH:$ORACLE_HOME/bin
prompt> sqlplus "sys/password as sysdba"
SQL> shutdown immediate
```

Step 1: Shut Down the Oracle Application Server Environment

- **Shut down your middle-tier instances:**
 - Stop the Application Server Control.
 - Stop components.
- **Stop the Infrastructure:**
 - Set the ORACLE_HOME, ORACLE_SID, and PATH environment variables.
 - Stop the Metadata Repository instance.
 - Stop the Net Listener.
 - Stop the Application Server Control.
 - Stop components.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Step 1: Shut Down the Oracle Application Server Environment

Shut Down Your Middle-Tier Instances

- Stop Application Server Control:
prompt> \$ORACLE_HOME/bin/emctl stop iasconsole
- Stop the components:
prompt> \$ORACLE_HOME/opmn/bin/opmnctl stopall

Stop the Infrastructure

If your Infrastructure contains a Metadata Repository, stop it as follows:

1. Set the ORACLE_HOME, ORACLE_SID, and PATH environment variables:
export ORACLE_HOME=\$HOME/infra
export ORACLE_SID=infra
export PATH=\$PATH:\$ORACLE_HOME/bin
2. Stop the Metadata Repository instance:
prompt> sqlplus "sys/password as sysdba"
SQL> shutdown immediate
SQL> exit
3. Stop the Net Listener:
prompt> \$ORACLE_HOME/bin/lsnrctl stop

Step 1: Shut Down the Oracle Application Server Environment (continued)

If your Infrastructure contains Identity Management, stop it as follows:

1. Stop Application Server Control:

```
prompt> $ORACLE_HOME/bin/emctl stop iasconsole
```

2. Stop the components:

```
prompt> $ORACLE_HOME/opmn/bin/opmnctl stopall
```

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Step 2: Back Up the Infrastructure

- **Enable `archive`log mode for your OracleAS Infrastructure database.**
- **Perform a cold database backup of Metadata Repository.**
- **Back up the Infrastructure home directory.**
- **Back up the Infrastructure configuration files.**
- **Verify that the configuration files have been created.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Step 2: Back Up the Infrastructure

Enable `archive`log Mode for Your OracleAS Infrastructure

To enable `archive`log mode for the database, perform the following steps:

(**Note:** This is done only once during the lifetime of a database.)

1. Set the `ORACLE_HOME` and `ORACLE_SID` environment variables.
2. Shut down the database:

```
SQL> connect sys/password as sysdba
SQL> shutdown immediate
```
3. Specify the destination directory for your archives by including the `LOG_ARCHIVE_DEST` initialization parameter in the initialization file. If `spfile` is being used, issue the following command:

```
alter system set log_archive_dest='xxx' scope=spfile;
```

If `pfile` is used, edit the following initialization file:

```
INFRA_ORACLE_HOME/dbs/initSID.ora
```

Change the `LOGARCHIVE_DEST` parameter to:

```
LOG_ARCHIVE_DEST=/disk1/oraHome/archive
```
4. Start the database in mount state (not open):

```
SQL> startup mount
```


Step 2: Back Up the Infrastructure (continued)

Enable archiveLog Mode for Your OracleAS Infrastructure (continued)

5. Enable archiveLog mode:

```
SQL> ALTER DATABASE ARCHIVELOG;
```
6. Verify archiveLog mode of the database:

```
SQL> archive log list;
```
7. Start the database for the changes to take effect:

```
prompt> lsnrctl start  
prompt> sqlplus "sys/<password>@<sid> as sysdba"  
SQL> startup  
SQL> exit
```

Perform a Cold Database Backup of Metadata Repository

1. Change to the directory that has the OracleAS Backup and Recovery Tool configured for OracleAS Infrastructure, and use the command:

```
./bkp_restore.sh -m backup_cold
```
2. List the directory specified under the database_backup_path variable of the config.inp file, and verify that the backup file has been created. The files have the timestamp of when they were created:

```
ls /home/oracle/infra/backup_tool/dbbackup
```
3. Copy the backup files to transportable media, such as tape, and store them separately.
4. Shut down the database:

```
SQL> connect sys/password as sysdba  
SQL> shutdown immediate;  
SQL> exit
```

Back Up the Infrastructure Home Directory

Perform a complete backup of all files in the Infrastructure by using your preferred operating system command, such as tar or cpio. Be sure to perform this backup as root because some of the files in the directory are owned by root. It is important to perform the backup so that file owners, groups, permissions, and timestamps are preserved. For example:

```
tar -cvf ihonfrmebak.tar $HOME/infra
```

Back Up the Infrastructure Configuration Files

You perform a configuration file backup immediately after backing up the entire directory because it provides a snapshot of your initial configuration files, in case you start to reconfigure your system and then want to restore the configuration files to their original state. Use the OracleAS Backup and Recovery Tool to back up the configuration files, for example:

```
prompt> cd $HOME/infra/backup_tool/AS_BR  
prompt> ./bkp_restore.sh -m backup_config
```

Verify that the Configuration Files Have Been Created

A subdirectory with the name config_bkp_<YYYY_MM_DD_HH_MM> is created in the directory specified by the config_backup_path variable of the config.inp file.

Verify it with:

```
prompt> cd $HOME/infra/backup_tool/configbackup  
prompt> ls  
config_bkp_2005-03-18_12-19
```

Step 3: Back Up the Middle Tiers

- **For each middle-tier installation:**
 - **Back up the home directory.**
 - **Back up the configuration files.**
- **Back up the DCM file-based repository.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Step 3: Back Up the Middle Tiers

For each middle-tier installation in your environment, perform the following steps:

1. Back up the middle-tier home directory by performing a complete backup of all files in the middle-tier directory by using your preferred operating system command, such as `tar` or `cpio`. Be sure to perform this backup as `root` because some of the files in the directory are owned by `root`. It is important to perform the backup so that file owners, groups, permissions, and timestamps are preserved. For example:

```
prompt> cd $HOME/portal
prompt> tar cvf portal-home-031110.bak *
```

2. Back up the middle-tier configuration files by using the OracleAS Backup and Recovery Tool. For example:

```
prompt> cd $HOME/portal/backup_tool/AS_BR
prompt> ./bkp_restore.pl -m backup_config
```

3. If you are using a DCM file-based repository, back up (export) the repository. The file-based repository exists in one of your middle-tier instances (the repository host instance). You need to perform this step only in the directory of the repository host instance:

```
prompt> cd $ORACLE_HOME/dcm/bin/
prompt> ./dcmctl exportRepository -file
prompt> $HOME/portal/backup_tool/portal_home.bak
```

Oracle Application Server 10g R2: Administration I 19-32

Step 4: Back Up Your Oracle System Files

On each host in your Oracle Application Server environment:

- 1. Back up your Oracle system files using your preferred operating system command, such as `tar` or `cpio`.**
- 2. If the `oraInventory` directory resides outside your Oracle Application Server home directory, then back up the `oraInventory` directory using your preferred operating system command, such as `tar` or `cpio`.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Step 4: Back Up Your Oracle System Files

On each host in your Oracle Application Server environment:

1. Back up your Oracle system files using your preferred operating system command, such as `tar` or `cpio`. Consult your operating system-specific documentation to determine which directory contains your Oracle system files. For example, on Linux/UNIX systems, they may be in the `/etc` or the `/var/opt/oracle` directory.
2. If the `oraInventory` directory resides outside your Oracle Application Server home directory, back up the `oraInventory` directory using your preferred operating system command, such as `tar` or `cpio`.

Step 5: Start Your Oracle Application Server Environment

- Start the Infrastructure.
- Start the middle-tier instances.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Step 5: Start Your Oracle Application Server Environment

Start the Infrastructure

If your Infrastructure contains a Metadata Repository, start it as follows:

1. Set the ORACLE_HOME environment variable to the Infrastructure home directory.
2. Set the ORACLE_SID environment variable to the Metadata Repository SID.
3. Start the Net Listener:

```
prompt> $ORACLE_HOME/bin/lsnrctl start
```

4. Start the Metadata Repository instance:

```
prompt> $ORACLE_HOME/bin/sqlplus /nolog
```

```
SQL> connect SYS as SYSDBA
```

```
SQL> startup
```

```
SQL> quit
```

Step 5: Start Your Oracle Application Server Environment (continued)

If your Infrastructure contains Identity Management, start it as follows:

1. Start the components:

```
prompt> $ORACLE_HOME/opmn/bin/opmnctl startall
```

This command starts OPMN and all OPMN-managed processes, such as DCM, Oracle HTTP Server, OC4J instances, and Oracle Internet Directory.

2. Start Application Server Control:

```
prompt> $ORACLE_HOME/bin/emctl start iasconsole
```

Start the Middle-Tier Instances

To start a middle-tier instance, perform the following steps:

1. If the middle-tier instance uses Infrastructure services, such as Identity Management or a Metadata Repository, make sure that they are started.

2. Start the components:

```
prompt> $ORACLE_HOME/opmn/bin/opmnctl startall
```

This command starts OPMN and all OPMN-managed processes such as DCM, Oracle HTTP Server, OC4J instances, and Web Cache.

3. Start Application Server Control:

```
prompt> $ORACLE_HOME/bin/emctl start iasconsole
```

This procedure applies to all middle-tier instance types:

- J2EE and Web Cache
- Portal and Wireless

Restore Procedures

The following scenarios are possible to rebuild your Oracle Application Server environment:

- Restoring an Infrastructure to the same host
- Restoring an Infrastructure to a new host
- Restoring and recovering Metadata Repository
- Restoring the Infrastructure configuration files
- Restoring a middle-tier installation to the same host
- Restoring a middle-tier installation to a new host
- Restoring middle-tier configuration files

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Restore Procedures

The following scenarios are possible to rebuild your Oracle Application Server environment:

- Restoring an Infrastructure to the same host includes the following steps:
 - Step 1. Stop the Infrastructure.
 - Step 2. Restore the Infrastructure Oracle home.
 - Step 3. Restore the Infrastructure configuration files.
 - Step 4. Restore and recover Metadata Repository.
 - Step 5. Start the Infrastructure.
- Restoring an Infrastructure to a new host (covered in detail in later slides) needs the following approach:
 - Step 1. Prepare the new host.
 - Step 2. Restore the Oracle system files.
 - Step 3. Restore the Infrastructure Oracle home.
 - Step 4. Restore the Infrastructure configuration files.
 - Step 5. Restore and recover Metadata Repository.
 - Step 6. Start the Infrastructure.

Restore Procedures (continued)

- Restoring and recovering Metadata Repository: The best practices for restoring and recovering Metadata Repository are explained in detail in the *Oracle Application Server Administrator's Guide 10g Release 2*, and help you to determine the best method for restoring and recovering your Metadata Repository.
- Restoring Infrastructure configuration files (covered in detail in later slides):
 - Step 1. Stop the Infrastructure.
 - Step 2. Restore the Infrastructure configuration files.
 - Step 3. Apply the recent administrative changes.
 - Step 4. Start the Infrastructure.
- Restoring a middle-tier installation to the same host (covered in detail in later slides):
 - Step 1. Stop the middle-tier instance.
 - Step 2. Restore the middle-tier Oracle home.
 - Step 3. Restore the middle-tier configuration files.
 - Step 4. Start the middle-tier instance.
- Restoring a middle-tier installation to a new host (covered in detail in later slides):
 - Step 1. Prepare the new host.
 - Step 2. Restore the Oracle system files.
 - Step 3. Restore the middle-tier Oracle home.
 - Step 4. Restore the middle-tier configuration files.
 - Step 5. Restore the DCM file-based repository (if required).
 - Step 6. Set the new host name and IP address (if required).
 - Step 7. Start the middle-tier instance.
- Restoring middle-tier configuration files (covered in detail in later slides):
 - Step 1. Stop the middle-tier instance.
 - Step 2. Restore the middle-tier configuration files.
 - Step 3. Apply recent administrative changes.
 - Step 4. Start the middle-tier instance.

Restoring OracleAS Infrastructure to a New Host

- You may need to restore OracleAS Infrastructure to a new host system, in case of a host (system) failure.
- To restore OracleAS Infrastructure to a new host, perform the following steps:
 1. Prepare the new host.
 2. Restore the Oracle system files.
 3. Restore the Infrastructure home directory.
 4. Restore and recover Metadata Repository.
 5. Start Metadata Repository.
 6. Start OracleAS Infrastructure and Application Server Control.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Restoring OracleAS Infrastructure to a New Host

1. Prepare the new host:
 - The first step is to prepare a new host that has identical system configuration as the original host. On the new host, make sure that the information such as host name, virtual host name, domain name, IP address, hardware platform, operating system release, and patch levels is identical to the original host.
 - Make sure that the entries such as IP address, host name, and aliases for the new host in `/etc/hosts` are identical to the old `/etc/hosts` file.
 - Check the port usage on the new host and ensure that there are no conflicts.
 - On the new host, create an operating system user that is identical to the user who installed OracleAS Infrastructure on the original host. The following attributes should be the same: username, user ID, group name, group ID, environment profile, and shell. The user may have the same password or a different password from the original user.

Restoring OracleAS Infrastructure to a New Host (continued)

2. Restore the Oracle system files:
 - Restore the Oracle system files and Oracle Inventory from your complete cold backup. If the oraInventory directory resided in a directory that was separate from the Infrastructure home directory, then restore it too.
3. Restore the Infrastructure home directory:
 - Create an empty directory using the same mount point and full path as the original Infrastructure home directory. Do not use symbolic links anywhere in the path.
 - Make sure that the directory is on a file system with enough space to hold the Infrastructure.
 - Make sure that the directory is owned by the same user and group as on the original host.
 - To restore the OracleAS Infrastructure home directory:
 - Restore the backup (tar, cpio) of the Infrastructure home directory from your complete cold backup. Ensure that your method of restoring the files preserves the original owner, group, permissions, and timestamps.
 - Restore the configuration file backup from your most recent partial online backup.

For example, to do this by using the OracleAS Backup and Recovery Tool:

```
prompt> cd $HOME/infra/backup_tool/AS_BR
prompt> ./bkp_restore.sh -m restore_config -t
config_bkp_2003-11-10_12-19
```
4. Restore and recover Metadata Repository:
 - Restore and recover Metadata Repository from your most recent backup.
 - You can perform this step by using your own procedures or the OracleAS Backup and Recovery Tool.
 - For example, to do this using the tool:

```
prompt> cd $HOME/infra/backup_tool/AS_BR
prompt> ./bkp_restore.sh -m restore_db
```
 - Set file permissions by running the following command as root:

```
prompt> $HOME/infra/root.sh
```

Note: It is not possible to do a full backup of Metadata Repository to a new node because complete recovery is not possible due to the absence of online redo logs. A restore is done with a control file or through point-in-time recovery (that is, `bkp_restore.sh -m restore_db -c -u <timestamp>`). If this command returns an error and the log shows that the data files were restored and recovered, then issue the SQL command `alter database open resetlogs`. The database is opened in a consistent state. Also, after recovery is complete, create a TEMP tablespace on the recovered database using:

```
SQL> alter tablespace "TEMP" add tempfile
'ORACLE_HOME/oradata/GDB/temp01.dbf'
size 5120K autoextend on next 8k maxsize unlimited;
```

Restoring OracleAS Infrastructure to a New Host (continued)

5. Start OracleAS Infrastructure:

- Log in as the user that owns the Infrastructure home directory. Set the ORACLE_HOME and ORACLE_SID environment variables.
- Start the Metadata Repository listener:
prompt> \$ORACLE_HOME/bin/lsnrctl start
- Start Metadata Repository:
prompt> sqlplus /nolog
SQL> connect sys/password as sysdba
SQL> startup

6. Start Identity Management and Application Server Control:

```
prompt> $ORACLE_HOME/opmn/bin/opmnctl startall  
prompt> $ORACLE_HOME/bin/emctl start iasconsole
```

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Restoring OracleAS Infrastructure Configuration Files

- You may have to restore the OracleAS Infrastructure configuration files if the:
 - OracleAS Infrastructure configuration files are corrupted or lost
 - OracleAS Infrastructure installation files are available
- To restore the configuration files, perform the following steps:
 1. Stop OracleAS Infrastructure.
 2. Restore the configuration files from backup.
 3. Apply administrative changes made after backup was taken.
 4. Start OracleAS Infrastructure.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Restoring OracleAS Infrastructure Configuration Files

Use the following steps to restore OracleAS Infrastructure configuration files:

1. Stop the OracleAS Infrastructure instance if it is operational:

```
prompt> cd $HOME/infra/opmn/bin
prompt> ./opmnctl stopall
```
2. Restore configuration files:

Restore the configuration files from your most recent partial online backup using the OracleAS Backup and Recovery Tool:

```
prompt> cd $HOME/infra/backup_tool/AS_BR
prompt> ./bkp_restore.sh -m restore_config -t
config_bkp_2003-11-10_12-19
```
3. Apply recent administrative changes:

If you made any administrative changes since the last time you did an online backup (used in the preceding step), reapply them now.
4. Start the Infrastructure:

```
prompt> cd $HOME/infra/opmn/bin
prompt> ./opmnctl startall
```

Restoring Middle Tier to the Same Host

- **You may need to restore the middle tier to the same host when:**
 - You have lost some or all of the Oracle software files
 - Your host system and configuration files are available
- **To restore the middle tier to the same host, perform the following steps:**
 1. Stop the middle-tier instance if it is still running.
 2. If the instance was associated with a configuration repository, then make sure that the corresponding repository instance is operational.
 3. Restore the middle-tier home directory.
 4. Start the middle tier.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Restoring Middle Tier to the Same Host

1. Stop the middle-tier instance if it is still running:

```
prompt> cd $HOME/mi01/opmn/bin
prompt> ./opmnctl stopall
```
2. If the middle tier was associated with OracleAS Infrastructure, then make sure that the Infrastructure is up. During the restoration of the middle tier, the configuration files are synchronized with OracleAS Metadata Repository.
3. Restore the middle-tier home directory:
 - Restore the backup of the middle-tier home directory from your complete cold backup. Ensure that your method of restoring the files preserves the original owner, group, permissions, and timestamps.
 - Restore the configuration file backup from your most recent partial online backup:

```
prompt> cd $HOME/mi01/backup_tool/AS_BR
prompt> ./bkp_restore.sh -m restore_config -t
config_bkp_2003-11-07_11-21
```
4. Start the middle-tier instance:

```
prompt> cd $HOME/mi01/opmn/bin
prompt> ./opmnctl startall
```

Restoring Middle Tier to a New Host

- You may need to restore and recover a middle-tier installation to a new host when the system hosting the middle tier has a failure.
- Restoring a middle-tier installation to a new host includes the following steps:
 1. Prepare your host.
 2. Restore the Oracle system files.
 3. Restore the middle-tier home directory.
 4. Synchronize the middle tier with the new host.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Restoring Middle Tier to a New Host

1. Prepare your host:
 - The first step is to prepare a new host that has an identical system configuration as the original host. Refer to the record you created for the OracleAS Backup and Recovery Tool.
 - On the new host, make sure that the hardware platform and the operating system release and patch levels are identical to the original host.
 - Make sure that the entry for the new host in `/etc/hosts` lists the IP address, host name, and aliases in the same order as the old `/etc/hosts` file.
 - Check the port usage on the new host. Make sure that there are no processes using the same ports as the Oracle Application Server installations you are about to restore. If there are, you must reconfigure these processes to use different ports before you begin restoring your Oracle Application Server installations.
 - On the new host, create an operating system user that is identical to the user who installed Oracle Application Server on the original host. The following attributes should be the same: username, numerical user ID, group name, numerical group ID, environment profile, and shell.

Note: You can restore the middle tier to a new node with a different host name and IP address. In this case, after full recovery of the middle-tier node, you must run the `hostname/IP change` script before bringing up the middle-tier instance.

Oracle Application Server 10g R2: Administration I 19-43

Restoring Middle Tier to a New Host (continued)

2. Restore Oracle system files:

- Create the middle-tier home directory:
Create an empty directory using the same mount point and full path as the original middle-tier home directory. Do not use symbolic links anywhere in the path.
Make sure that the directory is on a file system with enough space to hold the middle-tier installation.
Make sure that the directory is owned by the same user and group as on the original host.
- Restore the Oracle system files from your complete cold backup.
- If the `oraInventory` directory resided in a directory that was separate from the middle-tier home directory, then restore that.

3. Restore the middle-tier home directory:

- Restore the backup (`tar`, `cpio`) of the middle-tier home directory from your complete cold backup. Ensure that your method of restoring the files preserves the original owner, group, permissions, and timestamps.
- Restore the configuration file backup from your most recent partial online backup:

```
prompt> cd $HOME/mi01/backup_tool/AS_BR
prompt> ./bkp_restore.sh -m restore_config -t
config_bkp_2003-11-07_11-21
```

- If the file-based repository on the original host was lost, you must restore (import) the file-based repository to the new host. This step is required only if you are using a file-based repository, you are restoring to a new host, and you are restoring the repository host instance.

Stop the DCM daemon on all other instances in the farm by running the following command in the home directory of each instance:

```
prompt> cd $ORACLE_HOME/opmn/bin
prompt> ./opmnctl stopproc ias-component=dcm-daemon
```

- Restore (import) the file-based repository from the most recent backup to the new host:

```
prompt> cd $ORACLE_HOME/dcm/bin/
prompt> ./dcmctl importRepository -file file_name
```

- When you run the `importRepository` command, the middle-tier instance that you are currently restoring on the new host becomes the repository host instance. If you intend to continue to use the original host, you must notify the original host that it is no longer the repository host instance. To do this, run the following command in the middle-tier instance on the original host:

```
prompt> $ORACLE_HOME/dcm/bin/dcmctl repositoryRelocated
```

- Start the DCM daemon on all other instances in the farm by running the following command in the home directory of each instance (do not start DCM in the instance you are currently restoring):

```
prompt> cd $ORACLE_HOME/opmn/bin
prompt> ./opmnctl startproc ias-component=dcm-daemon
```

Restoring Middle Tier to a New Host (continued)

4. Synchronize the middle tier with the new host:
 - Set file permissions by running the following command as root:

```
prompt> $ORACLE_HOME/root.sh
```
 - If the new host has the same host name and IP address as the old host, then you can start the middle-tier instance as follows, and you have completed the procedure:

```
prompt> $ORACLE_HOME/opmn/bin/opmnctl startall  
prompt> $ORACLE_HOME/bin/emctl start iasconsole
```

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Restoring Middle-Tier Configuration Files

- You can restore the lost or corrupted configuration files in a middle tier.
- To restore configuration files, perform the following steps:
 1. Stop the middle tier.
 2. Restore configuration files.
 3. Apply recent administrative changes.
 4. Start the middle tier.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Restoring Middle-Tier Configuration Files

1. Stop the middle tier.
Use `opmnctl` to stop all processes of the middle-tier instance:

```
prompt> cd $HOME/mi01/opmn/bin
prompt> ./opmnctl stopall
```
2. Restore the configuration files.
Restore the configuration files from your most recent partial online backup:

```
prompt> cd $HOME/mi01/backup_tool/AS_BR
prompt> ./bkp_restore.sh -m restore_config -t
config_bkp_2003-11-07_11-21
```
3. Apply recent administrative changes.
If you made any administrative changes since the last time you did a partial online backup, then reapply them now.
4. Start the middle tier.
Use `opmnctl` to start all processes of the middle-tier instance.

```
prompt> cd $HOME/mi01/opmn/bin
prompt> ./opmnctl startall
```


Summary

In this lesson, you should have learned how to:

- **Install the Oracle Application Server Backup and Recovery Tool**
- **Configure the Backup and Recovery Tool by using Oracle Application Server Control**
- **Configure Oracle Application Server for full backup**
- **Perform a full Oracle Application Server backup**
- **Perform a partial online backup**
- **Restore a middle tier from backup**
- **Recover OracleAS Infrastructure from backup**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Appendix A

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Practice 2: Oracle Application Server 10g: Key Components and Features

This practice is designed to reinforce your understanding of the key features of Oracle Application Server, and the installation and deployment options that Oracle Application Server provides.

1. Which feature of Oracle Application Server significantly improves the performance and scalability of heavily loaded Web sites?
 - a. Oracle Internet Directory
 - b. OracleAS Clustering
 - c. OracleAS Single Sign-On
 - d. OracleAS Web Cache
2. Select the two main services of the Oracle Application Server architecture.
 - a. Management Services
 - b. Security Services
 - c. Integration and Commerce Services
 - d. Data Warehouse Services
 - e. Reports Services
3. OracleAS Portal is a simple, browser-based environment for building and deploying enterprise information portals (EIPs). What does an EIP provide?
 - a. An EIP provides access to summarized versions of applications and Web content in defined regions of the page or portlets.
 - b. An EIP is a user tool that can be used only on an individual basis.
 - c. An EIP provides a management tool for the administrator to consolidate all of the company's Web pages.
4. OracleAS Web Cache enables you to perform which of the following? (Choose three.)
 - a. Accelerate static and dynamic content delivery
 - b. Implement Oracle Internet Directory
 - c. Reduce your hardware and administration costs
 - d. Cluster multiple OracleAS Web Cache instances to avoid a single point of failure
5. Which is *not* considered an installation type of Oracle Application Server?
 - a. Portal and Wireless
 - b. Data Warehousing
 - c. J2EE and Web Cache
6. The components of OracleAS Infrastructure are _____.
 - a. OracleAS Metadata Repository, Single Sign-On (SSO) server, and Oracle Internet Directory server
 - b. OracleAS Metadata Repository and Oracle Forms server
 - c. OracleAS Metadata Repository and OracleAS Developer Kits
 - d. OracleAS Metadata Repository only
7. Assume the scenario of a middle tier containing multiple instances of J2EE and Web Cache, and Portal and Wireless. Which of these can be combined on one host?

- a. J2EE and Web Cache, and Portal and Wireless only
 - b. J2EE and Web Cache, and Unified Messaging only
 - c. Any combination of these can coexist on one host
8. Which component is required to be installed and configured before installing Portal and Wireless?
- a. Oracle Internet Directory
 - b. OracleAS Infrastructure
 - c. Single Sign-On (SSO) server
9. Specify the order in which you should install Identity Management and Metadata Repository on different systems.
- a. The order of installation does not matter when you install the components on separate systems.
 - b. Install Identity Management and then install Metadata Repository.
 - c. Install Metadata Repository and then install Identity Management.
 - d. You cannot install Metadata Repository and Identity Management on separate systems.

Practice 3: Installing OracleAS Infrastructure

This practice reinforces the understanding of the process of installing OracleAS Infrastructure.

Host Name	ORACLE_HOME	Instance Name	Database SID
	/home/oracle/infra	infra	infra

1. Log in to your system as the `oracle` user, and check the free space available in your `$HOME` directory.
2. Make sure that your system has the correct host file settings.

Verify that you have 1 GB of memory and swap area available on your system. (Use the `free` command.)
3. Verify that you have the Red Hat Linux operating system by using the `uname` or `rpm` command.
4. Check the kernel semaphore (`sem`) and shared memory settings, such as `shmmx`, `shmmni`, and `shmall`.

5. Verify that the `nofile` value in the `/etc/security/limits.conf` file is as follows:

```
*      soft  nproc  2047
*      hard  nproc  16384
*      soft  nofile  2048
*      hard  nofile  65536
```

6. Change the directory to the `Disk1` directory of the `/modules/stage/AS10g` directory.
7. Install OracleAS Infrastructure with the following parameters:

Window	Choices
Initially, you would notice the Welcome screen that highlights the Oracle Universal Installer.	
1. Specify the inventory directory and credentials.	Inventory directory: /home/oracle/oraInventory Operating System Group Name: dba
After you specify the inventory directory and credentials, you will be prompted by the Oracle Universal Installer (Installer) to run the <code>/home/oracle/oraInventory/orainstRoot.sh</code> script. Invoke a separate shell window, and run the <code>/home/oracle/oraInventory/orainstRoot.sh</code> script as the super (<code>root</code>) user.	
2. Specify File Locations	Oracle Home name: infra Destination path: /home/oracle/infra
3. Select a Product to Install	OracleAS Infrastructure 10g 10.1.2.0.2
4. Select Installation Type	Identity Management and Metadata Repository

5. Confirm Preinstallation requirements	Root Privileges
6. Select Configuration Options	Oracle Internet Directory OracleAS Single Sign-On OracleAS Delegated Administration Service OracleAS Directory Integration and Provisioning OracleAS Certificate Authority (OCA)
7. Select Port Configuration Options	Select Automatic .
8. Specify Namespace in Internet Directory	Suggested Namespace
9. Specify OCA Distinguished Name	Typical DN Common Name: ST Certificate Authority Organizational Unit: ST Curriculum Organization: Oracle Corporation
10. Specify OCA Key Length	2048
11. Specify OCA Administrator Password	welcome1
12. Specify Database Configuration Options	Global Database Name: infra.us.oracle.com SID: infra Database File Location: /home/oracle/oradata
13. Specify Database Schema Passwords	Select Use the same password for all accounts . Enter welcome1 as the password.
14. Specify Instance Name and ias_admin Password	Instance Name: infra ias_admin Password: welcome1 Confirm Password: welcome1
15. Before the configuration assistants are invoked, run root.sh from the /home/oracle/infra directory.	Run root.sh with default selections. Do not make any changes when you run the root.sh script.

8. At the end of the installation, note the URLs for Oracle HTTP Server and Application Server Control Console for Oracle Application Server.
9. Which file has the information about the ports that are allocated to different Oracle Application Server components during the installation?
10. Invoke the Welcome page of your Oracle HTTP Server.
11. Invoke Application Server Control for your OracleAS Infrastructure.
12. Invoke the Oracle Internet Directory Server page from Application Server Control and note the OID server status and port.

13. View the `set_infra_env.sh` script in your `$HOME/labs` directory. Use this script to set your `ORACLE_HOME`, `ORACLE_SID`, and `PATH` environment variables. (**Note:** Ensure that the user has permission to execute the script.)

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Practice 4: Installing OracleAS Middle Tier

The practice demonstrates how to install the portal and wireless installation type of the middle tier.

Host Name	ORACLE_HOME	Instance Name	Oracle HTTP Server Port	Application Server Control Port
	/home/oracle/portal	portal		

1. Verify that Application Server Control is running on OracleAS Infrastructure.
2. Invoke the Oracle Universal Installer (OUI) to begin installing Oracle Application Server. The installable image is located in the /modules/stage/AS10g directory.

If you get the following error, enter y and proceed:

Some optional prerequisite checks have failed (see above). Continue? (y/n)

Window	Choice/Action
Specify File Locations	Name = portal Path = /home/oracle/portal
Select a Product to Install	Oracle Application Server 10g 10.1.2.0.2.
Select Installation Type	Portal and Wireless
Confirm Pre-Installation Requirements	Root Privileges
Select Configuration Options	OracleAS 10g Portal
Select Port Configurations	Select Automatic.
Register with Oracle Internet Directory	Infrastructure host name, and Oracle Internet Directory port. Use the Application Server Control Console Ports property page for viewing the current Oracle Internet Directory port (Non SSL Port). <i>Do not select</i> "Use only SSL connections with this Oracle Internet Directory."
Specify OID Login	Enter the password (welcome1).
Select OracleAS 10g Metadata Repository	Note the database connect string
Specify Instance Name and ias_admin Password	Instance Name: portal Password: welcome1

- When prompted by the Installer, **invoke a separate shell window and run the /home/oracle/portal/root.sh script as superuser (root). Select not to overwrite the files when running the root.sh script.**
- At the end of successful installation, **note the URLs for Application Server Control and Oracle HTTP Server for the middle-tier installation.**

3. Test Oracle HTTP Server, Web Cache, and Application Server Control.
4. Access Application Server Control of your middle-tier installation with a URL.

The second part of the practice demonstrates how to upgrade OracleAS Portal middle tier from 10.1.2.0.2 to 10.1.4.

1. Stop all processes associated with the middle tier that uses the existing portal schema.
2. Stop the Application Server Console.
3. Verify that the OracleAS Metadata Repository database, listener for the metadata repository database, and OID instance processes are running.
4. Perform the metadata repository upgrade.
5. Start all processes associated with the middle tier that uses the existing portal schema.
6. Start the Application Server Console.
7. Verify the metadata repository upgrade.

Practice 5: Using Oracle Application Server Management Tools

This practice demonstrates how to use the Application Server Control Console, `opmnctl`, and `dcmdctl` to manage Oracle Application Server installations.

1. Verify that the database listener for the database with OracleAS Infrastructure is running.
2. Verify that the database is started up.
3. Start Application Server Control for OracleAS Infrastructure and the middle-tier instances if they have not already started.
4. Connect to Application Server Control of your OracleAS Infrastructure to obtain information about the Infrastructure instance. Check whether any component is not running.
5. Using Application Server Control, access the middle-tier instance and check the status of the components.
6. Access the Oracle HTTP Server home page in the middle-tier instance. Stop the Oracle HTTP Server component and start it again.
7. View and monitor components with Topology Viewer.

Using the graphical, real-time viewer tool that Application Server Control offers, perform the following common administrative tasks:

- a. Access the Topology link from the Application Server Control Console of the portal instance.
 - b. Hide the navigator so that the entire page can be used for viewing the topology.
 - c. Verify that your OC4J component is up.
 - d. Scroll down the topology to see the process used by the OC4J instance and view real-time performance metrics, such as CPU Time (seconds) and Memory Usage (MB).
 - e. From the Topology Viewer, click the icon displayed for the process, and navigate to the component home page. After viewing the component home page, click Back in the browser to return to the Topology page.
 - f. Click the arrow icon (>) next to OC4J component, and select Collapse Node.
 - g. Change the refresh option to Manual Refresh and Zoom to Small.
8. View performance metric details.

You can obtain an overview of the application server availability and system resource usage for the server and the individual components from its home page. You can also view a detailed list of all performance metrics being monitored by Application Server Control. You can view the `mod_oc4j` metrics for the Oracle HTTP Server component. Change the refresh interval to Real Time: 30 Second Refresh to see a brief history of metric data.

9. Change your Application Server Control Console port to 1812 by using the `emctl` utility.
10. The Application Server Control Console enables you to list and search log files across Application Server components. View the log files for Oracle HTTP Server from the Application Server Control Console page.
11. Obtain the status of your OracleAS Infrastructure instance using the `opmnctl` command-line utility.
12. Obtain the status of your Oracle Application Server middle-tier instance using the `opmnctl` command-line utility.
13. Stop your portal instance by using `opmnctl`, and stop Application Server Control.
14. Start only the OPMN process, and verify the status of your `portal` instance.
15. Start the Oracle HTTP Server process in your portal instance.
16. Start your OC4J home component, and verify that you have started only home and Oracle HTTP Server.
17. Start up all the components of the portal instance. (Note that the DCM daemon does not start up immediately.)
18. Verify the memory used by the components of your middle tier.
19. Back up the configuration of your portal instance to the `$HOME/labs/backup` directory using the `createArchive` and `exportArchive dcmctl` commands. Use the following information:
 - Name the archive: `portal-1`
 - Location of the exported archive: `$HOME/labs/backup/arch-portal-1`
20. Start Application Server Control for the portal instance:

Practice 6: Configuring and Managing Oracle HTTP Server

This practice demonstrates how to perform the basic configuration for your Oracle HTTP Server. It also demonstrates how to locate the appropriate configuration files on the operating system, and how to use Application Server Control or the appropriate `dcmdctl` commands.

1. Verify that your Oracle HTTP Server of the middle tier is running. Start Oracle HTTP Server if it is not already running. (Use `opmnctl` to get the status of Oracle HTTP Server.)
2. Check whether Application Server Control is running.
3. Name the directory where Oracle HTTP Server for your middle-tier instance is installed, and locate the main configuration file.
4. Obtain the port number on which your Oracle HTTP Server is listening.
5. Enable your Oracle HTTP Server to listen on an additional port: 7785.
6. Verify that the change is reflected in the `httpd.conf` file.
7. Ensure that you have `index.html` file in your `$HOME/labs` directory. Then, change your default document root directory to refer to `/home/oracle/labs` instead of `htdocs`. To test your success, enter the following URL:
`http://<host name>.<domain>:<Oracle HTTP Server port of portal instance>`
8. Invoke another browser window, clear the browser cache, and then access Oracle HTTP Server with the following URL: `http://<host name>.<domain>:<Oracle HTTP Server port of portal instance>`
9. Change the document root back to the original setting. `DocumentRoot` should now point to `/home/oracle/portal/Apache/Apache/htdocs` before you start the next practice.
10. When you try to access the URL `http://<host name>.<domain>:<Oracle HTTP Server port>/`, which file is served by default?
11. Reconfigure Oracle HTTP Server so that it does not listen on port 7785.

Practice 7: Configuring Directives and Virtual Hosts

1. Add two entries (*<IP ADDRESS of your machine> mymachine.us.oracle.com* and *<IP ADDRESS of your machine> mymachine1.us.oracle.com*) in the hosts file to create name-based virtual hosts.
2. Enable your HTTP Server to listen on two additional ports—7800 and 7801.
3. Create a virtual host by using Application Server Control.
4. Note the changes in the `httpd.conf` file.
5. Access the new virtual host.

Make the following changes to your proxy settings:

In your browser, select Edit > Preferences > Advanced > Proxies.

Enter `mymachine.us.oracle.com` and `mymachine1.us.oracle.com` in the “No Proxy for” field.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Practice 8: Configuring and Managing OracleAS Web Cache

This practice demonstrates the tasks in administering OracleAS Web Cache.

1. Stop and start OracleAS Web Cache by using `opmnctl`.
2. Change the OracleAS Web Cache administrator password to `oracle1`.
3. Add a listening port for OracleAS Web Cache.
4. Add a Site to Server mapping for the newly added port.
5. Clear your browser cache and history in the Application Server Control browser window.
6. Clear the OracleAS Web Cache contents. Use the basic content invalidation mechanism.
7. Access the Oracle HTTP Server Welcome page directly by entering the URL `http://<host name>.<domain>:<Oracle HTTP Server Listen port of Portal instance>`, and verify that the page is not cached.
8. Access the Oracle HTTP Server Welcome page using the new Web Cache (port 7890), and verify that the new access has been cached.
9. Using Application Server Control, view the Web Cache Statistics now.
10. Using Application Server Control, disable caching for `.swf` files.
11. Verify that the caching rule reflects the change you have made.
12. Delete the Site to Server mapping you added.
13. Delete the Additional port you added.
14. Re-enable caching for files with the extension `.swf`.

Practice 9: Configuring and Managing OC4J

This practice demonstrates how to access OC4J home pages; identify the different OC4J instances running; and start, stop, and restart instances.

1. Start the Application Server Control of the Portal instance if it is not running.
2. Navigate to the home OC4J component home page. Note the default application properties.
3. List the range of ports that can be used for the communication between the OC4J home component and Oracle HTTP Server.
4. Create a new OC4J component named `my_OC4J`. Which directories and files are created?
5. Start the newly created OC4J instance.
6. Stop and delete the `my_OC4J` OC4J instance.

Practice 10: Deploying Java 2, Enterprise Edition (J2EE) Applications

This practice demonstrates how to configure and deploy various types of J2EE applications, a Web application, and a J2EE (EAR) application to Oracle Application Server, and how to inspect the directories and files that are automatically created when the applications are deployed.

1. Set your Oracle environment by using `$HOME/labs/set_infra_env.sh`. Set up the database with the `USERS` tablespace.
2. Create the `ora01` user and the necessary tables in the `ora01` schema. (Use the `hr_main.sql` script in the `$HOME/labs/HR_Setup` directory. This script creates the `ora01` user and tables for `hrapp`.)
3. Deploy the provided JSP files: `login.jsp` and `error.jsp`. Access the JSPs.
4. Examine the directory:
`$HOME/portal/j2ee/home/application-deployments/default/defaultWebApp/persistence/_pages`. What does it contain and why?
5. Create and start an OC4J component named `my_OC4J` in your Portal instance.
6. Deploy a simple Web Application Archive (WAR) file to the `my_OC4J` component.
7. Using Application Server Control verify that the application has indeed been deployed.
8. Where do you expect to find the file relating to the deployed application?
9. Where do you find the `server.xml` file for this application?
10. Where would you find the `orion-application.xml` file for this application?
11. Where do you find the `lab10-web.ear` file?
12. Examine the `mod_oc4j.conf`, `server.xml` and `default-web-site.xml` files.
13. Deploy the `hrapp.ear` application to `my_OC4J`. The `hrapp.ear` file is located in your `$HOME/labs` directory.
 - a. Navigate to Applications property page.
 - b. Enter or select the following values in the Deploy Application window:
J2EE Application: Click Browse and select `hrapp.ear` from the `/home/oracle/labs` directory
Application Name: `hrapp`

Parent Application: default
Click Continue.

- c. On the URL Mappings for Web Modules page, the URL mapping should be /hrapp.
 - d. The Resource Reference Mappings page appears. In the Data Sources for CMP Entity Beans table, the Data Source field for the Employees and Departments Entity Bean should be jdbc/hrDS. Click Finish.
14. After the application has been deployed, verify that hrapp appears in the Deployed Applications list. Create a Data Source with the following information at the application level:
- Name: hrDS
 - Data Source Class: com.evermind.sql.DriverManagerDataSource
 - JDBC URL: jdbc:oracle:thin:@<hostname>.<domain>:1521:infra (Point to your OracleAS Metadata Repository.) If the database connection fails, try the JDBC URL with the domain name such as
jdbc:oracle:thin:@<hostname>.<domain>:1521:infra.<domain>.
 - JDBC Driver: oracle.jdbc.driver.OracleDriver
 - Username: ora01
 - Select Use Cleartext Password
 - Password: oracle
 - Location: jdbc/hr
 - Transactional(XA) Location: jdbc/xa/hrXADS
 - EJB Location: jdbc/hrDS
 - Connection Retry Interval (seconds): 1
 - Cached Connection Inactivity Timeout (seconds): 30
15. Observe the data-sources.xml configuration file in the
\$HOME/portal/j2ee/my_OC4J/config directory.
16. Observe the server.xml and default-web-site.xml files in the
\$HOME/portal/j2ee/my_OC4J/config directory.
17. On the hrapp application home page, you should be able to see the component modules of the application: Web Module hrweb and EJB Module hrejb.
18. Enter the following URL to run the HR application from the browser:
`http://<host name>.<domain>:<Oracle HTTP Server port of portal instance>/hrapp/`.
You should see the application's welcome page.
19. Click Departments on the Welcome page to list all departments in the HR database.
20. Select a department on the Department list page to list all the employees in that department, or click Employees on the Welcome page to list all employees in the HR database. You can also search employees by name. Select Search on the Welcome page, enter a first name (for example, John), and click Search to list all employees with that first name.

Practice 12: Configuring Oracle Application Server Components in Oracle Internet Directory

This practice demonstrates how to start and stop, and to search Oracle Internet Directory.

1. Match the following:

Command	Description
a. bulkload	1. Can be used to back up directory data
b. ldapadd	2. Deletes a subtree
c. ldifwrite	3. Loads one or more entries using the standard I/O
d. bulkdelete	4. Loads a large number of entries to Oracle Internet Directory server using LDIF files as input

2. Verify that the ORACLE_SID and ORACLE_HOME environment variables are set. Run the \$HOME/labs/set_infra_env.sh script if they have not been already set.
3. Using OPMN, verify that the Oracle Internet Directory server is running.
4. Connect to an Oracle Internet Directory server by using Oracle Directory Manager (ODM).
5. ODM displays the navigation tree, menu bar, and toolbar. When you click any of the node names in the navigation tree, its description is displayed in the right pane.
6. Get the password for the Portal Schema.
7. Close the ODM interface.
8. List the components of Enterprise Identity Management.
9. List the various directory roles.
10. A user should belong to the _____ group to configure Oracle Application Server components.
11. Create a user with ODM using the following values:
Distinguished Name: cn=newOIDuser,cn=Users,dc=us,dc=oracle,dc=com
In the Mandatory Properties:
cn=newOIDuser
sn=newOIDuser
In the Optional Properties:
employeenumber=newOIDuser
givenname=newOIDuser
mail= newOIDuser@host.com
orclIsEnabled=delete the existing value and leave it empty.

```
uid=newOIDuser  
userpassword=newOIDuser1  
Verify that the new user is created.
```

12. Grant the `newOIDuser` user the privilege to create new users.
13. Modify the default password policy by changing the attribute value of Password Maximum Failure (`pwdmaxfailure`) to 2.
14. Change the Oracle Internet Directory administrator password.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Practice 13: Managing OracleAS Portal

This practice provides an overview of managing an OracleAS Portal instance.

1. Extract the scripts from the `lab12.zip` file. Create synonyms and provide execute privileges on necessary packages to the newly created user. You can use `provsyns.sql`.
2. Create database providers in the new database user (`ora01`) by using the `provider.sql` script.
3. Open the home page of OracleAS Portal (Portal:portal component) in Application Server Control. Familiarize yourself with the page controls and information displayed on the page. Explore the Cache Configuration page, the Parallel Page Engine Services home page, and the Providers home page by clicking the corresponding links.
4. Log in to OracleAS Portal as `portal`. Use the password (`welcome1`) that has been specified for the `ias_admin` user during the installation of the Oracle Application Server instance that includes OracleAS Portal.
5. Familiarize yourself with the OracleAS Portal user interface: OracleAS Portal Builder, Portal Navigator, Portlet Repository, and online Help.
6. Browse the OracleAS Portal users and groups by using the Users and Groups portlets.
7. Create a new portal user with the data defined in the table below or specify your own. Assign the DBA and `PORTAL_ADMINISTRATORS` roles to the user. After you create the user, log in to the OracleAS Portal instance as the new user.

Property	Value
Username	mycompany_admin
E-mail Address	admin@mycompany.com
Password	admin123
Roles Assignment	Check DBA and PORTAL_ADMINISTRATORS

8. Import the CompanyPortal page group into the OracleAS Portal instance using the following export/import script and the dump file that are located in the `$HOME/labs` directory: `companyportal.csh` and `companyportal.dmp`. You need the password for the Portal schema.
Analyze the script output by checking the log file in the `labs` directory.
9. Verify the import of the CompanyPortal page group by opening its root page. Add this to your bookmarks.

10. Deploy the sample Web provider as a Web application of the my_OC4J instance. Use SampleWebProvider.ear in the \$HOME/labs file for the deployment and name the Web application as My Web Provider.
11. Register the database provider installed earlier with the following parameters:

Parameter Name	Parameter Value
Name	MY_PROVIDER
Display Name	My Provider
Timeout	20
Timeout Message	My Provider timed out
Implementation Style	Database
Owning Schema	Ora01
Package Name	MY_PROVIDER
Login Frequency	Never

12. Navigate to your portal page and add the installed PL/SQL portlet to the page.
13. Change the password for the mycompany_admin subscriber to welcome1.
14. Enable Refresh Cache for Oracle Internet Directory Parameters.

Practice 14: Configuring OracleAS Portal

1. In your OracleAS Portal instance, set up the self-registration feature that does not require approval.
2. Test the self-registration feature from the CompanyPortal page. To accomplish this task, you need to log out of the OracleAS Portal instance and open the root page of the CompanyPortal by using the bookmark that you created in the last practice of the lesson titled “Managing the OracleAS Portal Instance.” You also need to register a new portal user by using the self-registering feature. Specify information about the new user as defined in the table below or specify your own.

Property	Value
Username	mycompany_user
Password	user123
E-mail Address	user@mycompany.com

3. Install an additional language to the OracleAS Portal instance (for example, French) by using `ptllang`. Make sure that the `ORACLE_HOME` variable is set to the Oracle home directory of the middle tier before running the script. Verify the language installation by checking the `portal_f.log` file.
4. Test the installed language in CompanyPortal by refreshing its root page.

Practice 15: Administering the OracleAS Single Sign-On Server

1. List the components of the OracleAS Single Sign-On server.
2. Increase the duration of the SSO server session to 12 hours.
3. Restart the SSO server by using Application Server Control.
4. Add an external application to the SSO server and save its credentials in the server.
 - Application Name: otn
 - Login URL: <http://otn.oracle.com>
 - User Name/ID Field Name: login
 - Password Field Name: passwd
 - In the Authentication Method section, select POST from the list.
 - In the Additional Fields section, enter the following details:
 - .persistentY in the Field Name field and [off] in the Field Value field.

Note: Because of the firewalls and the security policies of Oracle, you may not be able to access external applications sites.

5. View the OracleAS SSO server monitoring pages.

Practice 16: Managing Access Using Oracle Delegated Administration Services

1. Add “modify user privilege” to the `newOIDuser` user created in the earlier practice.
2. Unlock the account of a user account. In this case, `newOIDuser` is locked. In case it is not, you can lock the user account by entering a wrong password twice.
3. Create a user using OracleAS Portal. OracleAS Portal provides links to Oracle Internet Directory Self-Service Console. You can enter information as shown below:

Basic Information

First Name: newuser2

Last Name: newuser2

User ID: newuser2

Password: newuser2

Confirm Password: newuser2

E-mail Address: newuser2@xyz.com

Additional Personal Details

Single Sign On Enabled: Enabled

Practice 17: Managing and Configuring OracleAS Certificate Authority

Note

- The OCA operational steps are dependent on the Web browser.
- Because you will be using a single browser, it is advisable to clear the certificates each time you change the user (from administrator to user and back to administrator).
- To enable the browser to prompt you before accepting certificates, you can perform the following steps:
 - a. Open Mozilla or Netscape browser and select Edit > Preferences.
 - b. In the Category pane, expand the Privacy and Security node and select Certificates. In the right pane, the certificate-related information is displayed.
 - c. In the Client Certificate Selection section, click the Ask Every Time option button. This enables you to select the client certificate as required for a particular operation. Otherwise, the browser provides the certificate automatically, which may not be correct as per the operation and can cause unexpected errors.
- Ensure that there is no OCA Web Administration certificate. Remember that welcome1 is the password for OCA Administrator.
 - a. Set the ORACLE_HOME-related environments using the `set_infra_env.sh` script.
 - b. Stop the OCA server using the `$HOME/infra/oca/bin/ocactl stop` command
 - c. Run the command:
`$HOME/infra/oca/bin/ocactl revokecert -type WEBADMIN`
 - d. Start the OCA server using the `$HOME/infra/oca/bin/ocactl start` command.
- Before accessing the OCA Administration or User pages, find the port number for the OCA server (usually 6600) using Oracle Enterprise Manager 10g Application Server Control.
- The browser may prompt you when you shift from an encrypted page to an unencrypted page or vice versa. Select the appropriate response and continue.

Practices

1. View the status of the OCA server and start it if it is not started already.
2. Access the OCA administration page and enroll for a certificate. Enter the details as shown below:
Common Name: ocawebadmin
Organization: ABC
Password: welcome1

Practice 18: Securing OracleAS Components by Using SSL

Note

- The OCA operational steps are dependent on the Web browser.
- Because you will be using a single browser, it is advisable to clear the certificates each time you change the user (from administrator to user and back to administrator).
- To enable the browser to prompt you before accepting certificates, you can perform the following steps:
 - a. Open Mozilla or Netscape browser and select Edit > Preferences.
 - b. In the Category pane, expand the Privacy and Security node and select Certificates. In the right pane, the certificate-related information is displayed.
 - c. In the Client Certificate Selection section, click the Ask Every Time option button. This enables you to select the client certificate as required for a particular operation. Otherwise, the browser provides the certificate automatically, which may not be correct as per the operation and can cause unexpected errors.
- Ensure that there is no OCA Web Administration certificate. Remember that welcome1 is the password for OCA Administrator.
 - a. Set the ORACLE_HOME related environments using the `set_infra_env.sh` script.
 - b. Stop the OCA server using the `$HOME/infra/oca/bin/ocactl stop` command.
 - c. Run the command:
`$HOME/infra/oca/bin/ocactl revokecert -type WEBADMIN`
 - d. Start the OCA server using the `$HOME/infra/oca/bin/ocactl start` command.
- Before accessing the OCA Administration or User pages, find the port number for the OCA server (usually 6600) using Oracle Enterprise Manager 10g Application Server Control.
- The browser may prompt you when you shift from an encrypted page to an unencrypted page or vice versa. Select the appropriate response and continue.

Practices

1. Create a new wallet.
2. Create a certificate request for the wallet created.
3. Use OCA User Pages to request for a client certificate. For this practice, select “Use your OracleAS Single sign-on name and password.”
4. Approve the server certificate request ID using the OCA Administration Pages.

Note: Because you are using the same machine and browser profile for accessing and administering certificates, you may need to take the following actions:

- a. Stop the OCA (`ocactl stop`).

b. Revoke Web Administrator Certificate:

\$ ocactl revokecert -type WEBADMIN

c. Start OCA (ocactl start).

d. Also, remove the certificate from your browser.

5. Change the HTTP server configuration to enable SSL.
6. Use OCA User Pages to request for a client certificate.
7. Use Oracle Application Server OCA User Pages to request for a client certificate by using the manual approval authentication.
8. Approve the client certificate request ID using the OCA Administration Pages.
9. Change the HTTP server configuration to “require” client certificate.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Practice 19: Backing Up and Restoring Oracle Application Server

This practice provides experience in backing up and restoring configuration settings. The practices for this lesson are as follows:

- Lab 19.1: Set up the environment
- Lab 19.2: Install the backup and recovery tool
- Lab 19.3: Perform a backup of the middle-tier installation
- Lab 19.4: Restore and verify the backup

For these practices, set `ORACLE_HOME` to `/home/oracle/portal`.

Lab 19.1: Set Up the Environment

Set the `ORACLE_SID`, `ORACLE_HOME`, and `PATH` environment variables appropriately. You can modify the `ORACLE_HOME` environment variable in the `set_infra_env.sh` script in your `$HOME/labs` directory and use it.

Lab 19.2: Install the Backup and Recovery Tool

1. Ensure that the `bkp_restore.pl` script has execute permission.
2. Create directories to hold the backup and log files:
`$HOME/labs/lesson19/backups/portal/log_files`
`$HOME/labs/lesson19/backups/portal/config_files`
3. Edit the `config.inp` file in the `$ORACLE_HOME/backup_restore/config/` directory and enter values for the following variables:
`oracle_home=/home/oracle/portal`
`log_path=/home/oracle/labs/lesson19/backups/portal/log_files`
`config_backup_path=/home/oracle/labs/lesson19/backups/portal/c`
`onfig_files`
4. Execute the script to configure the backup parameters.

Note: This updates parameters in `config.inp` to indicate that this is a middle-tier backup.

Lab 19.3: Perform a Backup of the Middle-Tier Installation

1. Shut down your middle-tier instance.
2. Perform a complete backup of the middle tier.
 - a. Create a DCM archive of the middle-tier instance:
 - b. Back up the Oracle Home of the middle tier:
 - c. Back up the middle-tier configuration files:
Note: If you do not have SSO configured, you will receive an error message stating that the SSO configuration file (`osso.conf`) does not exist and could not be copied. Ignore this error.
 - d. Verify that the configuration files were backed up:

3. Modify a configuration file, for example `httpd.conf`:
 `$ vi httpd.conf`
 `/KeepAliveTimeout 15` (to search for `KeepAliveTimeout`)
 `w` (to advance to the 15)
 `cw` (to change the 15)
 `20` (to set the `KeepAliveTimeout` to 20)
 `Esc` (to escape out of the change)
 `ZZ` (to save and quit out of `vi`)
4. To update the changes to the DCM repository, run the `updateConfig` command.
5. To reflect these changes, start the `HTTP_Server` component.
6. Shut down the middle tier again by stopping all the OPMN processes.
7. Take an online incremental backup of the middle-tier instance configuration.

Lab 19.4: Restore and Verify the Backup

1. Restore the middle tier using the original backup:
 Note: Enter just the name of `<timestamp>` located in:
 `$HOME/labs/lesson19/backups/portal/config_files/`
2. Check the configuration file setting:
 `$ vi httpd.conf`
 `/KeepAliveTimeout` (to see that `KeepAliveTimeout` is set to 15)
 `:q` (to quit from `vi`)
3. Restore the middle tier using the incremental backup.
 Note: Enter just the name of the `<timestamp>` located in:
 `$HOME/labs/lesson19/backups/portal/config_files/`
4. Check the configuration file setting:
 After the DCM archive is restored completely, open a new terminal window and check the configuration file setting.
5. Restart your middle-tier instance.
6. To save space for subsequent practices, remove the backup files.

Appendix B

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Practice 2: Oracle Application Server 10g: Key Components and Features

This practice is designed to reinforce your understanding of the key features of Oracle Application Server, and the installation and deployment options that Oracle Application Server provides.

1. Which feature of Oracle Application Server significantly improves the performance and scalability of heavily loaded Web sites?
 - a. Oracle Internet Directory
 - b. OracleAS Clustering
 - c. OracleAS Single Sign-On
 - d. OracleAS Web Cache

Answer: d

2. Select the two main services of the Oracle Application Server architecture.
 - a. Management Services
 - b. Security Services
 - c. Integration and Commerce Services
 - d. Data Warehouse Services
 - e. Reports Services

Answer: a, b

3. OracleAS Portal is a simple, browser-based environment for building and deploying enterprise information portals (EIPs). What does an EIP provide?
 - a. An EIP provides access to summarized versions of applications and Web content in defined regions of the page or portlets.
 - b. An EIP is a user tool that can be used only on an individual basis.
 - c. An EIP provides a management tool for the administrator to consolidate all of the company's Web pages.

Answer: a

4. OracleAS Web Cache enables you to perform which of the following? (Choose three.)
 - a. Accelerate static and dynamic content delivery
 - b. Implement Oracle Internet Directory
 - c. Reduce your hardware and administration costs
 - d. Cluster multiple OracleAS Web Cache instances to avoid a single point of failure

Answer: a, c, d

5. Which is *not* considered an installation type of Oracle Application Server?
 - a. Portal and Wireless
 - b. Data Warehousing
 - c. J2EE and Web Cache

Answer: b

6. The components of OracleAS Infrastructure are ____.
- a. OracleAS Metadata Repository, Single Sign-On (SSO) server, and Oracle Internet Directory server
 - b. OracleAS Metadata Repository and Oracle Forms server
 - c. OracleAS Metadata Repository and OracleAS Developer Kits
 - d. OracleAS Metadata Repository only

Answer: a

7. Assume the scenario of a middle tier containing multiple instances of J2EE and Web Cache, and Portal and Wireless. Which of these can be combined on one host?
- a. J2EE and Web Cache, and Portal and Wireless only
 - b. J2EE and Web Cache, and Unified Messaging only
 - c. Any combination of these can coexist on one host

Answer: c

8. Which component is required to be installed and configured before installing Portal and Wireless?
- a. Oracle Internet Directory
 - b. OracleAS Infrastructure
 - c. Single Sign-On (SSO) server

Answer: b

9. Specify the order in which you should install Identity Management and Metadata Repository on different systems.
- a. The order of installation does not matter when you install the components on separate systems.
 - b. Install Identity Management and then install Metadata Repository.
 - c. Install Metadata Repository and then install Identity Management.
 - d. You cannot install Metadata Repository and Identity Management on separate systems.

Answer: c

Practice 3: Installing OracleAS Infrastructure

This practice reinforces the understanding of the process of installing OracleAS Infrastructure.

Host Name	ORACLE_HOME	Instance Name	Database SID
	/home/oracle/infra	infra	infra

1. Log in to your system as the `oracle` user, and check the free space available in your `$HOME` directory.

```
Red Hat Enterprise Linux AS release 3 (Taroon Update 3)
Kernel 2.4.21-20.EL on an i686
login: oracle
Password:
```

```
$ df -m $HOME
Filesystem            1M-blocks      Used Available Use% Mounted on
/dev/hda7              13119         4722      7730   38% /
```

2. Make sure that your system has the correct host file settings.

```
$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1      EDRSR16P1 localhost.localdomain localhost
139.185.35.116EDRSR16P0 edRSR16P0.us.oracle.com
```

Verify that you have 1 GB of memory and swap area available on your system. (Use the `free` command.)

```
$ free
              total        used        free      shared    buffers     cached
Mem:          1030084      299928      730156           0       13460      163836
-/+ buffers/cache:  122632      907452
Swap:         1831368           0      1831368
```

3. Verify that you have the Red Hat Linux operating system by using the `uname` or `rpm` command.

```
$ uname -a
Linux EDRSR16P1 2.4.21-20.EL #1 Wed Aug 18 20:58:25 EDT 2004 i686 i686
i386 GNU/Linux
$ rpm -qa | grep kernel
kernel-2.4.21-20.EL
kernel-pcmcia-cs-3.1.31-13
kernel-utils-2.4-8.37.6
$
```

4. Check the kernel semaphore (`sem`) and shared memory settings, such as `shmmmax`, `shmmni`, and `shmall`.

```
$ cat /proc/sys/kernel/sem
```

```

256      32000    100      142
$ cat /proc/sys/kernel/shmmax
4294967295
$ cat /proc/sys/kernel/shmmni
4096
$ cat /proc/sys/kernel/shmall
3279547

```

5. Verify that the `nofile` value in the `/etc/security/limits.conf` file is as follows:

```

*      soft    nproc  2047
*      hard    nproc  16384
*      soft    nofile 2048
*      hard    nofile 65536
$ cat /etc/security/limits.conf

```

6. Change the directory to the `Disk1` directory of `/modules/stage/AS10g` directory.

```
$ cd /modules/stage/AS10g/Disk1
```

7. Install OracleAS Infrastructure with the following parameters:

Window	Choices
Initially, you would notice the Welcome screen that highlights the Oracle Universal Installer.	
1. Specify the inventory directory and credentials.	Inventory directory: /home/oracle/oraInventory Operating System Group Name: dba
After you specify the inventory directory and credentials, you will be prompted by the Oracle Universal Installer (Installer) to run the <code>/home/oracle/oraInventory/orainstRoot.sh</code> script. Invoke a separate shell window, and run the <code>/home/oracle/oraInventory/orainstRoot.sh</code> script as the super (root) user.	
2. Specify File Locations	Oracle Home name: infra Destination path: /home/oracle/infra
3. Select a Product to Install	OracleAS Infrastructure 10g 10.1.2.0.2
4. Select Installation Type	Identity Management and Metadata Repository
5. Confirm Preinstallation requirements	Root Privileges
6. Select Configuration Options	Oracle Internet Directory OracleAS Single Sign-On OracleAS Delegated Administration Service OracleAS Directory Integration and Provisioning OracleAS Certificate Authority (OCA)
7. Select Port Configuration Options	Select Automatic .
8. Specify Namespace in Internet Directory	Suggested Namespace

9. Specify OCA Distinguished Name	Typical DN Common Name: ST Certificate Authority Organizational Unit: ST Curriculum Organization: Oracle Corporation
10. Specify OCA Key Length	2048
11. Specify OCA Administrator Password	welcome1
12. Specify Database Configuration Options	Global Database Name: infra.us.oracle.com SID: infra Database File Location: /home/oracle/oradata
13. Specify Database Schema Passwords	Select Use the same password for all accounts . Enter welcome1 as the password.
14. Specify Instance Name and ias_admin Password	Instance Name: infra ias_admin Password: welcome1 Confirm Password: welcome1
15. Before the configuration assistants are invoked, run root.sh from the /home/oracle/infra directory.	Run root.sh with default selections. Do not make any changes when you run the root.sh script.

\$./runInstaller

8. At the end of the installation, note the URLs for Oracle HTTP Server and Application Server Control Console for Oracle Application Server.

- Oracle HTTP Server: **http://<host name>.<domain>:<Oracle HTTP Server port>**
- Application Server Control: **http://<host name>.<domain>:<Application Server Control port>**

Exit the Installer. Close the installation session and access the command prompt (\$) in the session that invoked the Installer.

9. Which file has the information about the ports that are allocated to different Oracle Application Server components during the installation?

The **portlist.ini** file in the **<ORACLE_HOME>/install** directory contains the information about the ports that are allocated to different components. This file is not updated when you alter the port allocations subsequently.

10. Invoke the Welcome page of your Oracle HTTP Server.

Invoke the Web browser and access the URL **http://<host name>.<domain>:<Oracle HTTP Server port>**, for example, **http://edrsr16p1.us.oracle.com:7777**.

Welcome

to Oracle Application Server 10g Release 2 (10.1.2)

Overview



Oracle Application Server 10g Release 2 (10.1.2) is an integrated, standards-based application platform suite that allows organizations of all sizes to respond better to changing business requirements.

The Oracle Application Server application platform suite can improve your organization's ability to predict and respond to market dynamics, enhance productivity, and simplify your information technology environment, all while allowing you to use your existing investments to their full potential. Oracle Application Server 10g Release 2 (10.1.2) achieves these goals through:

- **Service-Oriented Computing:** Oracle Application Server uses a service-oriented computing architecture to facilitate the development of enterprise applications as business services, which enables you to build a flexible enterprise application infrastructure.
- **Grid Computing:** The Oracle Application Server architecture coordinates the use of large numbers of low cost, modular servers and storage to act as one large computer to run your enterprise applications. This allows you to start small, minimize unused resources, and add processing or storage capacity as you need it.

Release Notes

Read the latest Release Notes on Oracle Technology Network for important information about Oracle Application Server 10g Release 2 (10.1.2).

New Features

For details about new features for Oracle Application Server 10g Release 2 (10.1.2), visit [Oracle Technology Network](#).

Oracle Application Server Logins

To manage and monitor Oracle Application Server, log on to Oracle Enterprise Manager 10g Application Server Control:
username: ias_admin
password: specified during install

11. Invoke Application Server Control for your OracleAS Infrastructure.

- Invoke the Web browser and access Application Server Control using the URL `http://<host name>.<domain>:<Oracle Application Server Control port>`, for example, `http://edrsr16p1.us.oracle.com:1156`.
- To log in, use `ias_admin` as the username and `welcome1` as the password, and click OK. The Farm page appears.

Farm: infra.us.oracle.com

Instances can be grouped and managed together by configuring standalone instances in a common repository. This collection of instances is known as an Oracle Application Server Farm.

Repository Type **Database**

Clusters

Create Cluster

Select Name

Status Instances

There are no clusters in the farm.

Standalone Instances

These instances belong to the farm but are not part of any cluster.

Join Cluster

Select Name	Host	Oracle Home
infra.edrsr16p1	edrsr16p1	/home/oracle/infra

12. Invoke the Oracle Internet Directory Server page from Application Server Control and note the OID server status and port.

- Invoke the browser, and access Application Server Control of your Infrastructure installation (`http://<host name>.<domain>:<Application Server Control Port>`).
- Log in using `ias_admin` as the username and `welcome1` as the password.
- Notice that the Farm page appears as the first interface.
- Click the Infrastructure Instance link under Standalone Instances.

- e. Click Internet Directory in the System Components table to invoke the Oracle Internet Directory Page.
 - f. Note the Oracle Internet Directory Server status and port.
13. View the `set_infra_env.sh` script in your `$HOME/labs` directory. Use this script to set your `ORACLE_HOME`, `ORACLE_SID`, and `PATH` environment variables.
(**Note:** Ensure that the user has permission to execute the script.)

```
$ cat $HOME/labs/set_infra_env.sh
export ORACLE_HOME=$HOME/infra
export ORACLE_SID=infra
export PATH=$ORACLE_HOME/bin:$PATH

$ chmod u+x $HOME/labs/set_infra_env.sh
$ . $HOME/labs/set_infra_env.sh
$ echo $ORACLE_HOME
/home/oracle/infra
```

Practice 4: Installing OracleAS Middle Tier

The first part of the practice demonstrates how to install the portal and wireless installation type of the middle tier.

Host Name	ORACLE_HOME	Instance Name	Oracle HTTP Server Port	Application Server Control Port
	/home/oracle/portal	portal		

1. Verify that Application Server Control is running on OracleAS Infrastructure.

```
$ cd $HOME/infra/bin
$ ./emctl status iasconsole
TZ set to US/Pacific
Oracle Enterprise Manager 10g Application Server Control Release
10.1.2.0.2
Copyright (c) 1996, 2005 Oracle Corporation. All rights reserved.
http://EDRSR16P1:1156/emd/console/aboutApplication
Oracle Enterprise Manager 10g Application Server Control is running.
```

2. Invoke the Oracle Universal Installer (OUI) to begin installing Oracle Application Server. The installable image is located in the /modules/stage/AS10g directory.

If you get the following error, enter y and proceed:

Some optional prerequisite checks have failed (see above). Continue? (y/n)

```
cd /modules/stage/AS10g/Disk1
./runInstaller
```

Window	Choice / Action
Specify File Locations	Name = portal Path = /home/oracle/portal
Select a Product to Install	Oracle Application Server 10g 10.1.2.0.2.
Select Installation Type	Portal and Wireless
Confirm Pre-Installation Requirements	Root Privileges
Select Configuration Options	OracleAS 10g Portal
Select Port Configurations	Select Automatic.
Register with Oracle Internet Directory	Infrastructure host name, and Oracle Internet Directory port. Use the Application Server Control Console Ports property page for viewing the current Oracle Internet Directory port (Non SSL Port). <i>Do not select</i> "Use only SSL connections with this Oracle Internet Directory."

	Host: <input type="text" value="edrsr16p1"/> Port: <input type="text" value="389"/> <input type="checkbox"/> Use only SSL connections with this Oracle Internet Directory
Specify OID Login	Enter the password (welcome1).
Select OracleAS 10g Metadata Repository	Note the database connect string
Specify Instance Name and ias_admin Password	Instance Name: portal Password: welcome1

- When prompted by the Installer, **invoke a separate shell window and run the /home/oracle/portal/root.sh script as superuser (root). Select not to overwrite the files when running the root.sh script.**
 - At the end of successful installation, **note the URLs for Application Server Control and Oracle HTTP Server for the middle-tier installation.**
 - Exit from the Installer window, and close the installation session in the command window that you used to invoke the Installer.
3. Test Oracle HTTP Server, Web Cache, and Application Server Control.
- Access the Welcome page of the middle tier. Oracle HTTP Server for the middle tier may be on port 7779, therefore, use this URL; for example, `http://<host name>.<domain>:<Oracle HTTP Server port>`.
 - Click the Demonstrations tab. In the navigation bar to the left, click the Demonstrations link.
 - Click the Oracle Application Server Web Cache link in the table.
4. Access Application Server Control of your middle-tier installation with the URL `http://<host name>.<domain>:<Application Server Control port>`; for example, `http://edrsr16p1.us.oracle.com:1810`.
Log in as the `ias_admin` user with the password `welcome1`.

Farm: infra.us.oracle.com		
Instances can be grouped and managed together by configuring standalone instances in a common repository. This collection is an Oracle Application Server Farm.		
Repository Type Database		
Clusters		
<div> <div>Select Name</div> <div>There are no clusters in the farm.</div> </div>		
Standalone Instances		
These instances belong to the farm but are not part of any cluster.		
Join Cluster		
Select Name	Host	Oracle Home
infra.edrsr16p1	edrsr16p1	/home/oracle/infra
portal.edrsr16p1	edrsr16p1	/home/oracle/portal

The second part of the practice demonstrates how to upgrade OracleAS Portal middle tier from 10.1.2.0.2 to 10.1.4.

1. Stop all processes associated with the middle tier that uses the existing portal schema.

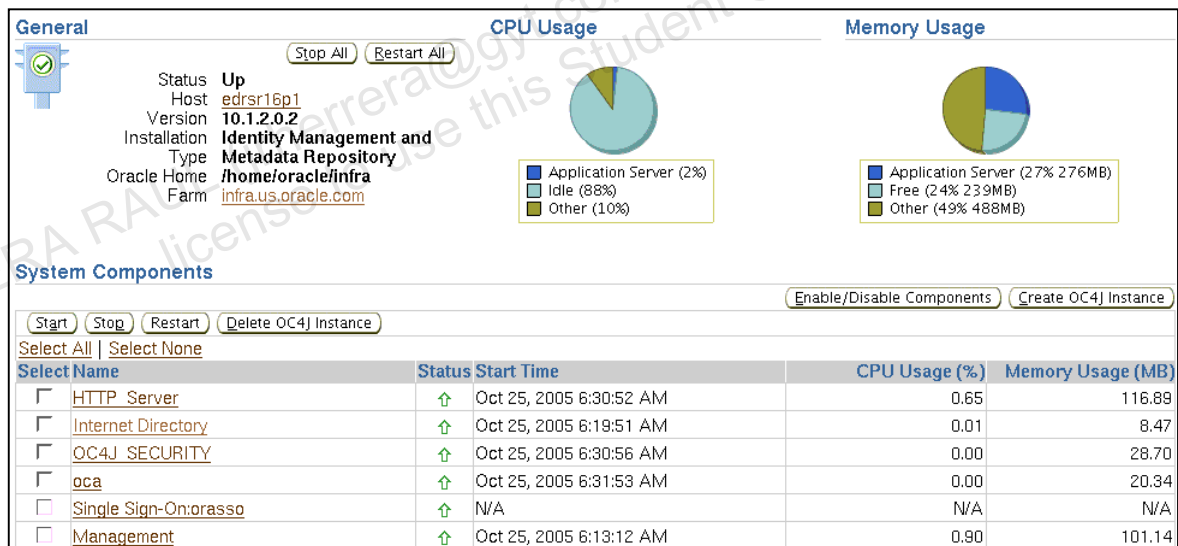
```
$ cd $HOME/portal/opmn/bin/opmnctl stopall
```

2. Stop the Application Server Console.

```
$ cd $HOME/portal/bin/emctl stop iasconsole
```

3. Verify that the OracleAS Metadata Repository database, listener for the metadata repository database, and OID instance processes are running.

- a. Invoke the Web browser and access Application Server Control using the URL `http://<host name>.<domain>:<Oracle Application Server Control port>`; for example, `http://edrsr16p1.us.oracle.com:1156`.
- b. To log in, use `ias_admin` as the username and `welcome1` as the password, and click OK. The Farm page appears.
- c. Click the Infrastructure Instance link under Standalone Instances. Notice that all the processes are running.



4. Perform the metadata repository upgrade.

You need to run the `mrua.sh` script:

```
$ cd /modules/stage/Portal_10.1.4/mrui
$ ./mrui.sh -oracle_home /home/oracle/infra
-oid_host edrsr16p1 -oid_ssl_port 636
```

You need to provide the password for the database SYS user account:

Enter the password for SYS:

You need to provide the password for the OID `cn=orcladmin` administrator account:

Enter the password for `cn=orcladmin`:

5. Start all processes associated with the middle tier that uses the existing portal schema.

```
$ cd $HOME/portal/opmn/bin/opmnctl startall
```


6. Start the Application Server Console.

```
$ cd $HOME/portal/bin/emctl start iasconsole
```

7. Verify the metadata repository upgrade.

- In the Web browser, return to the Farm page and click the portal instance link under Standalone Instances.
- Notice all the components running for the mid-tier and click the Portal:portal component.
- In the OracleAS Metadata Repository Used By Portal section, notice the Repository Version as 10.1.4.0.0

Portal:portalPage Refreshed Nov 7, 2005 3:04:26 A

General

Status

Up

Average Page Requests Per Hour

0

Homepage Download (seconds)

23.881

OracleAS Metadata Repository Used By Portal

Status

Up

Name

infra

Start Time

Nov 3, 2005 2:11:11 AM

Database Version

10.1.0.4.2

Repository Version

10.1.4.0.0


Component Status



OracleAS components used by Portal.

Component	Up/Down
HTTP Server	↑
Parallel Page Engine Services	↑
Providers	↑
Ultra Search	↑

Severity Status

OracleAS components used by Portal that indicate severity status.

Component	Severity
Parallel Page Engine Services	
Providers	✓

OK ✓ Warning  Critical ✗ Unknown 

Practice 5: Using Oracle Application Server Management Tools

This practice demonstrates how to use the Application Server Control Console, `opmnctl`, and `dcmctl` to manage Oracle Application Server installations.

1. Verify that the database listener for the database with OracleAS Infrastructure is running.
 - a. Set the `ORACLE_SID`, `ORACLE_HOME`, and `PATH` environment variables appropriately. You can use the `set_infra_env.sh` script in your `$HOME/labs` directory:

```
$ . $HOME/labs/set_infra_env.sh
```
 - b. Use the `lsnrctl status` command to verify that the database listener is operational:

```
$ lsnrctl status
```
2. Verify that the database is started up.
 - a. Use SQL*Plus to connect to the database and check that you do not receive the message "Connected to an idle instance."

```
$ sqlplus "/ as sysdba"
```
 - b. Enter `exit` to quit SQL*Plus.
3. Start Application Server Control for OracleAS Infrastructure and the middle-tier instances if they have not already started.

```
$ cd $HOME/infra/bin
$ ./emctl status iasconsole
TZ set to US/Pacific
Oracle Enterprise Manager 10g Application Server Control Release
10.1.2.0.2
Copyright (c) 1996, 2005 Oracle Corporation. All rights reserved.
http://EDRSR16P1:1156/emd/console/aboutApplication
Oracle Enterprise Manager 10g Application Server Control is running.
-----
Logs are generated in directory /home/oracle/infra/sysman/log

$ cd $HOME/portal/bin
$ ./emctl status iasconsole
TZ set to US/Pacific
Oracle Enterprise Manager 10g Application Server Control Release
10.1.2.0.2
Copyright (c) 1996, 2005 Oracle Corporation. All rights reserved.
http://EDRSR16P1:1810/emd/console/aboutApplication
Oracle Enterprise Manager 10g Application Server Control is running.
-----
```

4. Connect to Application Server Control of your OracleAS Infrastructure to obtain information about the Infrastructure instance. Check whether any component is not running.
 - a. Click the OracleAS Infrastructure instance link in the table of available instances of the Application Server Control Farm page. If requested, enter `ias_admin` and `welcome1` in the User Name and Password fields.
 - b. On the OracleAS Infrastructure instance home page, verify the status of components in the System Components table.

All the components are running.

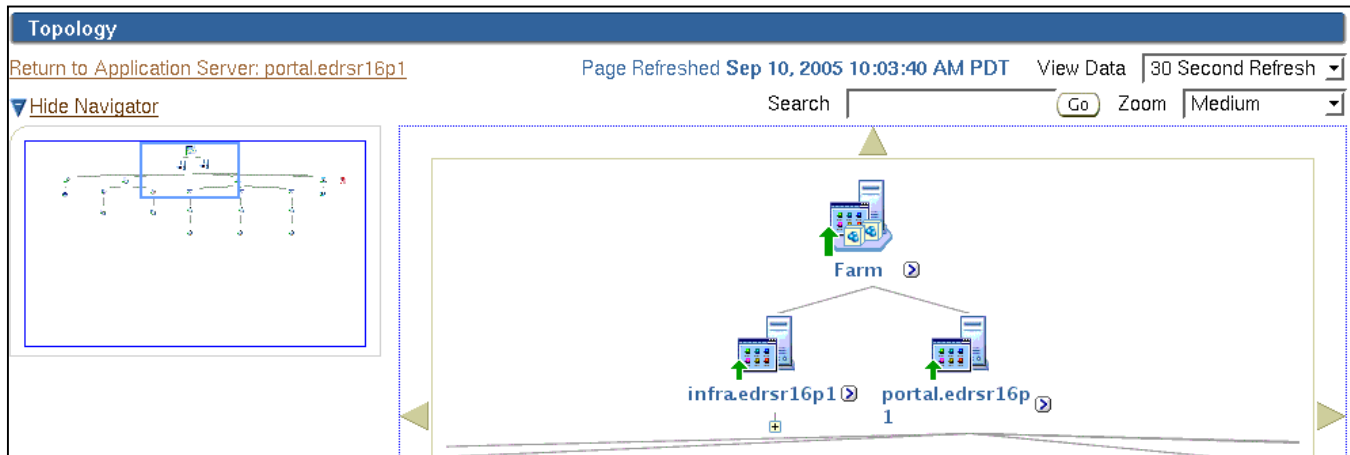
5. Using Application Server Control, access the middle-tier instance and check the status of the components.
 - a. Navigate to Application Server Control Farm page.
 - b. Click the OracleAS Portal instance link in the table of available instances of the Application Server Control Farm page. If requested, enter `ias_admin` and `welcome1` in the User Name and Password fields, respectively.
 - c. Verify that Oracle HTTP Server on the middle tier is running (Status is up). If not, select Oracle HTTP Server in the Systems Components table, and click Start.
6. Access the Oracle HTTP Server home page in the middle-tier instance. Stop the Oracle HTTP Server component and start it again.
 - a. From your middle-tier instance page of Application Server Control, click the HTTP Server link to open the Oracle HTTP Server home page.
 - b. Click Stop to stop Oracle HTTP Server.
 - c. Click Yes on the confirmation page, and then click OK.
 - d. Click Start to start Oracle HTTP Server. Click Yes on the confirmation page, and then click OK.

In this case, Oracle HTTP Server has been started.

7. View and monitor components with Topology Viewer.

Using the graphical, real-time viewer tool that Application Server Control offers, perform the following common administrative tasks:

- a. Access the Topology link from the Application Server Control Console of the portal instance.
 1. Open your browser, and enter the URL `http://<host name>.<domain>:<port>` for portal instance.
 2. Log in as `ias_admin/<admin password you specified during install>`. Click OK. The Application Server home page appears.
 3. Click the Topology link at the top of the page.



- c. Verify that your OC4J component is up.
 1. Click the down arrow on the Topology page to scroll down the topology.
 2. Verify that your OC4J component is up.
 - d. Scroll down the topology to see the process used by the OC4J instance and view real-time performance metrics, such as CPU Time (seconds) and Memory Usage (MB).
 1. Click the down arrow on the Topology page to scroll down the topology.
 2. Note the process used by the OC4J instance.
 3. Record CPU Time (seconds), Memory Usage (MB), and Requests Per Second.
 - e. From the Topology Viewer, click the icon displayed for the process, and navigate to the component home page. After viewing the component home page, click Back in the browser to return to the Topology page.
 1. Click the icon displayed for the process to navigate to the OC4J server home page.
 2. Click Back in the browser to go back to the Topology page.
 - f. Click the arrow icon (>) next to OC4J component, and select Collapse Node.
 - g. Change the refresh option to Manual Refresh and Zoom to Small.
 1. Select Manual Refresh from the View Data list.
 2. Select Small from the Zoom list.
8. View performance metric details.

You can obtain an overview of the application server availability and system resource usage for the server and the individual components from its home page. You can also view a detailed list of all performance metrics being monitored by Application Server Control. You can view the `mod_oc4j` metrics for the Oracle HTTP Server component. Change the refresh interval to Real Time: 30 Second Refresh to see a brief history of metric data.

- a. Navigate to the Oracle Application Server Control home page.
- b. Click the HTTP_Server link.
- c. Click the All Metrics link in the Performance section. The All Metrics page provides you with a list of performance metrics that Application Server Control monitors for Oracle HTTP Server.

- d. Click Expand All to expand all the metrics that are being monitored for Oracle HTTP Server.
 - e. Scroll down to the mod_oc4j General Metrics link.
 - f. Click Requests to OC4J Instances (per second).
 - g. Start an OC4J application in a new window. Open a new window, and enter the following URL:
`http://<host name>.<domain>:<Oracle HTTP Server Port>/IsWebCacheWorking`
 - h. Switch to the Application Server Control window. Change the refresh interval to every 30 seconds by selecting Real Time: 30 Second Refresh from the View Data drop-down list.
 - i. Observe the page while it refreshes. Notice that Application Server Control charts the value for the metric over time such that trends can be observed quickly and easily.
9. Change your Application Server Control Console port to 1812 using the `emctl` utility.
 - a. Open a terminal window and change the directory to the bin directory of portal instance:
`$ cd $HOME/portal/bin`
 - b. Stop Application Server Control using the following command:
`emctl stop iasconsole`
 - c. Use the following command to change the Application Server Control Console port to 1812:
`emctl config iasconsole port 1812`
 - d. Start Application Server Control:
`emctl start iasconsole`
- Change the directory to the bin directory of the portal instance. Change your Application Server Control Console port to 1810 by using the `emctl` utility before moving to the next practice.**
10. The Application Server Control Console enables you to list and search log files across Application Server components. View the log files for Oracle HTTP Server from the Application Server Control Console page.
 - a. On the Application Server home page, click the Logs link at the top of the page to view the log files for this application server.
 - b. Select HTTP_Server in the Available Components list, and click Move to transfer this component to Selected Components. Click Search.
 11. Obtain the status of your OracleAS Infrastructure instance by using the `opmnctl` command-line utility.

Open a shell window, navigate to your `$HOME/infra/opmn/bin` directory, and enter the `opmnctl status` command:

```
$ cd $HOME/infra/opmn/bin
$ ./opmnctl status
```

Processes in Instance: infra.edrsr16p1

ias-component	process-type	pid	status
DSA	DSA	N/A	Down
LogLoader	logloaderd	N/A	Down

dcm-daemon	dcm-daemon	12394	Alive
OC4J	OC4J_SECURITY	17795	Alive
HTTP_Server	HTTP_Server	17746	Alive
OID	OID	11961	Alive

12. Obtain the status of your Oracle Application Server middle-tier instance by using the `opmnctl` command-line utility.

Open a shell window, navigate to your `$HOME/portal/opmn/bin` directory, and run the `opmnctl status` command.

```
$ cd $HOME/portal/opmn/bin
$ ./opmnctl status
```

Processes in Instance: portal.edrsr16p1

ias-component	process-type	pid	status
DSA	DSA	N/A	Down
LogLoader	logloaderd	N/A	Down
dcm-daemon	dcm-daemon	3599	Alive
OC4J	home	9289	Alive
OC4J	OC4J_Portal	9290	Alive
OC4J	OC4J_Temp	15399	Alive
WebCache	WebCache	9269	Alive
WebCache	WebCacheAdmin	9252	Alive
HTTP_Server	HTTP_Server	9254	Alive

13. Stop your portal instance by using `opmnctl`, and stop Application Server Control.

```
$ ./opmnctl stopall
opmnctl: stopping opmn and all managed processes...
$ $HOME/portal/bin/emctl stop iasconsole
```

14. Start only the OPMN process, and verify the status of your portal instance.

```
$ cd $HOME/portal/opmn/bin
$ ./opmnctl start
opmnctl: opmn started
$ ./opmnctl status
```

Processes in Instance: portal.edrsr16p1

ias-component	process-type	pid	status
DSA	DSA	N/A	Down
LogLoader	logloaderd	N/A	Down
dcm-daemon	dcm-daemon	N/A	Down
OC4J	home	N/A	Down
OC4J	OC4J_Portal	N/A	Down
OC4J	OC4J_Temp	N/A	Down

WebCache	WebCache	N/A	Down
WebCache	WebCacheAdmin	N/A	Down
HTTP_Server	HTTP_Server	N/A	Down

15. Start the Oracle HTTP Server process in your portal instance.

```
$ ./opmnctl startproc ias-component=HTTP_Server
opmnctl: starting opmn managed processes...
$
```

16. Start your OC4J home component, and verify that you have started only home and Oracle HTTP Server.

```
$ ./opmnctl startproc process-type=home
opmnctl: starting opmn managed processes...
$ ./opmnctl status
```

Processes in Instance: portal.edrsr16p1

ias-component	process-type	pid	status
DSA	DSA	N/A	Down
LogLoader	logloaderd	N/A	Down
dcm-daemon	dcm-daemon	N/A	Down
OC4J	home	25334	Alive
OC4J	OC4J_Portal	N/A	Down
OC4J	OC4J_Temp	N/A	Down
WebCache	WebCache	N/A	Down
WebCache	WebCacheAdmin	N/A	Down
HTTP_Server	HTTP_Server	25255	Alive

17. Start up all the components of the portal instance. (Note that the DCM daemon does not start up immediately.)

```
$ ./opmnctl startall
opmnctl: starting opmn and all managed processes...
$ ./opmnctl status
```

Processes in Instance: portal.edrsr16p1

ias-component	process-type	pid	status
DSA	DSA	N/A	Down
LogLoader	logloaderd	N/A	Down
dcm-daemon	dcm-daemon	N/A	Down
OC4J	home	25334	Alive
OC4J	OC4J_Portal	25398	Alive
OC4J	OC4J_Temp	25399	Alive
WebCache	WebCache	25433	Alive
WebCache	WebCacheAdmin	25401	Alive
HTTP_Server	HTTP_Server	25255	Alive

Note that even though the DCM daemon is down, it is brought up when the first call to the `dcmctl` command is made.

18. Verify the memory used by the components of your middle tier.

```
$ ./opmnctl status -l
```

```
Processes in Instance: portal.edrsr16p1
```

ias-component					process-type	pid	status	uid
memused					uptime	p		
orts								
DSA					DSA	N/A	Down	N/A
N/A					N/A	N		
/A								
LogLoader					logloaderd	N/A	Down	N/A
N/A					N/A	N		
/A								
dcm-daemon					dcm-daemon	N/A	Down	N/A
N/A					N/A	N		
/A								
OC4J					home	25334	Alive	366018573
65876					00:05:29	a		
jp:12502,rmi:12402,jms:12602								
OC4J					OC4J_Portal	25398	Alive	366018574
134908					00:03:58	a		
jp:12503,rmi:12403,jms:12603								
OC4J					OC4J_Temp	25399	Alive	366018575
60712					00:03:58	a		
jp:12504,rmi:12404,jms:12604								
WebCache					WebCache	25433	Alive	366018576
33236					00:03:57	h		
ttp:7778,invalidation:9401,statistics:9402								
WebCache					WebCacheAdmin	25401	Alive	366018577
15360					00:03:58	a		
dministration:9400								
HTTP_Server					HTTP_Server	25255	Alive	366018572
105412					00:06:08	h		
ttp1:7779,http2:7201								

19. Back up the configuration of your portal instance to the `$HOME/labs/backup` directory by using the `createArchive` and `exportArchive dcmctl` commands. Use the following information:

- Name the archive: `portal-1`
- Location of the exported archive: `$HOME/labs/backup/arch-portal-1`

Open a shell window and create a backup directory in your `HOME/labs` directory if not available.

```
$ mkdir $HOME/labs/backup

$ cd $HOME/portal/dcm/bin
$ dcmctl createArchive -arch portal-1
$ dcmctl exportArchive -arch portal-1 -f $HOME/labs/backup/arch-portal-1
$ dcmctl listArchives -arch portal-1
```

20. Start Application Server Control for the portal instance:

```
$ $HOME/portal/bin/emctl start iasconsole
```

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Practice 6: Configuring and Managing Oracle HTTP Server

This practice demonstrates how to perform the basic configuration for your Oracle HTTP Server. It also demonstrates how to locate the appropriate configuration files on the operating system, and how to use Application Server Control or the appropriate `dcmctl` commands.

1. Verify that your Oracle HTTP Server of the middle tier is running. Start Oracle HTTP Server if it is not already running. (Use `opmnctl` to get the status of Oracle HTTP Server.)

```
$ cd $HOME/portal/opmn/bin
$ ./opmnctl status
```

Processes in Instance: portal.edrsr16p1

ias-component	process-type	pid	status
DSA	DSA	N/A	Down
LogLoader	logloaderd	N/A	Down
dcm-daemon	dcm-daemon	26796	Alive
OC4J	home	25334	Alive
OC4J	OC4J_Portal	25398	Alive
OC4J	OC4J_Temp	25399	Alive
WebCache	WebCache	25433	Alive
WebCache	WebCacheAdmin	25401	Alive
HTTP_Server	HTTP_Server	25255	Alive

If the status of Oracle HTTP Server is Down, then start it up using the command:

```
./opmnctl start ias-component=HTTP_Server
```

2. Check whether Application Server Control is running.

```
$ cd $HOME/portal/bin/
$ ./emctl status iasconsole
TZ set to US/Pacific
Oracle Enterprise Manager 10g Application Server Control Release
10.1.2.0.2
Copyright (c) 1996, 2005 Oracle Corporation. All rights reserved.
http://EDRSR16P1:1810/emd/console/aboutApplication
Oracle Enterprise Manager 10g Application Server Control is running.
-----
Logs are generated in directory /home/oracle/portal/sysman/log
$
```

3. Name the directory where Oracle HTTP Server for your middle-tier instance is installed, and locate the main configuration file.
 - a. Access your middle-tier instance home page by using Application Server Control.
 - b. Click HTTP Server.
 - c. Click the Administration tab.
 - d. Click the Server Properties link.
 - e. Note Server Root Directory and Configuration File.

4. Obtain the port number on which your Oracle HTTP Server is listening.
 - a. Access the Oracle HTTP Server page.
 - b. Click the Administration tab, and click Server Properties.
 - c. Locate the Listening Addresses and Ports table.
5. Enable your Oracle HTTP Server to listen on an additional port: 7785.
 - a. In the Listening Addresses and Ports table of the Server Properties page, click Add Another Row.
 - b. Enter an additional port number (7785) in the Listening Port column, and click Apply (bottom-right corner).
 - c. Confirm to restart Oracle HTTP Server to apply the changes, by clicking Yes in the Confirmation window.
 - d. Click OK.
6. Verify that the change is reflected in the httpd.conf file.

```
$ grep 7785 $HOME/portal/Apache/Apache/conf/httpd.conf
Listen 7785
$
```

7. Ensure that you have the index.html file in your \$HOME/labs directory. Then, change your default document root directory to refer to /home/oracle/labs instead of htdocs. To test your success, enter the following URL:
 http://<host name>.<domain>:<Oracle HTTP Server port of portal instance>

Farm > Application Server: portal.edrsr16p1 > HTTP Server >

Server Properties

General

Version	10.1.2
Server Root Directory	/home/oracle/portal/Apache/Apache
Configuration File	/home/oracle/portal/Apache/Apache/conf/httpd.conf
Process ID File	/home/oracle/portal/Apache/Apache/logs/httpd.pid
* Document Root	/home/oracle/labs
Administrator E-Mail	you@your.address
User	oracle
Group	dba

- a.

```
$ ls $HOME/labs/ind*
/home/oracle/labs/index.html
$
```
- b. Access the Server Properties page of Oracle HTTP Server in Application Server Control.
- c. Edit Document Root to be /home/oracle/labs. Click Apply.
- d. Click Yes to restart Oracle HTTP Server.
- e. Click OK.

8. Invoke another browser window, clear the browser cache, and then access Oracle HTTP Server with the following URL: `http://<host name>.<domain>:<Oracle HTTP Server port of portal instance>`

You receive a Welcome message for accessing the HTTP Server port of the portal instance.

9. Change the document root back to the original setting. DocumentRoot should now point to `/home/oracle/portal/Apache/Apache/htdocs` before you start the next practice.
 - a. Change the Document Root field to `/home/oracle/portal/Apache/Apache/htdocs`.
 - b. Click Apply to save your changes, and restart your Oracle HTTP Server.
10. When you try to access the URL `http://<host name>.<domain>:<Oracle HTTP Server port>/`, which file is served by default?

The `index.html` file is served unless the `DirectoryIndex` directive explicitly points to another file.

11. Reconfigure Oracle HTTP Server so that it does not listen on port 7785.
 - a. In the Listening Addresses and Ports table of the Server Properties page, select the row with Listening Port 7785, and click Remove.
 - b. Click Apply (bottom-right corner).
 - c. Confirm to restart Oracle HTTP Server to apply the changes, by clicking Yes in the Confirmation window.
 - d. Click OK.

Listening Addresses and Ports	
Default Port 7778	
Select Item and... Remove	
Select All Select None	
Select	Listening IP Address
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	127.0.0.1
<input type="checkbox"/>	
Add Another Row	

Practice 7: Configuring Directives and Virtual Hosts

1. Add two entries (*<IP ADDRESS of your machine>* mymachine.us.oracle.com and *<IP ADDRESS of your machine>* mymachine1.us.oracle.com) in the hosts file to create name-based virtual hosts.

Add an entry in the hosts file:

- a. Open a new terminal and log in as root.
- b. Navigate to the /etc directory.
- c. Open the hosts file by using the following command:
`$ gedit hosts`
Ignore any error messages that appear.
- d. Do not remove any lines in this file. Append the following entries to the end of the file:
<IP ADDRESS of your machine> mymachine.us.oracle.com
<IP ADDRESS of your machine> mymachine1.us.oracle.com
For example: 139.185.35.125 mymachine.us.oracle.com
139.185.35.125 mymachine1.us.oracle.com
- e. Click Save, and exit the file.

Create file systems to serve the host:

Note: Change to the other terminal window to perform the following steps as the oracle user.

- a. `$ cd $HOME`
- b. `$ mkdir websites`
- c. `$ cd websites`
- d. `$ mkdir mysite`
- e. `$ mkdir mysitel`
- f. `$ cd mysite`
- g. `$ mv $HOME/labs/index7.html $HOME/websites/mysite/index.html`
- h. `$ mv $HOME/labs/index7a.html $HOME/websites/mysitel/index.html`

2. Enable your HTTP Server to listen on two additional ports—7800 and 7801.
 - a. Access your middle-tier instance home page by Application Server Control.
 - b. Click HTTP Server.
 - c. Click the Administration tab.
 - d. Click the Server Properties link.
 - e. In the Listening Addresses and Ports table of the Server Properties page, click Add Another Row.
 - f. Enter an additional port number (7800) in the Listening Port column, and click Apply (bottom-right corner).
 - g. Confirm to restart Oracle HTTP Server to apply the changes, by clicking Yes in the Confirmation window.
 - h. Click OK.
 - i. Repeat the above steps to add the second listening port number (7801).

3. Create a virtual host by using Application Server Control.

- a. Click the Virtual Hosts tab in the HTTP_Server home page.
- b. Click Create to create a new virtual host by using the wizard.
- c. To navigate to the General page, click Next.
- d. Notice that the Document Root Directory field is set to default as /home/oracle/portal/Apache/Apache/htdocs. Change this value to /home/oracle/websites/mysite. Verify that the Virtual Host Type is name based and then click Next.
- e. In the Server Name field, enter mymachine.us.oracle.com. Click Next.
- f. Select 7800 from the “Listen on a specific port” list. Click Next.

Enter the server name, server aliases, and IP address to be used with this name-based virtual host.


Server Name and Aliases

* Server Name

Select row and...

[Select All](#) | [Select None](#)

Select Server Alias

 **TIP** Values entered for Server Name and Server Alias should be valid DNS names. If you set name Server Aliases include www.name1.mydomain.com and name1.

- g. Click Next on the Create Virtual Host: Error Log page.
 - h. Review the information, and click Finish on the Create Virtual Host: Summary page.
 - i. Click Yes to save the configuration changes, and restart Oracle HTTP Server.
 - j. Click OK on the Confirmation page.
 - k. When you return to the HTTP_Server page, the new virtual host is now listed.
 - l. Select the new virtual host, and click Create Like.
 - m. Enter mymachine1.us.oracle.com in the Server Name field.
 - n. In the Document Root Directory field, enter /home/oracle/websites/mysite1.
 - o. Select 7801 from the “Listen on a specific” port list.
 - p. Click Create.
 - q. Click Yes to save the configuration changes, and restart Oracle HTTP Server.
 - r. Click OK on the Confirmation page.
 - s. When you return to the HTTP_Server page, the new virtual host is now listed.
4. Note the changes in the httpd.conf file.

- a. Click Administration.
- b. Click Advanced Server Properties.
- c. Click httpd.conf under Configuration Files.
- d. Scroll down to the end of the file. You can view the additional code created by the wizard in the httpd.conf file:

```
<VirtualHost *:7800>
    ServerName mymachine.us.oracle.com
    DocumentRoot "/home/oracle/websites/mysite"
```

```
</VirtualHost>

<VirtualHost *:7801>
    ServerName mymachine1.us.oracle.com
    DocumentRoot "/home/oracle/websites/mysite1"
</VirtualHost>

Listen 7800
Listen 7801
```

5. Access the new virtual host.

Make the following changes to your proxy settings:

- a. In your browser, select Edit > Preferences > Advanced > Proxies.
- b. Enter mymachine.us.oracle.com and mymachine1.us.oracle.com in the "No Proxy for" field.
- c. Open a new browser, and enter the following URL:
http://mymachine.us.oracle.com:<7800>
- d. Open a new browser, and enter the following URL:
http://mymachine1.us.oracle.com:<7801>

Practice 8: Configuring and Managing OracleAS Web Cache

This practice demonstrates the tasks in administering OracleAS Web Cache.

1. Stop and start OracleAS Web Cache by using `opmnctl`.

```
$ cd $HOME/portal/opmn/bin
$ ./opmnctl status
$ ./opmnctl stopproc ias-component=WebCache (if OracleAS Web Cache is up)
$ ./opmnctl startproc ias-component=WebCache (if OracleAS Web Cache is Down)
```
2. Change the OracleAS Web Cache administrator password to `oracle1`.
 - a. Invoke Application Server Control, and navigate to your portal instance.
 - b. Click the Web Cache link in the System Components table.
 - c. Click the Web Cache Administration link on the Web Cache home page.
 - d. Click Security in the Properties section to invoke the Security page.
 - e. Enter the old password (`welcome1`) in the Old Password field in the Administrator User Password section. Enter the new password (`oracle1`), confirm the same, and click OK.
 - f. Do NOT restart Web Cache in the Confirmation window.
 - g. Clean your browser history, cache, and cookie:
In your Mozilla or Netscape Browser window, select
 - Edit > Preferences > Navigator > History > Clear History, and Clear Location Bar
 - Edit > Preferences > Advanced > Cache > Clear Cache
 - Edit > Preferences > Privacy & Security > Cookies – Click Manage Stored Cookies. The cookie manager window appears. Click Remove All Cookies.
 - h. Click Restart Web Cache to restart OracleAS Web Cache using Application Server Control. Click Yes when asked for confirmation.
 - i. Enter the following URL:
`http://<host name>.<domain>:<Web Cache Manager port>`. You can obtain the Web Cache Manager port from the ports page of the Application Server Control Console. (The Web Cache Manager port is usually 9400.)
Click OracleAS Web Cache Manager in the Managing Web Cache section. Enter `ias_admin` in the User Name field and `oracle1` in the Password field to log in to Web Cache Manager.
3. Add a listening port for OracleAS Web Cache.
 - a. Invoke the Application Server Control Console of your Portal instance.
 - b. Click Web Cache in the System Components table.
 - c. Click the Administration tab on the Web Cache home page.
 - d. Scroll down to the Properties region, and click Ports. The Ports page appears.
 - e. In the Listen Ports table, click Add a row.
 - f. Enter the following values on the Listen Ports page:
IP Address: * Port Number: 7890 Protocol: HTTP
 - g. Click OK, and restart Web Cache. Click Yes.

4. Add a Site to Server mapping for the newly added port.
 - a. On the OracleAS Web Cache Administration page, scroll down to the Properties region.
 - b. Click Sites.
 - c. On the Sites page, in the Named Sites Definitions section, click Create .
 - d. Enter the host name and the additional port number (7890).

Create Named Site		
Create a named site to define site-specific caching rules, error pages, logs, sessions, named sites.		
<div> <div>General</div> <div>Advanced</div> </div>		
* Host	edrsr16p1.us.oracle.com	Prefix
	(Example: www.company.com. Do not use wildcards.)	
* Port	7890	

- e. Select the Application Web Servers (<your host name:7779>) to which the additional Web Cache port should be mapped. Click Move. The Application Web Server is now listed under Selected Origin Servers.
 - f. Click the Advanced tab.
 - g. Select “Enable session binding for this site.” Verify that Cookie-Based is selected in the Session Binding Mechanism list.
 - h. Click OK.
 - i. Click Restart Web Cache.
5. Clear your browser cache and history in the Application Server Control browser window:

In your Mozilla or Netscape Browser window, select:

 - Edit > Preferences > Navigator > History > Clear History, and Clear Location Bar
 - Edit > Preferences > Advanced > Cache > Clear Cache
6. Clear the OracleAS Web Cache contents. Use the basic content invalidation mechanism.
 - a. On the Application Server Control Web Cache Administration page, click Invalidation in the Operations region.
 - b. On the Invalidation page, from the “Invalidate these cached objects” list, select All objects. Click Next.
 - c. On the Invalidation: Removal Time page, from the “Specify when to remove objects from the cache” list, select “Remove objects immediately.” Refresh each object as soon as there is a browser request for it. Click Next.
 - d. Note the details on the Invalidation: Review page. Click Finish.
 - e. Click OK on the Invalidation Result page.
7. Access the Oracle HTTP Server Welcome page directly by entering the URL `http://<host name>.<domain>:<Oracle HTTP Server Listen port of Portal instance>`, and verify that the page is not cached.
 - a. Open a new browser, and enter the following URL:
`http://<host name>.<domain>:<Oracle HTTP Server Listen port of portal instance>`

- b. After accessing the Welcome Page in a separate window, click the Popular Requests link on the Web Cache Performance page of the Application Server Control Console.
 - c. The URL for the Welcome page is not listed in Popular Requests.
8. Access the Oracle HTTP Server Welcome page by using the new Web Cache (port 7890), and verify that the new access has been cached.
9. Using Application Server Control, view the Web Cache Statistics now.

Navigate to the Web Cache Performance page to view the statistics.

10. Using Application Server Control, disable caching for .swf files.
- a. Navigate to the Application Server Control Web Cache Administration page.
 - b. Click Rules in the Properties section.
 - c. On the Rules page, select the row with Name as cache swf, and click Edit. The Edit Rule page appears.
 - d. In the Instructions section, select “Do not cache.” Click OK.
 - e. Click Restart Web Cache, and click Yes to effect changes in the caching rule.
11. Verify that the caching rule reflects the change you have made.
Access the Rules page again, and verify the row for .swf.
12. Delete the Site to Server mapping you added.
- a. Navigate to the Application Server Control Web Cache Administration page.
 - b. In the Properties section, click Sites.
 - c. On the Sites page, select the newly added mapping that corresponds to port 7890.
 - d. Click Delete and confirm the deletion.
 - e. Click Restart Web Cache.

13. Delete the Additional port you added.

- a. Navigate to the Web Cache Administration page.
- b. Under Properties section, click Ports.
- c. On the Ports page, click Delete. Click OK to delete the newly added port: port 7890.

14. Re-enable caching for files with the extension .swf.

- a. Navigate to the Application Server Control Web Cache Administration page.
- b. Click Rules in the Properties section.
- c. On the Rules page, select the row with Name as cache swf, and click Edit. The Edit Rule page appears.
- d. In the Instructions section, select Cache. Click OK.
- e. Click Restart Web Cache to effect the changes in the caching rule.

Practice 9: Configuring and Managing OC4J

This practice demonstrates how to access OC4J home pages; identify the different OC4J instances running; and start, stop, and restart instances.

1. Start the Application Server Control of the Portal instance if it is not running.

```
$ cd $HOME/portal/bin
```

```
$ ./emctl status iasconsole
```

```
$ ./emctl start iasconsole
```

2. Navigate to the home OC4J component home page. Note the default application properties.

- a. Navigate to your Portal Instance home page.
- b. Click home to open the OC4J: home page.
- c. Click the Administration tab, and click the Server Properties link.

3. List the range of ports that can be used for the communication between the OC4J home component and Oracle HTTP Server.

On the Server Properties page under the Administration tab of OC4J:home, scroll down to the Ports section to view the range of ports.

4. Create a new OC4J component named my_OC4J. Which directories and files are created?

- a. Navigate to the Portal Instance home page.
- b. Click Create OC4J Instance in the System Components table.
- c. Enter the name my_OC4J, and click Create.
- d. You will receive a confirmation page indicating that your OC4J instance has been created. Click OK.
- e. Access the my_OC4J page, click the Administration tab, and click the Server Properties link.
- f. Note the directories and files that are created.

5. Start the newly created OC4J instance.

- a. Navigate to your Portal Instance home page.
- b. Select your newly created my_OC4J instance from the System Components table, and click Start.

6. Stop and delete the my_OC4J OC4J instance.

- a. From your Portal Instance home page, scroll down to the System Components table.
- b. Select the my_OC4J instance, and click Stop. Click Yes to confirm stopping the OC4J instance.
- c. Select my_OC4J, and click Delete OC4J Instance. Click Yes to confirm the deletion. Click OK.

Practice 10: Deploying Java 2, Enterprise Edition (J2EE) Applications

This practice demonstrates how to configure and deploy various types of J2EE applications, a Web application, and a J2EE (EAR) application to Oracle Application Server, and how to inspect the directories and files that are automatically created when the applications are deployed.

1. Set your Oracle environment by using `$HOME/labs/set_infra_env.sh`. Set up the database with the `USERS` tablespace.
 - a. Verify that the `USERS` tablespace exists in your `infra` database.
 - b. If the `USERS` tablespace does not exist, create the `USERS` tablespace with 20 MB size. Execute the `create_users.sql` script as the `SYS` or `SYSTEM` user.

```
$ . $HOME/labs/set_infra_env.sh
$ cd $HOME/labs
$ sqlplus "/ as sysdba"
SQL> select * from dba_tablespaces where
tablespace_name='USERS';
If the USERS tablespace does not exist, create the USERS tablespace.
SQL> @create_users.sql
SQL> exit
```

2. Create the `ora01` user and the necessary tables in the `ora01` schema. (Use the `hr_main.sql` script in the `$HOME/labs/HR_Setup` directory. This script creates the `ora01` user and tables for `hrapp`.)

```
$ cd $HOME/labs
$ unzip HR_Setup.zip
$ cd $HOME/labs/HR_Setup
$ sqlplus "/ as sysdba"
SQL> @hr_main.sql
SQL> exit
```

3. Deploy the provided JSP files: `login.jsp` and `error.jsp`. Access the JSPs.
 - a. The `login.jsp` and `error.jsp` files are available in `$HOME/labs`. Change the directory to `labs`.
`cd $HOME/labs`
 - b. Copy the two files to the `$HOME/portal/j2ee/home/default-web-app` directory.
`cp login.jsp $HOME/portal/j2ee/home/default-web-app`
`cp error.jsp $HOME/portal/j2ee/home/default-web-app`
 - c. Access the deployed `login.jsp` by using the following URL:
`http://<host name>.<domain>:<Oracle HTTP Server port of Portal instance>/j2ee/login.jsp`.
(for example, `http://edrsr16p1.us.oracle.com:7778/j2ee/login.jsp`)
 - d. If you enter any value and click the Login button, you should get a Login Error screen.

4. Examine the directory:
\$HOME/portal/j2ee/home/application-deployments/default/defaultWebApp/
persistence/_pages. What does it contain and why?

\$ cd
\$HOME/portal/j2ee/home/application-deployments/default/defaultWebApp/
persistence/_pages
\$ ls
It contains the generated servlets for login.jsp and error.jsp. This directory is the
default location for deployed applications.
5. Create and start an OC4J component named my_OC4J in your Portal instance.
 - a. Invoke the Application Server Control Console. Click the Portal instance to access your Portal instance home page.
 - b. Click Create OC4J Instance to access the Create OC4J Instance page.
 - c. Enter the OC4J Instance name (my_OC4J), and then click Create. Confirm the creation of OC4J instance. Click OK.
 - d. Select my_OC4J, and click Start to start my_OC4J instance.
6. Deploy a simple Web Application Archive (WAR) file to the my_OC4J component.
 - a. Invoke the Application Server Control Console and access the Portal instance.
 - b. Click my_OC4J in the System Components table to access the my_OC4J home page.
 - c. Click Applications to access the Applications page.
 - d. On the Applications property page, click Deploy WAR file.
 - e. In the Deploy Web Applications window, enter the details as follows:
 - Web Application: Click Browse and locate the lab10-web.war file in the \$HOME/labs directory.
 - Application Name: lab10-web
 - Map to URL: /lab10-weband then click Deploy. When asked for confirmation, confirm the deployment.

Deploy Web Application

Select the Web Application (.war file) you wish to deploy. This web application will be wrapped into a J2EE application (.ear file) before deployment.

Web Application

Specify the name you would like this application to be called and the URL to map this web application to.

Application Name

Map to URL

7. Using Application Server Control, verify that the application has indeed been deployed.

Invoke the Lab10 application by using the URL `http://<host name>.<domain>:<Oracle HTTP Server port of Portal instance>/lab10-web`.

8. Where do you expect to find the file relating to the deployed application?

In the \$HOME/portal/j2ee/my_OC4J directory

9. Where do you find the server.xml file for this application?

In the \$HOME/portal/j2ee/my_OC4J/config directory

10. Where would you find the orion-application.xml file for this application?

In the \$HOME/portal/j2ee/my_OC4J/application-deployments/lab10-web directory

11. Where do you find the lab10-web.ear file?

An EAR file is created automatically when a WAR file is deployed by using Application Server Control. It is located in the \$HOME/portal/j2ee/my_OC4J/applications directory.

Deploying a J2EE Application (EAR)

12. Examine the mod_oc4j.conf, server.xml, and default-web-site.xml files.

```
$ cat $HOME/portal/Apache/Apache/conf/mod_oc4j.conf
```

```
#####  
# Oracle iAS mod_oc4j configuration file: mod_oc4j.conf #  
#####  
LoadModule oc4j_module libexec/mod_oc4j.so
```

```
<IfModule mod_oc4j.c>
```

```
<LocationMatch "/j2ee/./Spy">  
    Order deny,allow  
    Deny from all  
</LocationMatch>
```

```
<LocationMatch "/j2ee/./AggreSpy">  
    Order deny,allow  
    Deny from all  
</LocationMatch>
```

```
<Location /oc4j-service>  
    SetHandler oc4j-service-handler  
    Order deny,allow  
    Deny from all  
    Allow from localhost edrsr16p1 edrsr16p1  
</Location>
```

```
Oc4jMount /j2ee/*  
Oc4jMount /portalHelp2 OC4J_Portal  
Oc4jMount /portalHelp2/* OC4J_Portal  
Oc4jMount /pdkstruts OC4J_Portal  
Oc4jMount /pdkstruts/* OC4J_Portal
```

```

Oc4jMount /jpdck OC4J_Portal
Oc4jMount /jpdck/* OC4J_Portal
Oc4jMount /ultrasearch OC4J_Portal
Oc4jMount /ultrasearch/* OC4J_Portal
Oc4jMount /ultrasearch/query OC4J_Portal
Oc4jMount /ultrasearch/query/* OC4J_Portal
Oc4jMount /ultrasearch/admin OC4J_Portal
Oc4jMount /ultrasearch/admin/* OC4J_Portal
Oc4jMount /ultrasearch/admin_sso OC4J_Portal
Oc4jMount /ultrasearch/admin_sso/* OC4J_Portal
Oc4jMount /ultrasearch/ohw OC4J_Portal
Oc4jMount /ultrasearch/ohw/* OC4J_Portal
Oc4jMount /uddi OC4J_Portal
Oc4jMount /uddi/* OC4J_Portal
Oc4jMount /portalHelp OC4J_Portal
Oc4jMount /portalHelp/* OC4J_Portal
Oc4jMount /portalTools/webClipping OC4J_Portal
Oc4jMount /portalTools/webClipping/* OC4J_Portal
Oc4jMount /portalTools OC4J_Portal
Oc4jMount /portalTools/* OC4J_Portal
Oc4jMount /portalTools/builder OC4J_Portal
Oc4jMount /portalTools/builder/* OC4J_Portal
Oc4jMount /portalTools/sample OC4J_Portal
Oc4jMount /portalTools/sample/* OC4J_Portal
Oc4jMount /portalTools/omniPortlet OC4J_Portal
Oc4jMount /portalTools/omniPortlet/* OC4J_Portal
Oc4jMount /provider/ultrasearch OC4J_Portal
Oc4jMount /provider/ultrasearch/* OC4J_Portal
Oc4jMount /portal OC4J_Portal
Oc4jMount /portal/* OC4J_Portal
Oc4jMount /uddirepl OC4J_Portal
Oc4jMount /uddirepl/* OC4J_Portal
Oc4jMount /portletapp home
Oc4jMount /portletapp/* home
Oc4jMount /webapp home
Oc4jMount /webapp/* home
Oc4jMount /IsWebCacheWorking home
Oc4jMount /IsWebCacheWorking/* home
Oc4jMount /lab10-web OC4J_Temp
Oc4jMount /lab10-web/* OC4J_Temp
</IfModule>

```

```
$ cat $HOME/portal/j2ee/my_OC4J/config/server.xml
```

```

<?xml version="1.0"?>
<!DOCTYPE application-server PUBLIC "-//Oracle//DTD OC4J Application-
server 9.04//EN" "http://xmlns.or
acle.com/ias/dtds/application-server-9_04.dtd">

<application-server localhostIsAdmin="true"
  application-directory="../applications"
  deployment-directory="../application-deployments"
  connector-directory="../connectors"

```



```

>
    <rmi-config path="./rmi.xml" />
    <sep-config path="./internal-settings.xml" />
    <jms-config path="./jms.xml" />
    <javacache-config path="../../../javacache/admin/javacache.xml"
/>

    <j2ee-logging-config path="./j2ee-logging.xml" />
    <log>
        <file path="../../log/server.log" />
    </log>
    <transaction-config timeout="30000" />
    <java-compiler name="javac" in-process="false"
extdirs="/home/oracle/portal/jdk/jre/lib/ext" /
>
    <global-application name="default" path="application.xml" />
    <application name="lab10-web" path="../../applications/lab10-
web.ear" auto-start="true" />
    <global-web-app-config path="global-web-application.xml" />
    <web-site default="true" path="../../default-web-site.xml" />
    <cluster id="385006397" />
</application-server>

```

```
$ cat $HOME/portal/j2ee/my_OC4J/config/default-web-site.xml
```

```

<?xml version="1.0"?>
<!DOCTYPE web-site PUBLIC "-//Oracle//DTD OC4J Web-site 9.04//EN"
"http://xmlns.oracle.com/ias/dtds/web-site-9_04.dtd">

<web-site port="12504" protocol="ajp13" display-name="OracleAS Java Web
Site">
    <default-web-app application="default" name="defaultWebApp"
root="/j2ee" />
    <web-app application="default" name="dms" root="/dmsoc4j" access-
log="false" />
    <web-app application="lab10-web" name="lab10-web" load-on-
startup="true" root="/lab10-web" />
    <access-log path="../../log/default-web-access.log" />
</web-site>

```

13. Deploy the hrapp.ear application to my_OC4J. The hrapp.ear file is located in your \$HOME/labs directory.

- Invoke the my_OC4J home page, navigate to Applications property page, and click Deploy EAR File.
- The Deploy Application window appears. Enter or select the following values:
J2EE Application: Click Browse and select hrapp.ear from the /home/oracle/labs directory
Application Name: hrapp
Parent Application: Default
Click Continue.

- c. On the URL Mappings for Web Modules page, the URL mapping should be /hrapp. Click Next.
 - d. The Resource Reference Mappings page appears. In the Data Sources for CMP Entity Beans table, the Data Source field for the Employees and Departments Entity Bean should be jdbc/hrDS. Click Finish.
 - e. The Review page should appear. Click Deploy.
14. After the application has been deployed, verify that hrapp appears in the Deployed Applications list. Create a Data Source with the following information at the application level:
- Name: hrDS
 - Data Source Class: com.evermind.sql.DriverManagerDataSource
 - JDBC URL: jdbc:oracle:thin:@<hostname>.<domain>:1521:infra (point to your OracleAS Metadata Repository) If the database connection fails, try the JDBC URL with the domain name such as
jdbc:oracle:thin:@<hostname>.<domain>:1521:infra.<domain>.
 - JDBC Driver: oracle.jdbc.driver.OracleDriver
 - Username: ora01
 - Select Use Cleartext Password
 - Password: oracle
 - Location: jdbc/hr
 - Transactional(XA) Location: jdbc/xa/hrXADS
 - EJB Location: jdbc/hrDS
 - Connection Retry Interval (seconds): 1
 - Cached Connection Inactivity Timeout (seconds): 30
- a. Invoke the my_OC4J home page.
 - b. Select hrapp in the Deployed Applications table, and click Edit. Click Data Sources in the Resources section.
 - c. Click Create.
 - d. In the Create Data Source page, enter the following values in the fields:

Create Data Source	
Use this page to configure a data source to connect to Oracle or non-Oracle databases. To connect to Oracle (pure Oracle) Data Source or an emulated (wrappers around Oracle Data Sources) Data Source. To connect to non-Oracle Data Source, select the appropriate Data Source Class. To connect to Oracle Data Source, select com.evermind.sql.DriverManagerDataSource with the Merant JDBC drivers. Please refer to the online help for more information.	
General	
* Name	hrDS
Description	
* Data Source Class	com.evermind.sql.DriverManagerDataSource
JDBC URL	jdbc:thin@edrsr16p1.us.oracle.com:1521:infra.us.oracle.com
JDBC Driver	oracle.jdbc.driver.OracleDriver
	<small>This field is required if you are using a generic Orion Data Source Class.</small>
Schema	

JNDI Locations

For an emulated Data Source, please specify all three location attributes. It is recommended that you reference the EJB Location attribute in your code to look up this Data Source. For a non-emulated Data Source, the location attribute is all that is needed.

* Location	jdbc/hr
Transactional(XA) Location	jdbc/xa/hrXADS
EJB Location	jdbc/hrDS

For emulated data sources, retrieve the data source using the JNDI value in this field.

Connection Attributes

Connection Retry Interval (seconds)	1
Max Connection Attempts	
Cached Connection Inactivity Timeout (seconds)	30

- e. Scroll down to the bottom of the page. Click Create to create the data source. When prompted to restart the OC4J_home instance, click Yes. Then, click OK. The new hrDS data source should appear in the Data Sources list.

15. Observe the data-sources.xml configuration file in the \$HOME/portal/j2ee/my_OC4J/config directory:
- ```
cd $HOME/portal/j2ee/my_OC4J/config
cat data-sources.xml
```

16. Observe the server.xml and default-web-site.xml files in the \$HOME/portal/j2ee/my\_OC4J/config directory.

- cd \$HOME/portal/j2ee/my\_OC4J/config
- cat server.xml
- cat default-web-site.xml

You can observe entries corresponding to the recently deployed hrapp application in these files. Also, you will find the EAR file and the corresponding directory for the hrapp application in the \$HOME/portal/j2ee/my\_OC4J/applications directory.

17. On the hrapp application home page, you should be able to see the component modules of the application: Web Module hrweb and EJB Module hrejb.

18. Enter the following URL to run the HR application from the browser:
- `http://<host name>.<domain>:<Oracle HTTP Server port of portal instance>/hrapp/.`
- You should see the application's welcome page.

19. Click Departments on the Welcome page to list all departments in the HR database.

20. Select a department on the Department list page to list all the employees in that department, or click Employees on the Welcome page to list all employees in the HR database. You can also search employees by name. Click Search on the Welcome page, enter a first name (for example, John), and click Search to list all employees with that first name.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable  
license to use this Student Guide.

## Practice 12: Configuring Oracle Application Server Components in Oracle Internet Directory

This practice demonstrates how to start and stop, and to search Oracle Internet Directory.

1. Match the following:

| Command       | Description                                                                                         |
|---------------|-----------------------------------------------------------------------------------------------------|
| a. bulkload   | 1. Backs up directory data                                                                          |
| b. ldapadd    | 2. Deletes a subtree                                                                                |
| c. ldifwrite  | 3. Loads one or more entries by using the standard I/O                                              |
| d. bulkdelete | 4. Loads a large number of entries to Oracle Internet Directory server by using LDIF files as input |

Solution: a-4, b-3, c-1, d-2

2. Verify that the ORACLE\_SID and ORACLE\_HOME environment variables are set. Run the \$HOME/labs/set\_infra\_env.sh script if they have not been already set.

```
$ echo $ORACLE_HOME
/home/oracle/infra
$ echo $ORACLE_SID
infra
$
```

The environment variables have been set. You do not have to run the script.

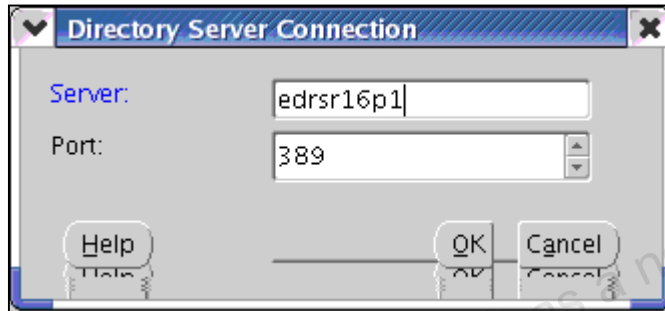
3. Using OPMN, verify that the Oracle Internet Directory server is running.

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

```
Processes in Instance: infra.edrsr16p1
```

| ias-component |               | process-type |       | pid | status |
|---------------|---------------|--------------|-------|-----|--------|
| DSA           | DSA           | N/A          | Down  |     |        |
| LogLoader     | logloaderd    | N/A          | Down  |     |        |
| dcm-daemon    | dcm-daemon    | 12394        | Alive |     |        |
| OC4J          | OC4J_SECURITY | 17795        | Alive |     |        |
| HTTP_Server   | HTTP_Server   | 17746        | Alive |     |        |
| OID           | OID           | 11961        | Alive |     |        |

4. Connect to an Oracle Internet Directory server by using Oracle Directory Manager (ODM).
  - a. From the command prompt, start ODM by using the command `./oidadmin (/home/oracle/infra/bin/)`. This displays the Directory Server Connection dialog box when you start ODM for the first time. Click OK and continue.
  - b. The Directory Server Name Manager dialog box appears. Click Add.
  - c. The Directory Server Connection dialog box appears. Enter the server (your server name) and port. (To verify the port, navigate to the Infrastructure property page of the Infrastructure instance by using the Application Server Control Console.) Click OK.



- d. The new Oracle Internet Directory server is added to the list of Oracle Internet Directory servers that you can connect to.
  - e. Select the Oracle Internet Directory server you have added, and click OK. The Oracle Directory Manager Connect dialog box appears with the Oracle Internet Directory server and the port selected. Enter User: `orcladmin`, and Password: `welcome1`; and click Login to log in to ODM.
  - f. ODM appears.
5. ODM displays the navigation tree, menu bar, and toolbar. When you click any of the node names in the navigation tree, its description is displayed in the right pane.
6. Get the password for the Portal Schema. The Portal Schema entry is in this search path:  
EntryManagement > cn=OracleContext > cn=Products > cn=IAS > cn=IAS  
Infrastructure Databases > orclReferenceName=infra.us.oracle.com  
>OrclResourceName=PORTAL. **Note orclpasswordattribute in the right pane.**
7. Close the ODM interface:  
Select File > Exit.
8. List the components of Enterprise Identity Management.

The various components of Enterprise Identity Management are:

- a. Directory Server
- b. Single Sign-On
- c. Certificate Authority
- d. Delegated Administrative Service
- e. Directory Integration
- f. User Provisioning

9. List the various directory roles.

The various directory roles are:


- a. Oracle Internet Directory administrator
- b. Domain or Subscriber administrator
- c. Application specific

10. A user should belong to the \_\_\_\_\_ group to configure Oracle Application Server components.

Answer: iASAdmins

11. Create a user by using ODM.

To create a user in Oracle Application Server by using ODM, perform the following steps:

- a. Set your Oracle environment by using the `$HOME/labs/set_infra_env.sh` script.
- b. Use `oidadmin` to start ODM and connect as an Oracle Internet Directory administrator:  
`cd $HOME/infra/bin`  
`./oidadmin`
- c. Navigate to and expand the Entry Management node until the `cn=Users` node appears under `dc=com,dc=oracle,dc=us`.
- d. Select the `cn=PUBLIC` node under `cn=Users`. Right-click to see the pop-up menu options. Select **Create Like** from the menu options, to create a new user.
- e. The New Entry dialog box appears. This dialog box already contains the values for the `cn=PUBLIC` user. Replace the following fields with these values:
  - a. Distinguished Name: `cn=newOIDuser,cn=Users,dc=us,dc=oracle,dc=com`
  - b. In Mandatory Properties:  
`cn=newOIDuser`  
`sn=newOIDuser`  
In Optional Properties:  
`employeenumber=newOIDuser`  
`givenname=newOIDuser`  
`mail=newOIDuser@host.com`  
`orclIsEnabled=Delete the existing value and leave it empty.`  
`uid=newOIDuser`  
`userpassword=newOIDuser1`
- f. Click OK when done. Refresh `cn=Users` to see the new user created.
- g. You can test the creation of the `newOIDuser` user by logging in to ODM as `newOIDuser`. In the Oracle Directory Manager Connect dialog box, click the  button, next to User field. In the Select Distinguished Name (DN) Path: All Entries dialog box, navigate to `dc=com > dc=oracle > dc=us > cn=Users > cn=newOIDuser`. Click OK. Enter `newOIDuser1` in the Password field. Click Login.

12. Grant the newOIDuser user the privilege to create new users.

To grant newOIDuser the privilege to create new users, perform the following steps:

- a. Start ODM and connect as an Oracle Internet Directory administrator.
- b. Navigate to and expand the Entry Management node. Expand the cn=OracleContext node. Expand the cn=Groups node, and select the cn=OracleDASCreateUser node.
- c. The right pane displays the properties of the entry selected. Scroll down to the uniquemember field.
- d. Add the DN of newOIDuser as a new line in the field (cn=newOIDuser, cn=Users, dc=us, dc=oracle, dc=com), and click Apply.
- e. You can test the privilege granted by logging in as newOIDuser and creating a new user.

13. Modify the default password policy by changing the attribute value of Password Maximum Failure (pwdmaxfailure) to 2.

- a. Start ODM if not already started, and connect as an Oracle Internet Directory administrator.
- b. Navigate to and expand the Password Policy Management node, and select the Password Policy for Realm dc=us, dc=oracle, dc=com node.
- c. The password policy properties are displayed in the right pane on four tabbed pages. Click the Account Lockout tab.
- d. Click the Password Maximum Failure field and change the value from 10 to 2.
- e. Click the Apply button to save the changes.
- f. You can test the new password policy settings by logging in as newOIDuser and providing the wrong password twice. The next time you try to log in, an error message is displayed stating that the newOIDuser account is locked.
- g. You can unlock the newOIDuser account by resetting the password of newOIDuser (userpassword attribute) as an Oracle Internet Directory administrator.
- h. Set the pwdmaxfailure attribute back to 10.

14. Change the Oracle Internet Directory administrator password.

- a. Start ODM and connect as an Oracle Internet Directory administrator.
- b. Click the orcladmin@<host>:port node. The right pane displays various properties of the server in six tabs.
- c. Click the System Passwords tab. You can see the different system usernames and passwords.
- d. Enter a new password (oracleas10g) in the Super User Password field. The password mentioned must have a numeric value.
- e. Click the Apply button to save the new password.
- f. Test the change by logging in as an Oracle Internet Directory administrator with the new password.
- g. Close the ODM interface.

**Note:** The Oracle Internet Directory administrator password is the superuser password that provides you the access to full directory information.



## Practice 13: Managing OracleAS Portal

This practice provides an overview of managing an OracleAS Portal instance.

1. Extract the scripts from the lab13.zip file. Create synonyms and provide execute privileges on necessary packages to the newly created user. You can use provsyns.sql:

```
$ cd $HOME/labs
$ unzip lab13.zip
```

Verify that the ORACLE\_SID and ORACLE\_HOME environment variables are set. Run the \$HOME/labs/set\_infra\_env.sh script if they have not been already set:

```
$ sqlplus portal/<portal_schema_pwd>
```

Get the password for Portal Schema. The Portal Schema entry is in the following search path:

```
EntryManagement > cn=OracleContext > cn=Products > cn=IAS > cn=IAS
Infrastructure Databases > orclReferenceName=infra.us.oracle.com
>OrclResourceName=PORTAL.
```

**Note: orclpasswordattribute in the right pane.**

```
SQL> @provsyns ora01
```

2. Create database providers in the new database user (ora01) by using the provider.sql script:

```
SQL> connect ora01/oracle
SQL> @provider.sql
```

3. Open the home page of OracleAS Portal (Portal:portal component) in Application Server Control. Familiarize yourself with the page controls and information displayed on the page. Explore the Cache Configuration page, the Parallel Page Engine Services home page, and the Providers home page by clicking the corresponding links.
  - a. Open Oracle Application Server Control in your browser, and navigate to the home page of the Oracle Application Server middle-tier instance.
  - b. In the System Components table, click Portal:portal. The home page of the OracleAS Portal instance is displayed.
  - c. You can see the overall status of the OracleAS Portal instance, data on how the OracleAS Portal instance is using OracleAS Metadata Repository, and status of all other Oracle Application Server components that the OracleAS Portal instance is dependent on. You can also see the severity status for components specifically used by the OracleAS Portal instance.
  - d. In the Administration section, click the Portal Cache Settings link. The link takes you to the Cache Configuration page from where you can configure caching directory, size of cache, and cleanup activity.
  - e. Click the Back button on your browser to return to the OracleAS Portal instance home page.
  - f. Click the Parallel Page Engine Services link in either section. The Parallel Page Engine Services home page is displayed. The page shows you the status of the Parallel Page Engine Services. For detailed information or to administer the OC4J\_Portal instance that the Parallel Page Engine Services run in your Oracle Application Server middle-tier instance, click the OC4J\_Portal link.
  - g. In the Component Status section, click the Providers link. The link takes you to the Providers home page from which you can monitor all providers registered with the OracleAS Portal

- h. Click the “Application Server <your Oracle Application Server middle-tier instance name>” locator link at the top of the page to return to the Oracle Application Server middle-tier instance home page.

- Oracle Application Server 10g R2: Administration I B-44

7. Create a new portal user with the data defined in the following table or specify your own. Assign the DBA and PORTAL\_ADMINISTRATORS roles to the user. After you create the user, log in to the OracleAS Portal instance as the new user.

| Property         | Value                                  |
|------------------|----------------------------------------|
| Username         | mycompany_admin                        |
| E-mail Address   | admin@mycompany.com                    |
| Password         | admin123                               |
| Roles Assignment | Check DBA and<br>PORTAL_ADMINISTRATORS |
| Last Name        | mycompany                              |

- a. Click Create New Users on the Portal Builder page. The Create User Page appears. Enter the details as provided in the preceding table, and click Submit.
  - b. A confirmation page appears to indicate that the user has been created. Click Done on that page to return to the Portal Builder page. On the Portal Builder page, log out as PORTAL user and continue to the next step.
  - c. On the Portal page, click Login to open the Sign In window. Enter the username and password of the new user, and click Login.
  - d. The Oracle Application Server Portal Builder page is displayed. You are now logged in as the new user.
  - e. Verify the user data by clicking the Account Info link on the page banner. The Edit Account Information page is displayed. On this page, you can view the user profile and change the password.
8. Import the Company Portal page group into the OracleAS Portal instance using the following export/import script and the dump file that are located in the \$HOME/labs directory: companyportal.csh and companyportal.dmp. You need the password for the Portal schema.
- a. Navigate to the \$HOME/labs directory for this lesson. The directory should contain the export/import script, the companyportal.csh file, and the companyportal.dmp dump file.
  - b. Set the ORACLE\_HOME, ORACLE\_SID, and PATH variables using the \$HOME/labs/set\_infra\_env.sh script:  
\$ . \$HOME/labs/set\_infra\_env.sh
  - c. Ensure that the user has privileges to execute the companyportal.csh script:  
\$ chmod u+x \$HOME/labs/companyportal.csh
  - d. Launch SQL\*Plus and connect to the Infrastructure database as PORTAL as follows:  
sqlplus portal/<portal schema password>  
Quit SQL\*Plus.
  - e. If the connection to the Infrastructure database is successful, run the export/import script in IMPORT mode as follows: (You should provide the <portal\_schema\_password>.)  
\$ cd \$HOME/labs  
\$ companyportal.csh -mode import -s portal -p  
<portal\_schema\_password>  
-pu mycompany\_admin -pp admin123

- ```
-company NONE -c infra -d companyportal.dmp
```
- f. Analyze the script output by checking the log file in the `labs` directory. The name of the log file should be similar to the following:
`companyportal_F51851DD3943BB28E030B98B7D235F77_imp.log`
 - g. In the browser, open the Portal Builder page. Log in with the credentials `Portal/welcome1`, and click the Portal subtab of the Administer tab if it is not already opened.
 - h. In the Export/Import Transport Set portlet, click the Browse Transport Sets icon next to the Name field in the Import a Transport Set section. The list of transport sets ready for import is displayed in a popup window.
 - i. Select the CompanyPortal transport set by clicking its name in the list. The name of the selected transport set is returned to the portlet and displayed in the Name field.
 - j. Click Import to launch the Import Transport Set Wizard.
 - k. In the Import Transport Set Wizard, verify that you are importing the right transport set, accept all the settings on the page, and click Import Now to start the import.
 - l. On the View Log page, click the View Log Of Actions link. The beginning of the import log is displayed. The import activity may take several minutes to complete. Perform periodical checks on the process by clicking the Refresh Log button. When the import finishes, analyze the import log for possible errors.
 - m. Click Close to return to the View Log page.
 - n. Click Close to return to the Administer tabbed page of the Portal Builder page.
9. Verify the import of the CompanyPortal page group by opening its root page. Add this to your bookmarks.
 - a. On the Portal Builder page, click the Navigator link on the page banner to open the Portal Navigator page.
 - b. On the Portal Navigator page, click the Page Groups tab if it is not selected. The list of the available page groups is displayed. Click the view root page of CompanyPortal.
 - c. Add this to your bookmarks.
 10. Deploy the sample Web provider as a Web application of the `my_OC4J` instance. Use `SampleWebProvider.ear` in the `$HOME/labs` file for the deployment, and name the Web application as My Web Provider.
 - a. On the Application Server Control home page of the `my_OC4J` instance, click the Applications tab.
 - b. On the Applications page, click the Deploy EAR file button.
 - c. On the Deploy Application page, locate and select the `SampleWebProvider.ear` file in the J2EE Application field, enter My Web Provider in the Application Name field, and click Continue.
 - d. In the first step of the wizard, verify that the virtual path of the Web application is `/samplewebprovider` by default, and click Next.
 - e. In the second step of the wizard, click Next.
 - f. In the third step of the wizard, review the settings and click Deploy. (Deploying the application may take a while, so be patient.)
 - g. Click OK when you receive confirmation about the successful deployment of the Web application.


11. Register the database provider installed earlier with the following parameters:

Parameter Name	Parameter Value
Name	MY_PROVIDER
Display Name	My Provider
Timeout	20
Timeout Message	My Provider Timed Out
Implementation Style	Database
Owning Schema	Ora01
Package Name	MY_PROVIDER
Login Frequency	Never

Navigate to the Portal Builder page by clicking the Home icon on your current portal page banner.


- Log in to the Portal page as the PORTAL user (with the welcome1 password). Click the Administer tab and then the Portlets subtab.
- Click the Register a Provider link in the Remote Providers portlet. The first step of the Register Provider Wizard is displayed:
In the Name field, enter MY_PROVIDER.
In the Display Name field, enter My Provider.
In the Timeout field, enter 20.
In the Timeout Message field, enter My Provider Timed Out.
In the Implementation Style field, select Database.
- Click Next to proceed to the second step of the wizard:
In the Owning Schema field, enter the name of the provider schema (ora01).
In the Package Name field, enter MY_PROVIDER.
Verify that the Login Frequency parameter is set to Never.
- Click Finish to register the provider and return to the Portal Builder page.
- Click OK.
- Click Logout to log out of Portal.

12. Navigate to your portal page and add the installed PL/SQL portlet to the page.

- Log in to the Portal page as the portal user (with the welcome1 password), open the Navigator, and click the Page Groups tab if it is not enabled. In the Page Groups list, click the Edit Root Page link next to your CompanyPortal page group. The root page is displayed in Edit mode.
- Click the Add Portlet icon  in the second empty region.
- The Add Portlets page appears. Click Portlet Staging Area in the Portlet Repository pane.
- The Available Portlets appears. Click My Provider.
- The list of the available portlets is displayed. Click My Portlet.
- My Portlet appears in the Selected Portlets pane. Select My Portlet in the Selected Portlets pane.
- Click OK to submit the change and return to the page. You should see the portlet output on the page.

13. Change the password for the mycompany_admin subscriber to welcome1.
 - a. Start ODM and connect as an Oracle Internet Directory administrator.
 - b. Navigate to and expand the Entry Management node until the cn=Users node under dc=com, dc=oracle, dc=us, and access the mycompany_admin user.
 - c. Various properties of the entry are displayed in the right pane. Scroll down until you find the userpassword field.
 - d. Change the value in the userpassword field to welcome1. Then, click the Apply button.
14. Enable Refresh Cache for Oracle Internet Directory Parameters.
 - a. Open a browser window and open the Oracle Portal Web site by entering `http://<host name>.<domain>:<Oracle HTTP Server port of Portal instance>/pls/portal`. Click the Login link on this page.
 - b. The SSO login page is displayed. Enter the username (portal) and password (welcome1), and click Login.
 - c. The OracleAS Portal Builder page is displayed. Click the Administer link to display the administration page. Click the Portal subtab if it is not already clicked.
 - d. Click the Global Settings link under Services.
 - e. The global setting page displays tabs for different component settings. Click the SSO/OID tab to set the properties of the Oracle Internet Directory server.
 - f. Scroll down to the Cache for OID Parameters and select the Refresh Cache for OID Parameters check box. Click Apply. This refreshes the Oracle Internet Directory and DAS parameter information used by OracleAS Portal.

Practice 14: Configuring OracleAS Portal

1. In your OracleAS Portal instance, set up the self-registration feature that does not require approval.
 - a. In your browser, enter the URL `http://<host name>.<domain>:<Oracle HTTP Server port of portal instance>/pls/portal` (for example, `http://edrsr16p1.us.oracle.com:7778/pls/portal`).
 - b. On the OracleAS Portal Welcome page, click the Login link. The Sign-On page is displayed.
 - c. Log in to the OracleAS Portal instance by entering PORTAL as a username and the password that you have specified during the installation of Oracle Application Server middle-tier instance for the `ias_admin` user (password `welcome1`). The OracleAS Portal Builder page is displayed.
 - d. Click the Administer tab, and then click the Portal subtab if it is not selected.
 - e. Click the Global Settings link in the Services portlet. The Main tab of the Global Settings page is displayed.
 - f. Scroll down to the Self-Registration Options section, and select the Enable Self-Registration check box. Select the No Approval Required option.
 - g. Click OK to confirm the change and return to the Portal Builder page.
 - h. On the Portal Builder page, click the Navigator link.
 - i. Click the Edit Root Page link next to CompanyPortal.
 - j. Click the Edit Defaults icon above Logout. 
 - k. On the Login Portlet Settings page, scroll down to the Self-Registration section. Select Enable Self-Registration. Click Apply. Click OK.
2. Test the self-registration feature from the CompanyPortal page. To accomplish this task, you need to log out of the OracleAS Portal instance and open the root page of the CompanyPortal by using the bookmark that you created in the last practice of the lesson titled "Managing the OracleAS Portal Instance." You also need to register a new portal user by using the self-registering feature. Specify information about the new user as defined in the table below or specify your own.

Property	Value
Username	mycompany_user
Password	user123
E-mail Address	user@mycompany.com

The Login portlet should now display the Create New Account link.

- a. Log out of the portal by clicking the Logout link on the page banner. The Single Sign-Off page appears.
- b. Navigate to the root page of the CompanyPortal page group by either clicking Back in the browser or selecting the bookmark created in the last practice of the lesson titled "Managing the OracleAS Portal Instance." The root page of the Company Portal page group may appear as if you were still logged in. This is because the page content has been cached. Reload the page in the browser. You should be able to see the Company Portal home page as a public user. Note that the Login portlet now contains the Create New Account link.

- c. Register a new portal user using the self-registration feature. Specify information about the new user as defined in the following table or specify your own.
 - d. In the Login portlet, click the Create New Account link. The Self-Registration page is displayed.
 - e. In the Account Details region, enter information about the new user in the corresponding fields. Optionally, you can enter the personal information of the user in the Personal Details region. Then, click Create.
 - f. After successful registration, you should see the following confirmation message at the top of the page:
Thank you for registering. You may now log in to the portal with the username and password you specified.
 - g. Click Cancel to return to the root page of Company Portal.
 - h. Log in to the portal as the new portal user.
 - i. In the Login portlet, enter the username and password of the new user, and click Login. The root page of the Company Portal page group is displayed.
3. Install an additional language to the OracleAS Portal instance (for example, French) by using `ptllang`. Make sure that the `ORACLE_HOME` variable is set to the Oracle home directory of the middle tier before running the script. Verify the language installation by checking the `portal_f.log` file that contains the Portal assistant output.
 - a. In a terminal window, set the `ORACLE_HOME` variable to `/home/oracle/portal`. Then, navigate to the directory from where you can execute `ptllang`:


```
$ export ORACLE_HOME=$HOME/portal
$ cd /modules/stage/Portal10-1-4/assistants/opca
```
 - b. Run the `ptllang` script in `LANGUAGE` mode to install the NLS strings that store translations in a new language.

Note:

 - The `<db_connect_string>` parameter is `<port of the infrastructure>:<infra>`.
 - In case the parameter above for `<db_connect_string>` does not work, you need to replace the value by `<port of the infrastructure>:<infra.oracle.us.com>` or `<host.us.oracle.com>:<infrastructure port>:<infra>`.

For example, enter the following command to install French:

```
$ ptllang.sh -i custom -s portal -sp <portal_schema_password>
-c <db_connect_string> -lang f
```
 - c. When the script finishes, view the `portal_f.log` file for possible errors. If the installation is successful, then at the end of the log file you should see the list of the installed languages that includes the new language. For example, you should see the following line for French:

French	f	INSTALLED AVAILABLE
--------	---	---------------------
 4. Test the installed language in CompanyPortal by refreshing its root page.
 - a. The new language should be displayed in the Set Language portlet.

- b. Click the link that represents the new language. The Company Portal home page is displayed in the selected language.
- c. Change the language back to English.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Practice 15: Administering the OracleAS Single Sign-On Server

1. List the components of the OracleAS Single Sign-On server.

The various components of the OracleAS Single Sign-On server are:

- a. SSO server
- b. Partner application
- c. External application
- d. mod_osso

2. Increase the duration of the SSO server session to 12 hours.

- a. Open a browser window, and open the Oracle SSO server home page by entering `http://<host name>.<domain>:<Oracle HTTP Server port of infra instance>/pls/orasso`. Then, click Login.
- b. The SSO login page appears. Enter the username (orcladmin) and password (welcome1), and click Login. Now you have access to modify the SSO server properties.
- c. Click SSO Server Administration. This displays the page from which you can change the SSO server properties.
- d. Click Edit SSO Server Configuration.
- e. On the Edit SSO Server page, enter 12 for the Single Sign-On session duration field. Then, click the OK button.
- f. This will increase a user session with the SSO server to 12 hours from 8 hours (default).

3. Restart the SSO server by using Application Server Control.

- a. Open a browser window, and open the Application Server Control Console of the Infrastructure instance.
- b. In the Standalone Instances section, click the link for the Infrastructure instance. The page that displays the various components of the server appears. The System Components table lists various components with their statuses. Select the check box beside Oracle HTTP Server and OC4J_SECURITY, and click Restart. Click Yes to confirm the request. This restarts the Oracle HTTP server and the OC4J Security components, and the SSO server is started.
- c. Set the ORACLE_HOME, ORACLE_SID, and PATH variables using the `$HOME/labs/set_infra_env.sh` script:

```
$ . $HOME/labs/set_infra_env.sh
```
- d. In a terminal window, check whether the database listener is operational:

```
$ORACLE_HOME/bin/lsnrctl status | grep status
```
- e. Check whether the database repository is operational:

```
sql> connect sys/password_for_sys as sysdba
sql> startup
sql> exit
```

4. Add an external application to the SSO server and save its credentials in the server.

- a. Open a browser window, and open the Oracle SSO server home page by entering `http://<host name>.<domain>:<Oracle HTTP Server port of infra instance>/pls/orasso`. Then, click Login.

- b. Enter the username (orcladmin) and password (welcome1), and click Login. Now you have access to modify the SSO server properties.
- c. Click the SSO Server Administration link. This displays the page on which you can add an external application to the SSO server.
- d. Click the Administer External Applications link.
- e. This displays the Administer External Applications page. Click the Add External Application link.
- f. The Create External Application page appears. Scroll down to the External Application Login section and enter the details. Here, an example of otn login is shown. However, this may not be available in your classrooms.
 - Application Name: otn
 - Login URL: http://otn.oracle.com
 - User Name/ID Field Name: login
 - Password Field Name: passwd
 - In the Authentication Method section, select POST from the list.
 - In the Additional Fields section, enter the following details:
 - Field Name field: .persistentY
 - Field Value field: [off]
- g. Click the OK button to create the external application.
- h. The Administer External application displays the new external application, otn.

Note: Because of the firewalls and the security policies of Oracle, you may not be able to access external applications sites.

External Application Login

Enter the application name, the login URL, and the user name and password HTML field names used by the application's login form. The login URL is typically the submit action of the application's login form. It will be used in conjunction with the user name and password field names to perform a single sign-on login into this application. The login URL as well as the user name and password field names should be determined by inspecting the source of the application's standard login form. User name/id, password, additional field etc. values are not required for Basic authentication and Login URL should be a url which requires authentication.

Application Name:	<input type="text" value="otn"/>
Login URL:	<input type="text" value="http://otn.oracle.com"/>
User Name/ID Field Name:	<input type="text" value="login"/>
Password Field Name:	<input type="text" value="passwd"/>

Authentication Method

Select the authentication method used by this application. The POST method submits the credentials with the body of the form. The GET method submits the login credentials as part of the login URL.

Type of Authentication Used:

Additional Fields

Type the names and values of any additional fields that are submitted with the login form of the external application.

Field Name	Field Value	Display to User
<input type="text" value=".persistentY"/>	<input type="text" value="[off]"/>	<input type="checkbox"/>

5. View the OracleAS SSO server monitoring pages.
 - a. Open a browser window, and open Application Server Control. Navigate to the Infrastructure instance home page.
 - b. Click the Single Sign-On:orasso link in the System Components table to display the SSO server monitoring pages.
 - c. The SSO Server Monitoring page displays the login details for the last 24 hours. It displays the success and failure percentages, and the logins that failed in the last 24 hours with the count.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Practice 16: Managing Access Using Oracle Delegated Administration Services

1. Add “modify user privilege” to the newOIDuser user created in the earlier practice.
 - a. Open a browser window and open the Oracle Internet Directory Self Service Console home page by entering `http://<host name>.<domain>:<Oracle HTTP Server port of infrastructure instance>/oiddas`. Click Login.
 - b. The SSO login page appears. Enter the username (orcladmin) and password (welcome1), and click Login.
 - c. Click the Directory tab. Then, enter the username (new%) in the search field, and click Go. This displays newOIDuser. Click the option button beside newOIDuser, and click the Privileges button to assign the privilege to the user.
 - d. On the Assign Privileges to User page, select Allow user editing and Allow Delegated Administration Service configuration, and click Submit.
 - e. You can test the privilege assigned by logging in as newOIDuser (if it is not locked) and by modifying your properties.
 - f. Log out from Oracle Internet Directory Self-Service Console.
2. Unlock the account of a user account. In this case, newOIDuser is locked. In case it is not, you can lock the user account by entering a wrong password twice. The steps to unlock newOIDuser are:
 - a. Open a browser window and open the Oracle Internet Directory Self Service Console home page by using the URL `http://<host name>.<domain>:<Oracle HTTP Server port of Infrastructure instance>/oiddas`. Click Login.
 - b. The SSO login page appears. Enter the username (orcladmin) and password (welcome1) for the Oracle Internet Directory administrator, and click Login.
 - c. Click the Directory tab. Then, search the user accounts that are locked by entering the username in the search field for the user and clicking Go. This displays all the user accounts that are locked.
 - d. Select newOIDuser, and click Unlock. Click Yes.
3. Create a user by using OracleAS Portal. OracleAS Portal provides links to Oracle Internet Directory Self-Service Console.
 - a. Open a browser window, and open the OracleAS Portal home page by entering `http://<host name>.<domain>:<Oracle HTTP Server port of portal instance>/pls/portal`. Click the Login icon.
 - b. The SSO login page appears. Enter the username (newOIDuser) and password (newOIDuser1), and click Login.
 - c. Click the Administer tab. Then, click Create New Users in the User section. This opens Oracle Internet Directory Self-Service Console.

- d. In the Create User window, you can enter various details about the new user. You can enter the information as shown below:

Basic Information

First Name: newuser2

Last Name: newuser2

User ID: newuser2

Password: newuser2

Confirm Password: newuser2

E-mail Address: newuser2@xyz.com

Additional Personal Details

Single Sign On Enabled: Enabled

Click Submit when done.

- e. Finally, click Done.

You can verify the creation of the new user (newuser2) by searching for the user in the Edit/Delete User section.

Practice 17: Managing and Configuring OracleAS Certificate Authority

Note

- The OCA operational steps are dependent on the Web browser.
- Because you will be using a single browser, it is advisable to clear the certificates each time you change the user (from administrator to user and back to administrator).
- To enable the browser to prompt you before accepting certificates, you can perform the following steps:
 - a. Open Mozilla or Netscape browser, and select Edit > Preferences.
 - b. In the Category pane, expand the Privacy and Security node and select Certificates. In the right pane, the certificate-related information is displayed.
 - c. In the Client Certificate Selection section, select the Ask Every Time option button and click OK. This enables you to select the client certificate as required for a particular operation. Otherwise, the browser provides the certificate automatically, which may not be correct as per the operation and can cause unexpected errors.
- Ensure that there is no OCA Web Administration certificate. Remember that welcome1 is the password for OCA Administrator.
 - a. Stop the OCA server by using the `$HOME/infra/oca/bin/ocactl stop` command.
 - b. Run the command:
`$HOME/infra/oca/bin/ocactl revokecert -type WEBADMIN`
 - c. Start the OCA server by using the `$HOME/infra/oca/bin/ocactl start` command.
- Before accessing the OCA Administration or User pages, find the port number for the OCA server (usually 6600) using Oracle Enterprise Manager 10g Application Server Control.
- The browser may prompt you when you shift from an encrypted page to an unencrypted page or vice versa. Select the appropriate response and continue.

Practices

1. View the status of the OCA server and start it if it is not started already.
 - a. Open a terminal window and ensure that the `ORACLE_HOME`, `ORACLE_SID`, and `PATH` environment variables are appropriately set. To ensure this, you can run the `set_infra_env.sh` script in your `$HOME/labs` directory:
`$. $HOME/labs/set_infra_env.sh`
 - b. Change your directory to the `oca/bin` directory of your Infrastructure Oracle Home and run the command as shown:
`ocactl status`
Enter `welcome1` if requested for the OracleAS Certificate Authority administrator password.
 - c. Run the `ocactl start` command if OCA service is not running. Enter `welcome1` if requested for the OracleAS Certificate Authority administrator password.

2. Access the OCA administration page and enroll for a certificate.

- a. Open a browser, and open the OCA administration page by using the URL `https://<host name>.<domain>:<Oracle HTTP Server OCA Server Authentication (SSL) port>/oca/admin` in the address bar. You can obtain the Oracle HTTP Server Authentication (SSL) port from the ports page of Application Server Control Console for Infrastructure instance. As the communication between your browser and the server is secure, the OCA server downloads a certificate. Click OK to accept the certificate.
- b. You may be prompted with the Security warning. Click OK to access the Certificate Authority Page. Click the “Click here” link to enroll for a certificate.
- c. On the Web Administrator Enrollment page, enter the details as follows. Enter the OCA administrator password as `welcome1`. Then, click Submit.
Common Name: `ocawebadmin`
Organization: `ABC`
Password: `welcome1`
- d. In case you are prompted for the Change Master password for a security device, enter `welcome1` as the password.
- e. This displays the Approved Certificate Information page. Click Install in Browser to import the certificate into the browser.
Note: There is no confirmation message after you import the certificate into the browser. This certificate will be used every time you connect to the OCA administration page.
- f. After you import the certificate, once again access the OCA Administration URL (`https://<host name>.<domain>:<Oracle HTTP Server OCA Server Authentication (SSL) port>/oca/admin`). This displays the OCA Administration Pages page.
- g. You can click the Certificate Management tab to manage certificates. This prompts you for the certificate to be used for authentication. Select the certificate that you just imported to the browser, and click OK. The Certificate Management page is displayed.

Practice 18: Securing OracleAS Components by Using SSL

Note

- The OCA operational steps are dependent on the Web browser.
- Because you will be using a single browser, it is advisable to clear the certificates each time you change the user (from administrator to user and back to administrator).
- To enable the browser to prompt you before accepting certificates, you can perform the following steps:
 - a. Open Mozilla or Netscape browser, and select Edit > Preferences.
 - b. In the Category pane, expand the Privacy and Security node and select Certificates. In the right pane, the certificate-related information is displayed.
 - c. In the Client Certificate Selection section, click the Ask Every Time option button. This enables you to select the client certificate as required for a particular operation. Otherwise, the browser provides the certificate automatically, which may not be correct as per the operation and can cause unexpected errors.
- Ensure that there is no OCA Web Administration certificate. Remember that welcome1 is the password for OCA Administrator.
 - a. Stop the OCA server by using the `$HOME/infra/oca/bin/ocactl stop` command.
 - b. Run the command:
`$HOME/infra/oca/bin/ocactl revokecert -type WEBADMIN`
 - c. Start the OCA server by using the `$HOME/infra/oca/bin/ocactl start` command.
- Before accessing the OCA Administration or User pages, find the port number for the OCA server (usually 6600) by using Oracle Enterprise Manager 10g Application Server Control.
- The browser may prompt you when you shift from an encrypted page to an unencrypted page or vice versa. Select the appropriate response and continue.

Practices

1. Create a new wallet.
 - a. Open a terminal window and set the Oracle environment by using the `$HOME/labs/set_infra_env.sh` script. Change your directory to `$ORACLE_HOME/bin` and execute `owm` at the command prompt. This displays Oracle Wallet Manager:

```
$ . $HOME/labs/set_infra_env.sh
$ cd $ORACLE_HOME/bin
$ ./owm
```
 - b. Click Wallet in the menu, and select New.
 - c. A dialog box appears prompting you to create the default wallet directory.
 - d. Click Yes. (Click Yes again if you get a message “Unable to create system default wallet directory.”) This displays the New Wallet dialog box prompting you to enter the wallet password and confirming it. Enter `welcome1` as the wallet password.

- e. Click OK to create the new wallet. This shows a dialog box prompting you to create an empty wallet or to request a certificate. Click No. The new wallet appears in the navigation tree with trusted certificates.
2. Create a certificate request for the wallet created.
 - a. In Oracle Wallet Manager, select the empty certificate. Then, click the Operations menu and select Add Certificate Request.
 - b. In the Create Certificate Request dialog box, enter the details as shown, and click the OK button:

Common Name: ken
Organizational Unit: finance
Organization: abc
Locality/City: sfo
State/Province: ca
Country: United States
Key Size: 1024 bits
 - c. In the Oracle Wallet Manager confirmation dialog box, click OK.
 - d. A certificate request is created. You can submit this request to Certificate Authority. You can use the OCA User Pages to submit this request for a certificate.
 3. Use OCA User Pages to request for a server certificate. For this practice, select “Use your OracleAS Single sign-on name and password”.
 - a. Close all browser windows. You may want to clear the certificates from your browser, because you are to log in as a user to the OCA, and not as an administrator.
 - b. Open a new browser window and open the OracleAS Certificate Authority with the URL `https://<host name>.<domain>:<Oracle HTTP Server OCA Server Authentication (SSL) port>/oca/user`. You can confirm the Oracle HTTP Server OCA Server Authentication (SSL) port from Oracle Enterprise Manager 10g Application Server Control. Then, click the Server / SubCA Certificates tab.
 - c. Click Request a Certificate.
 - d. Navigate to the Oracle Wallet Manager window and copy the certificate request as follows:

Certificate Request:

```

-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBkDCB+gIBADBRMQswCQYDVQQGEwJVUzELMAkGA1UECBMCY2ExDDAK
A1UEChMDYWJjMQswCQYDVQQLEwJocjEEMMAoGA1UEAxMDam91MIGfMA0C
ADCBiQKBgQCvBCMve9/yN5ZRUAZ06SA0t6XfpgRorwJC1YPRD9ORrp1f
uh4vw1eDxBasRtys59AoX7fFG3FbskRmNLapBMopeb9GUwnDjN+NBKb+
b6h2rKBg4stK367GvvXAY21QWj8qQIDAQABoAAwDQYJKoZIhvcNAQEE
MEgWhg+iTSGmQyJEbs8MO2WARTfDdCQ7GHES43XANZ0kzQgPCuViuv2
xdt525TjRH2JcvbMoRxs5ShI9A+ZN+/OrCTK8TAav7qONRgsHoM7g9g
g4rKvCQ=
-----END NEW CERTIFICATE REQUEST-----

```

- e. Navigate to the Server / SubCA Certificates request page. Paste the certificate into the text area.
 - f. Enter the following details:
 - Name: Ken
 - E-Mail ID: Ken@abc.com
 - Phone Number: 3455432
 Click Submit.
 - g. The OCA server generates the server certificate and stores it in the Oracle Internet Directory server. The Approved Certificate Information page displays the certificate information. Note the certificate ID. Click OK. You can use this certificate for any future SSL communication.
 - h. To close Oracle Wallet Manager, navigate to the Oracle Wallet Manager window and click Wallet. Then, click Exit
 - i. When prompted for saving the current wallet, click No.
4. Approve the server certificate request ID by using the OCA Administration Pages.

Note: Because you are using the same machine and browser profile for accessing and administering certificates, you may need to take the following actions:

- a. Stop the OCA (`ocactl stop`).
 - b. Revoke Web Administrator Certificate:
`$ ocactl revokecert -type WEBADMIN`
 - c. Start OCA (`ocactl start`).
 - d. Also, remove the certificate from your browser.
- a. Close all browser windows. Enroll the Administrator again as in Practice 2 of the lesson titled “Managing and Configuring OracleAS Certificate Authority.” (Access the OCA administration page and enroll for a certificate.)
 - b. Access the OCA administration page in your browser window using the URL `https://<host name>.<domain>:<Oracle HTTP Server OCA Server Authentication (SSL) port>/oca/admin`. You can confirm the Oracle HTTP Server OCA Server Authentication (SSL) port from Oracle Enterprise Manager 10g Application Server Control.
 - c. Click the Certificate Management tab to open the User Identification Request page. You can manage various certificates by using this request page. The certificates pending your approval are displayed by default. Select the certificate request ID generated in the earlier practice. Then, click View Details.
 - d. The Certificate Request Details page appears displaying the details of the user certificate request. Verify the details provided by the user, and then click Approve.
 - e. A notification of the certificate request approval appears. Click OK.
 - f. When you search for the certificate by using OCA user pages, the newly approved certificate is displayed. Note that the status of the certificate is displayed as Valid.
5. Change the HTTP server configuration to enable SSL.
 - a. Open the browser and enter the following URL:
`http://<host name>.<domain>:<Application Server Control of portal instance>`
 Log in as the `ias_admin` user. This starts Oracle Application Server Control.

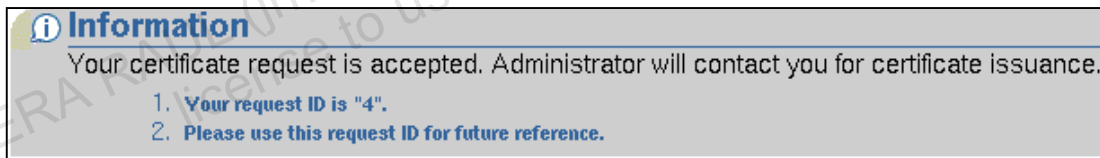
- b. Navigate to the Portal instance home page.
- c. Click Process Management at the bottom of the page.
- d. In the `opmn.xml` file, change the value for the "HTTP_Server" "start-mode" to "ssl-enabled".
- e. Click the Apply button. Click OK on the Confirmation page.

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

6. Use OCA User Pages to request for a client certificate.
 - a. Close all browser windows. You may want to clear the certificates from your browser, because you are to log in as a user to the OCA, and not as an administrator.
 - b. Open a new browser window and open the OracleAS Certificate Authority with the URL: `https://<host name>.<domain>:<Oracle HTTP Server OCA Server Authentication (SSL) port>/oca/user`. You can confirm the Oracle HTTP Server OCA Server Authentication (SSL) port from Oracle Enterprise Manager 10g Application Server Control. Then, click the User Certificates tab.
 - c. The Authentication page appears. You can select from any of the options displayed. For this practice, select "Use your OracleAS Single sign-on name and password." Then, click the Submit button.
 - d. The Single Sign-On page appears. Enter your SSO username (`newOIDuser`) and password (`newOIDuser1`). Then, click Login.
 - e. The User Certificates – SSO Authentication page appears. To get a certificate, click Get Certificate. This certificate is used for any future SSL communication.
 - f. The Certificate Request Form – SSO Authentication page is displayed. Here, the following values are listed:

User DN: `cn=newOIDuser,cn=users,dc=us,dc=oracle,dc=com`
 Certificate Key Size: 2048 (High Grade)
Note: To change the key size, select 1024 (Medium Grade) in the Certificate Key Size drop-down menu.
 Certificate Usage: Authentication, Signing, Encryption
 Then, click Submit.
 - g. The OCA server generates the user certificate and stores it in the Oracle Internet Directory server. The Approved Certificate Information page displays the certificate information. Click Install in Browser to store the certificate in your browser. If the browser prompts for entering Master password for storing certificate in a security device, enter `welcome1`.
 - h. You can use this certificate for any future SSL communication.

7. Use Oracle Application Server OCA User Pages to request for a client certificate by using the manual approval authentication.
 - a. Open a browser window and go to the OracleAS Certificate Authority user pages by entering the URL `https://<host name>.<domain>:<Oracle HTTP Server OCA Server Authentication (SSL) port>/oca/user` in the address bar. Then, click the User Certificates tab. You can confirm the Oracle HTTP Server OCA Server Authentication (SSL) port from Oracle Enterprise Manager 10g Application Server Control.
 - b. The Authentication page appears. You can select from any of the options. For this practice, select “Use manual approval / authentication.” Then, click Submit.
 - c. The User Certificates – Manual Authentication page appears. Click the Request a Certificate button to get a certificate.
 - d. On the Certificate Request Form – Manual Authentication page, enter the details as shown:
 - Common Name: Joe
 - Organization: xyz
 - Name: Joe
 - E-Mail ID: joe@xyz.com
 - Phone Number: 203 456 7890
 - Certificate Usage: Encryption
 - Validity Period: 6 months
 - e. Click Submit to submit the request for the information.
 - f. The OCA server assigns a request ID for your request for any future reference and sends a message. Note the request ID in this message, and click OK.



- g. The OCA administrator needs to approve your request using the OCA Administration interface. After the approval, you can use the certificate to authenticate.
8. Approve the client certificate request ID by using OCA Administration Pages.

Note: Because you are using the same machine and browser profile for accessing and administering certificates, you may need to take the following actions:

- a. Stop the OCA (`ocactl stop`).
 - b. Revoke Web Administrator Certificate:
`$ ocactl revokecert -type WEBADMIN`
 - c. Start OCA (`ocactl start`).
 - d. Also, remove the certificate from your browser.
 - a. Close all browser windows. Enroll the Administrator again as in Practice 2 of the lesson titled “Managing and Configuring OracleAS Certificate Authority.” (Access the OCA administration page and enroll for a certificate).

- b. Access the OCA administration page in your browser window using the URL: `https://<host name>.<domain>:<Oracle HTTP Server OCA Server Authentication (SSL) port>/oca/admin`. You can confirm the Oracle HTTP Server OCA Server Authentication (SSL) port from Oracle Enterprise Manager 10g Application Server Control.
 - c. Click the Certificate Management tab to manage various certificates. The certificates pending your approval are displayed by default. Select the certificate request ID generated in the earlier practice. Then, click View Details.
 - d. The Certificate Request Details page appears displaying the details of the user certificate request. Verify the details provided by the user, and then click Approve.
 - e. A notification of the certificate request approval appears. Click OK.
 - f. When you search for the certificate by using OCA user pages, the newly approved certificate is displayed. Note that the status of the certificate is displayed as Valid.
 - g. The user can import this certificate from the OCA user interface.
9. Change the HTTP server configuration to “require” client certificate.
- a. Access the portal instance home page.
 - b. Click HTTP_Server in the System Components table.
 - c. Navigate to the Administration property page.
 - d. Click Advanced Server Properties.
 - e. Click `httpd.conf`. Add these directives at the end of the file:
 `SSLVerifyClient require`
 - f. Click Apply.
 - g. Click Yes to restart Oracle HTTP Server.
 - h. Click OK.

Practice 19: Backing Up and Restoring Oracle Application Server

This practice provides experience in backing up and restoring configuration settings. The practices for this lesson are as follows:

- Lab 19.1: Set up the environment
- Lab 19.2: Install the backup and recovery tool
- Lab 19.3: Perform a backup of the middle-tier installation
- Lab 19.4: Restore and verify the backup

For these practices, set `ORACLE_HOME` to `/home/oracle/portal`.

Lab 19.1: Set Up the Environment

Set the `ORACLE_SID`, `ORACLE_HOME`, and `PATH` environment variables appropriately. You can modify the `ORACLE_HOME` environment variable in the `set_infra_env.sh` script in your `$HOME/labs` directory and use it.

Lab 19.2: Install the Backup and Recovery Tool

1. Ensure that the `bkp_restore.pl` script has execute permission:

```
prompt> cd $HOME/portal/backup_restore
prompt> chmod +x bkp_restore.sh
```
2. Create directories to hold the backup and log files:

```
prompt> mkdir -p $HOME/labs/lesson19/backups/portal/log_files
prompt> mkdir -p $HOME/labs/lesson19/backups/portal/config_files
```
3. Edit the `config.inp` file in the `$ORACLE_HOME/backup_restore/config/` directory and enter values for the following variables:

```
oracle_home=/home/oracle/portal
log_path=/home/oracle/labs/lesson19/backups/portal/log_files
config_backup_path=/home/oracle/labs/lesson19/backups/portal/config_files
$ cd $ORACLE_HOME/backup_restore/config
$ gedit config.inp
```

Make the suggested changes and save the file.
4. Execute the script to configure the backup parameters:

```
$ cd $ORACLE_HOME/backup_restore
$ ./bkp_restore.sh -m configure
```

Note: This updates parameters in `config.inp` to indicate that this is a middle-tier backup.

Lab 19.3: Perform a Backup of the Middle-Tier Installation

1. Shut down your middle-tier instance.
 - a. Change to the `ORACLE_HOME` directory:

```
$ cd $ORACLE_HOME
```
 - b. Stop the console:

```
$ ./bin/emctl stop iasconsole
```

- c. Stop the instance:


```
$ ./opmn/bin/opmnctl stopall
```
2. Perform a complete backup of the middle tier.
 - a. Create a DCM archive of the middle-tier instance:


```
$ ./dcm/bin/dcmctl createArchive -archive portal_archive
```
 - b. Back up the Oracle Home of the middle tier:


```
$ tar cvf midtierbackup.tar *
```
 - c. Back up the middle-tier configuration files:


```
$ cd backup_restore
$ ./bkp_restore.sh -m backup_config
```

Note: If you do not have SSO configured, you will receive an error message stating that the SSO configuration file (`osso.conf`) does not exist and could not be copied. Ignore this error.

- d. Verify that the configuration files have been backed up:


```
$ cd $HOME/labs/lesson19/backups/portal/config_files/
$ ls -l *
```
3. Modify a configuration file, for example `httpd.conf`:


```
$ cd $ORACLE_HOME/Apache/Apache/conf/
$ cp httpd.conf httpd.conf.bak
$ vi httpd.conf
/KeepAliveTimeout 15 (to search for KeepAliveTimeout)
w (to advance to 15)
cw (to change 15)
20 (to set the KeepAliveTimeout to 20)
Esc (to escape out of the change)
ZZ (to save and quit vi)
```
4. To update the changes to the DCM repository, run the `updateConfig` command:


```
$ cd $ORACLE_HOME/dcm/bin
$ ./dcmctl updateConfig -ct ohs
```
5. To reflect these changes, start the `HTTP_Server` component:


```
$ cd $ORACLE_HOME/opmn/bin
$ ./opmnctl startproc ias-component=HTTP_Server
```
6. Shut down the middle tier again by stopping all the OPMN processes:


```
$ ./opmnctl stopall
```
7. Take an online incremental backup of the middle-tier instance configuration:


```
$ cd $ORACLE_HOME/backup_restore/
$ ./bkp_restore.sh -m backup_config_incr
```


Lab 19.4: Restore and Verify the Backup

1. Restore the middle tier by using the original backup:

```
$ cd $ORACLE_HOME/backup_restore/  
prompt> ./bkp_restore.sh -m restore_config -t <timestamp>
```

Enter y when asked whether you want to continue.

Note:

- Note the timestamp associated with the file name beginning with config_bkp located in \$HOME/labs/lesson19/backups/portal/config_files/.
- Enter just the name of <timestamp> in the command:
prompt> ./bkp_restore.sh -m restore_config -t <timestamp>

2. Check the configuration file setting:

```
$ cd $ORACLE_HOME/Apache/Apache/conf/  
$ vi httpd.conf  
    /KeepAliveTimeout (to see that KeepAliveTimeout is set to 15)  
:q (to quit from vi)
```

3. Restore the middle tier by using the incremental backup:

```
$ cd $ORACLE_HOME/backup_restore/  
$ ./bkp_restore.sh -m restore_config -t <timestamp>
```

Enter y when you are prompted whether you want to continue.

Note:

- Enter just the name of <timestamp> located in:
\$HOME/labs/lesson19/backups/portal/config_files/
- Note the timestamp associated with the file name beginning with config_bkp located in \$HOME/labs/lesson19/backups/portal/config_files/.

4. Check the configuration file setting:

After the DCM archive is restored completely, open a new terminal window and check the configuration file setting as follows:

```
prompt> cd $ORACLE_HOME/Apache/Apache/conf/  
prompt> vi httpd.conf  
    /KeepAliveTimeout (to see that KeepAliveTimeout is now set to 20)  
:q (to quit from vi)
```

5. Restart your middle-tier instance:

```
prompt> cd $ORACLE_HOME  
prompt> ./opmn/bin/opmnctl startall  
prompt> ./bin/emctl start iasconsole
```

6. To save space for subsequent practices, remove the backup files:

```
prompt> rm midtierbackup.tar  
prompt> ./dcm/bin/dcmctl removeArchive -archive portal_archive  
prompt> rm -rf  
$HOME/labs/lesson19/backups/portal/config_files/config*
```

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

C

Configuring mod_rewrite

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- **Use regular expressions for pattern matching**
- **Enable `mod_rewrite`**
- **Configure `mod_rewrite` for business operations**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Regular Expressions

Regular expressions are used to operate on strings and can be used for:

- **Pattern matching**
- **Modifying a string**
- **Extracting a substring**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Regular Expressions

There are several directives, such as `DirectoryMatch` and `LocationMatch`, that allow the use of regular expression for pattern matching. `mod_rewrite` enables regular expressions in specifying the `RewriteRule` directive.

Regular expressions are essentially a tiny, highly specialized programming language embedded in Oracle HTTP Server. The main purpose of a regular expression engine is to take a search pattern and see whether a string that matches the pattern occurs in its input. If it does, it is a successful match; if it does not, then the match fails.

Using this language, you specify the rules for the set of possible strings that you want to match; this set might contain English sentences, or e-mail addresses, or anything you like. You can then ask questions such as “Does this string match the pattern?” or “Is there a match for the pattern anywhere in this string?”

A second purpose is to use regular expressions to modify a string or to split it in various ways.

Matching Characters

Metacharacters to be used with regular expressions are the following:

- **.** (dot) matches any character.
- **[]** specifies a class (set of characters).

Examples:

- **[a - z]** matches any lowercase letter.
- **[a - zA - Z0 - 9]** matches any character or any digit.
- **[abc\$]** matches "a ", " b ", " c ", or " \$".
- **[^0 - 9]** matches anything except a digit.

Metacharacters are not active inside classes.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Matching Characters

Because regular expressions are used to operate on strings, you can start with the most common task: matching characters. Most letters and characters simply match themselves. For example, the regular expression `test` will match the string "test" exactly. Thus, to search for `test` in a file, enter `test` as the search pattern. The power of regular expressions comes from metacharacters, which are characters that specify an action.

For instance:

- The **.** (dot) character matches any character, so `test.` will match `test1`, `testa`, and `testb`.
- The brackets **[]** will match any one of a list of characters. If the brackets contain a simple list of characters, it will match any of those. If it contains two characters separated by a **-**, it will match any in the range between the two (for example, `[1 - 3]` will match 1, 2, and 3). The order of characters is insignificant.
A **^** (caret) after the **[** (bracket) means match any character but those specified by the brackets.

Note: Metacharacters are not active inside classes. For example, `[abc$]` will match any of the characters "a", "b", "c", or "\$"; "\$" is usually a metacharacter, but inside a character class it is stripped of its special nature.

Rules for Regular Expressions

The following rules apply to regular expressions:

- **Regular expressions are case sensitive; “hello” does not match “Hello.”**
- **Each character inside the search pattern is significant including whitespace characters (space, tab, new line).**
- **Alternating text can be enclosed in parentheses and alternatives separated with a pipe (|) character.**

For example, (on|ue|rda) matches “Monday,” “Tuesday,” or “Friday.”

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Rules for Regular Expressions

The simplest pattern matched by a regular expression is a literal character or a sequence of literal characters. Anything in the target text that consists of exactly those characters in exactly the order listed will match. A lowercase character is not identical with its uppercase version, and vice versa. A space in a regular expression matches a literal space in the target.

Using character class, you can indicate that at least one member (character or expression) of the class occurs in the specified spot. If you want to specify that either of the two whole subexpressions occur in a position in the regular expression, then use the alternation operator, that is, the pipe character (|). This is the symbol that is also used to indicate a pipe in UNIX/DOS shells, and is sometimes called the pipe character.

The pipe character in a regular expression indicates an alternation between everything in the group that encloses it. That is, even if there are several groups on either side of a pipe character, the alternation matches everything on both sides. To select the scope of the alternation, you must define a group with () that encompasses the patterns that may match. The example illustrates this.

Metacharacters ^ and \$

There are two special characters that can be used to search for lines starting or stopping with the matching string:

- **^ matches the start of a line.**
- **\$ matches the end of a line.**

Examples:

- **^apache matches any line that starts with apache.**
- **apache\$ matches any line that ends with apache.**
- **^apache\$ matches any line that consists of just the word apache.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Metacharacters ^ and \$

Two special characters are used in almost all regular expression tools to mark the beginning and end of a line: caret (^) and dollar sign (\$). To match a caret or dollar sign as a literal character, you must escape it, which will be explained later in this lesson.

An interesting aspect about the caret and dollar signs is that they match zero-width patterns; that is, the length of the string matched by a caret or dollar sign by itself is zero (but the rest of the regular expression can still depend on the zero-width match). Many regular expression tools provide another zero-width pattern for word-boundary (\b). Words might be divided by white space (such as spaces, tabs, and new lines) or other characters (such as nulls); the word-boundary pattern matches the actual point where a word starts or ends, not the particular white space characters.

Quantifiers for Characters

Regular expressions also allow multipliers that modify the behavior of the previous matching character:

- **? matches zero or one instance of the character.**
- **+ matches one or more instances of the character.**
- *** matches zero or more instances of the character.**

Examples:

- **test? matches tes and test.**
- **test+ matches test, testt, testtt, and so on.**
- **test* matches tes, test, testt, testtt, and so on.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Quantifiers for Characters

Matching varying sets of characters is the first task of a regular expression. The metacharacter for repeating characters “*” does not match the literal character “*”; instead, it specifies that the previous character can be matched zero or more times, instead of exactly once.

For example, `ca*t` will match “`ct`” (0 “a” characters), “`cat`” (1 “a”), and “`caaat`” (3 “a” characters). When repeating a regular expression, the matching engine will try to repeat it as many times as possible. Probably the easiest mistake to make in composing regular expressions is to match too much. When you use a quantifier, you want it to match everything (of the right sort) up to the point where you want to finish your match. However, when using “*”, “+”, or numeric quantifiers, it is easy to forget that the last bit you are looking for might occur later in a line than the one you are interested in.

Another repeating metacharacter is +, which matches one or more times. Pay careful attention to the difference between * and +; * matches zero or more times, so whatever is being repeated may not be present at all, while + requires at least one occurrence. To use a similar example, `ca+t` matches “`cat`” (1 “a”), “`caaat`” (3 “a”s), but does not match “`ct`”.

The question mark character (?) matches either once or zero times. For example, `home-?brew` matches either “`homebrew`” or “`home-brew`.”

“Escaped” Characters Literals

Characters that have a special meaning inside regular expressions must be escaped:

- `\?` matches the `?` character.
- `\\` matches the `\` character.
- `\.` matches the `.` character.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

“Escaped” Characters Literals

Perhaps the most important metacharacter is the backslash, “\”. The backslash can be followed by various characters to signal various special sequences. As already stated, a number of characters have special meanings to regular expressions. A symbol with a special meaning can be matched. But to do so, you must prefix it with the backslash character (this includes the backslash character itself; to match one backslash in the target, your regular expression should include “\\”).

Some of these special sequences represent predefined sets of characters that are often useful, such as the set of digits, the set of letters, or the set of anything that is not a whitespace. The following predefined special sequences are available:

- `\d` matches any decimal digit; this is equivalent to the class `[0-9]`.
- `\D` matches any nondigit character; this is equivalent to the class `[^0-9]`.
- `\s` matches any whitespace character; this is equivalent to the class `[\t\n\r\f\v]`.
- `\S` matches any nonwhitespace character; this is equivalent to the class `[^\t\n\r\f\v]`.
- `\w` matches any alphanumeric character; this is equivalent to the class `[a-zA-Z0-9_]`.
- `\W` matches any nonalphanumeric character; this is equivalent to the class `[^a-zA-Z0-9_]`.

These sequences can be included inside a character class. For example, `[s,.]` is a character class that will match any whitespace character, or “,” or “.”

Grouping Regular Expressions

- **Grouping is useful to build units.**
- **The pattern**
`\/(Apache|MyApache|YourApache)\/Apache\/
conf` **matches the following paths:**
 - `/Apache/Apache/conf`
 - `/MyApache/Apache/conf`
 - `/YourApache/Apache/conf`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Grouping Regular Expressions

A regular expression can have literal characters in it, and also zero-width positional patterns. Each literal character or positional pattern is an atom in a regular expression. You may also group several atoms together into a small regular expression that is part of a larger regular expression. You might be inclined to call such a grouping a “molecule,” but normally it is also called an atom.

Introduction to `mod_rewrite`

`mod_rewrite` is a powerful tool to accomplish the following URL manipulations:

- Restricting access to directories and files
- Enabling conditional redirection of access
- Relocating servers, file systems, or directories
- Regenerating static pages based on the HTTP header variables

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Introduction to `mod_rewrite`

Oracle HTTP Server provides `mod_rewrite` as a tool for URL manipulations. A rewriting engine based on a regular-expression parser is used by `mod_rewrite` to rewrite requested URLs. The granularity of URL manipulations can be affected by the formats of server variables, environment variables, HTTP headers, and time stamps.

This module operates on the full URLs (including the path-info part) both in per-server context (`httpd.conf`) and in per-directory context (`.htaccess`), and can generate query-string parts on result.

Functioning of `mod_rewrite`

- The `mod_rewrite` module gets the rule sets from its configuration structure.
- Rule sets are:
 - Created on startup (for per-server context)
 - Created during the directory walk of the Apache kernel (for per-directory context)
- `mod_rewrite` processes the rules in the order they appear.
- The `TestString` is expanded before the condition is checked against `CondPattern`.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

`mod_rewrite` Rules Processing

Apache processes HTTP in phases. A hook for each of these phases is provided by Apache. API. `mod_rewrite` uses two of these hooks:

- **The URL-to-file name translation hook:** Used after the HTTP request has been read but before any authorization starts
- **The Fixup hook:** Triggered after the authorization phases and after the per-directory configuration files (`.htaccess`) have been read, but before the content handler is activated

`mod_rewrite` reads the configured rule sets from its configuration structure. Server-level rule sets are best configured at startup, whereas directory level rule sets are configured during the directory access of the kernel.

`mod_rewrite` loops through the rule sets rule by rule (`RewriteRule` directive) and when a particular rule matches, it loops through corresponding conditions (`RewriteCond` directives). First the URL is matched against the *Pattern* of each rule. When it fails, `mod_rewrite` immediately stops processing this rule and continues with the next rule. If the *Pattern* matches, `mod_rewrite` looks for corresponding rule conditions. If none are present, it just substitutes the URL with a new value, which is constructed from the string *Substitution* and goes on with its rule looping. However, if conditions exist, it starts an inner loop for processing them in the order that they are listed.

mod_rewrite Rules Processing (continued)

For conditions, a string *TestString* is created by expanding variables, back references, map lookups, and so on and then the *CondPattern* is matched against the expanded *TestString*. If the pattern does not match, the complete set of conditions and the corresponding rules fail. If the pattern matches, then the next condition is processed until no more conditions are available. If all conditions match, processing is continued with the substitution of the URL with *Substitution*.

When a request seeks a Uniform Resource Identifier (URI) with more than one “/” (http://yourserver//oldpath/rqstdsrc) the ‘//oldpath’ may bypass RewriteCond and RewriteRule directives if they are not correctly written.

For example, consider the following rule:

```
RewriteRule ^/oldpath(.*) /newpath$1 [R]
```

Requesting http://yourserver/oldpath/filea will redirect and return the page http://yourserver/newpath/fila as expected.

However, requesting http://yourserver//oldpath/filea will bypass this particular rule, potentially serving a page that you were not expecting it to. You can work around the problem by making sure that rules will capture more than one slash. To fix the example above, you can use the following replacement:

```
RewriteRule ^/+somepath(.*) /otherpath$1 [R]
```

mod_rewrite Directives

- **RewriteEngine** [on / off]
 - The RewriteEngine directive enables (on) or disables (off) the run-time rewriting engine.
 - If it is set to off, this module does no run-time processing.
- **RewriteOptions**
 - The RewriteOptions directive sets inheritance of the rule sets configuration.
- **RewriteLog**
 - The RewriteLog directive sets the name of the file to which the server logs rewriting actions.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

mod_rewrite Directives

RewriteEngine: The RewriteEngine directive enables or disables the run-time rewriting engine. If it is set to off, this module does no run-time processing at all. Use this directive to disable the module instead of commenting out all the RewriteRule directives.

Rewrite configurations are not inherited by default. This means that you need to have a RewriteEngine on directive for each virtual host in which you want to use it.

RewriteOptions: By specifying RewriteOptions 'inherit', you can force the configuration of the parent by the children. In a virtual-server context, this means that the maps, conditions, and rules of the main server are inherited. In a directory context, this means that conditions and rules of the .htaccess configuration of the parent directory are inherited.

RewriteLog: The RewriteLog directive sets the name of the file to which the server logs any rewriting action that it performs. If the name does not begin with a slash (/), then it is assumed to be relative to Server Root. To disable logging, either remove or comment out the RewriteLog directive or use RewriteLogLevel 0. Avoid setting the file name to /dev/null to prevent logging. This can slow down the server with no advantage.

RewriteLogLevel: The RewriteLogLevel directive sets the verbosity level of the rewriting logfile. The default level 0 means no logging, whereas 9 or more means that practically all actions are logged.

mod_rewrite Directives

- **RewriteBase:**
 - RewriteBase sets the base URL for per-directory rewrites.
- **RewriteCond:**
 - RewriteCond defines a rule condition.
 - This condition should be true before the RewriteRule is processed.
 - Precede a RewriteRule directive with one or more RewriteCond directives.
 - The rewriting rule is used only if its pattern matches the current state of the URI and if the RewriteCond conditions apply.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

mod_rewrite Directives (continued)

RewriteBase: The RewriteBase directive explicitly sets the base URL for per-directory rewrites. Rewrite rule can be used in per-directory configuration (.htaccess) files. To effectively perform the substitution, the base URL should be added to the server processing. By default, this prefix is the corresponding file path itself. But in most Web sites, URLs are not directly related to physical file name paths; in such cases you must use the RewriteBase directive to specify the correct URL prefix.

If the URLs of your Web server are not directly related to physical file paths, then you have to use RewriteBase in every .htaccess files where you want to use the RewriteRule directives.

Example: Assume the following per-directory config file:

```
# # /abc/def/.htaccess -- per-dir config file for directory /abc/def
# /abc/def is the physical path of /xyz,
RewriteEngine On
RewriteBase /xyz
RewriteRule ^oldstuff\.html$ newstuff.html
```

In the preceding example, a request to /xyz/oldstuff.html gets rewritten to the physical file /abc/def/newstuff.html.

mod_rewrite Directives

- **The RewriteRule directive defines the rewriting rule.**
- **The order of the rules is used when applying the rules at run time.**
- **The rule contains a regular expression that gets applied to the current URL.**
- **For details on regular expressions, use the manual pages: `man regex`.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Rewrite Directives

As with every module in OHS, mod_rewrite is also configured using directives. The major configuration points are:

- Rewrite rules that comprise search strings. The rewrite rules also direct the mod_rewrite module the way to act when the search string is matched.
- Rewrite conditions when the rewrite rules should be applied

Rewrite Rule: Tips

.	Any character	Single Character
[abc]	“a”, “b”, or “c”	Single Character
[a-z]	“a”, “b” ... “z”	Single Character
.*	Any number/character	Single or Many Characters
^	Beginning Position	
\$	End Position	

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Some Hints About Rewrite Rules

If there was a request for `/demo1/not_just_index.html`, all the above rewrite rules would have redirected the request to `/alldemos/index.html`, which may not be what you want. It is quite possible that you may want to redirect to the corresponding files in `/alldemos` as follows:

Request for	Redirect to
<code>/demo1/happy.html</code>	<code>/alldemos/happy.html</code>
<code>/demo1/go.jpg</code>	<code>/alldemos/go.jpg</code>
<code>/demo1/lucky.jpg</code>	<code>/alldemos/lucky.jpg</code>

Then you have to use substitution in your rewrite rule as follows:

```
RewriteRule ^/demo1(.*)$ /alldemos/$1 [R NC]
```

The following is an explanation of this rule:

Take the value of the expression (such as `happy.html`, `go.jpg`, and `lucky.jpg`) that appears after `demo1/` as variable (`$1`) and substitute it after `/alldemos/`.

Redirecting: Examples

- All the documents that are served by the Web server are moved to another subdirectory.

```
RewriteEngine on
RewriteRule ^/(.*)$ /newroot/$1 [R,L]
```

- To redirect from one directory to another:

```
RewriteEngine on
RewriteRule ^/oldloc(.*)$ /newloc/$1 [R,L]
```

- To redirect based on the time of the day:

```
RewriteEngine on
RewriteCond %{TIME_HOUR}%{TIME_MIN} >1800
RewriteCond %{TIME_HOUR}%{TIME_MIN} <0800
RewriteRule ^/Demo(.*)$ /Offtime$1 [NC,R]
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Redirecting: Examples

For redirecting requests from the DocumentRoot to a directory called newroot, set the following `mod_rewrite` directives:

```
RewriteEngine on
RewriteRule ^/(.*)$ /newroot/$1 [R]
```

For redirecting requests for files from one directory (`olddir`) to another (`newdir`), set the following directives:

```
RewriteEngine on
RewriteRule ^/olddir(.*)$ /newdir/$1 [R]
```

In each of these cases, you should ensure that the requested resources are indeed available in the redirected location. The `mod_rewrite` module does not ensure the existence of the requested resource in the new location.

In the third example, all requests from the Demo directory are directed to the Offtime directory depending on the time the request is received. All requests to Demo received after 6:00 p.m. and before 8:00 a.m. are redirected to the Offtime directory.

Summary

In this lesson, you should have learned how to:

- **Use regular expressions for pattern matching**
- **Enable `mod_rewrite`**
- **Configure `mod_rewrite` for business operations**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

D

Deploying PL/SQL Applications

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Objectives

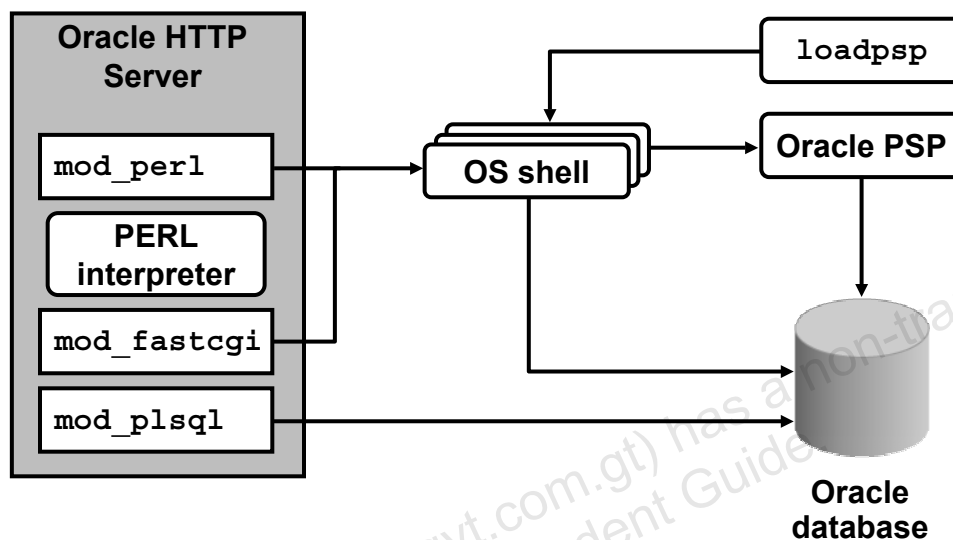
After completing this lesson, you should be able to do the following:

- **Configure `mod_plsql`**
- **Create a database access descriptor (DAD)**
- **Define authentication for PL/SQL applications**
- **Use Oracle PL/SQL Server Pages (PSPs)**
- **Configure `mod_perl` for the use of PERL**

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

Overview



Copyright © 2005, Oracle. All rights reserved.

Overview

In addition to the compiled Apache modules provided with Oracle HTTP Server, several of the standard modules have been enhanced, and Oracle-specific modules have been added. The following two modules are discussed in this lesson:

- **`mod_plsql`:** Routes PL/SQL requests to the Oracle PL/SQL service which, in turn, delegates the servicing of requests to PL/SQL programs. This module is recommended for PL/SQL program units.
- **`mod_perl`:** Forwards PERL application requests to the PERL interpreter that is embedded in Oracle HTTP Server. The primary advantages of using `mod_perl` are power and speed. The embedded PERL interpreter saves the overhead of starting an external interpreter, and the code-caching feature allows the server to run the code that has already been loaded and compiled.

mod_plsql Module

- **mod_plsql:**
 - Is an efficient PL/SQL interface for generating HTML
 - Uses standard database security features; users can be granted access to procedures but not to underlying tables through the Owner's/Definer's Rights Model
 - Enables you to reuse existing code and take advantage of in-house PL/SQL skills
 - Is productive; OracleAS Portal and Oracle Designer have PL/SQL generators
- **If you have many HTML pages, you can use Oracle PL/SQL Server Pages (PSPs) for rapid development of dynamic content.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

mod_plsql Module

mod_plsql enables Oracle Application Server to connect to an Oracle database server and execute stored procedures. Each mod_plsql request is associated with a database access descriptor (DAD), which is a named set of configuration values used for database access.

A DAD specifies information such as:

- The database alias (Oracle Net service name)
- A connect string if the database is remote
- A procedure for uploading and downloading documents

The PL/SQL procedures that are invoked from mod_plsql can perform some operations on the database and return the results to the user, or generate dynamic HTML pages containing data from the database. The procedure that mod_plsql invokes typically returns HTML data to the client.

To simplify this task, mod_plsql comes with the PL/SQL Web Toolkit, a set of packages that you can use in your stored procedure to get information about the request, construct HTML tags, and return header information to the client. By default, the PL/SQL Web Toolkit is installed in the SYS schema.

mod_plsql Module (continued)

You can produce the same results in the following ways:

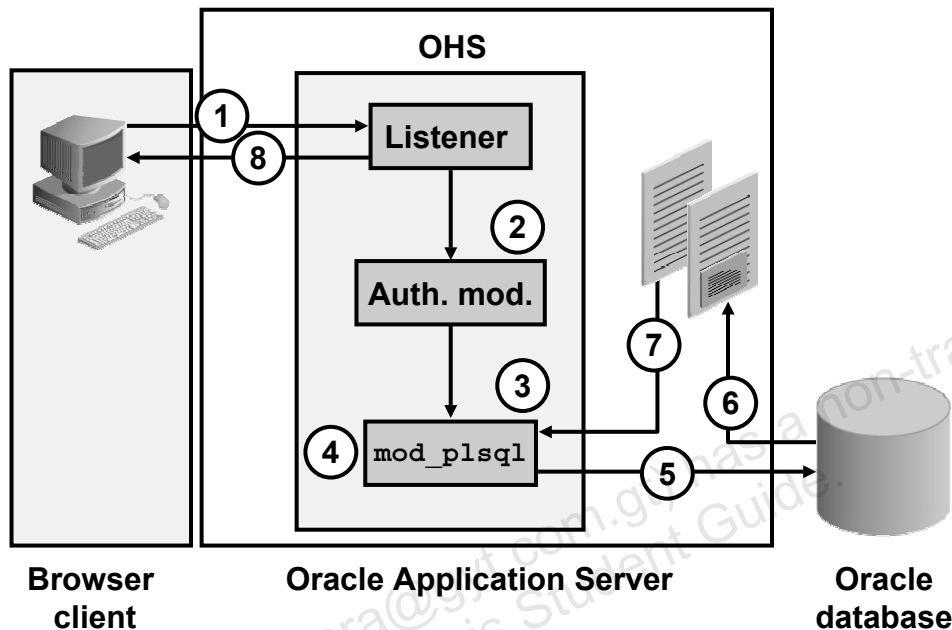
- Write an HTML page with embedded PL/SQL code and compile it as a PL/SQL server page. You can call procedures from the PL/SQL Web Toolkit, but not to generate every line of HTML output.
- Write a complete stored procedure that produces HTML tags by calling the HTP, HTF, and OWA_* packages in the PL/SQL Web Toolkit.

The key factors in choosing between these techniques are as follows:

- If you have a large body of HTML code and want to include dynamic content or make the HTML page the front end of a database application, then use PSPs.
- If you have a large body of PL/SQL code that produces formatted output, you may find it more convenient to produce HTML tags by setting your Print statements to call the HTP package of the PL/SQL Web Toolkit.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Communication Flow: The Path of HTTP Requests



Copyright © 2005, Oracle. All rights reserved.

Communication Flow: The Path of HTTP Requests

1. The browser sends a URL to the listener. The listener examines the URL and determines that the request is for a module (in this case, `mod_plsql`).
2. If authentication is required, the listener contacts an authorization module (such as `mod_auth` or `mod_oss1`) with the URL and browser credentials (such as authorization header, IP address, domain name, and SSL information).
3. The authorization module validates the request and returns the result to the required module.
4. `mod_plsql` uses the database access descriptor (DAD) configuration values to determine how to connect to the database.
5. `mod_plsql` connects to the database, prepares the call parameters, and invokes the PL/SQL procedure named in the URL request in the database.
6. The PL/SQL procedure generates an HTML page that can include dynamic data accessed from tables in the database, as well as static data.
7. The output from the procedure is returned by way of the response buffer to `mod_plsql`.
8. Oracle HTTP Server sends the response back to the client.

Enabling a PL/SQL Application

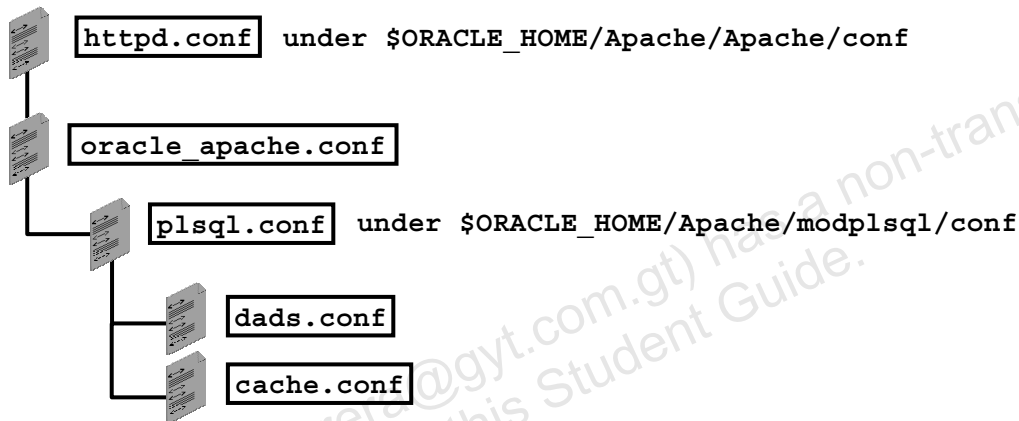
1. Configure the `mod_plsql` parameters.
2. Create a database access descriptor (DAD).
3. Restart Oracle HTTP Server.
4. Create a PL/SQL application.

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

mod_plsql Configuration Files

- The `oracle_apache.conf` file contains reference to other `mod_plsql` configuration files.
- The `httpd.conf` file includes reference to `oracle_apache.conf` file.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

mod_plsql Configuration Files

The installation of Oracle Application Server creates configuration files that you can edit, including the following, which affect `mod_plsql`. The primary Oracle HTTP Server configuration file is:

`$ORACLE_HOME/Apache/Apache/conf/httpd.conf`

The `httpd.conf` file contains an `Include` directive for:

`$ORACLE_HOME/Apache/Apache/conf/oracle_apache.conf`

The `oracle_apache.conf` file contains an `Include` directive for:

`$ORACLE_HOME/Apache/modplsql/conf/plsql.conf`

The `plsql.conf` file contains an `Include` directive for:

`$ORACLE_HOME/Apache/modplsql/conf/dads.conf`

`$ORACLE_HOME/Apache/modplsql/conf/cache.conf`

The configuration files are discussed in detail later in this lesson.

plsql.conf File

This file contains the main directives to load mod_plsql into Oracle HTTP Server:

```
LoadModule plsql_module
/home/oracle/infra/Apache/modplsql/bin/modplsql.so
<IfModule mod_plsql.c>
#
include
"/home/oracle/infra/Apache/modplsql/conf/cache.conf"
Include
"/home/oracle/infra/Apache/modplsql/conf/dads.conf"
...
</IfModule>
```

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

plsql.conf File

This file contains the `LoadModule` directive to load `mod_plsql` into Oracle HTTP Server and global settings for `mod_plsql`. This file also includes directives for `dads.conf` and `cache.conf`.

The directives are included in an `IfModule` block to make sure that they are only applied if the module is loaded successfully.

The `plsql.conf` file is organized like the `httpd.conf` file having different configuration sections, starting with a general section that contains directives that apply to the module in general or all database access descriptors (DADs). A DAD is a set of values that specify how `mod_plsql` connects to a database server to fulfill an HTTP request.

The `Include` directives belong to the cache or DAD setting sections that contain information only relevant for either PL/SQL caching or DADs.

dads.conf File

- The `dads.conf` file contains the configuration parameters for the PL/SQL database access descriptor (DAD).
- A DAD is a set of values that specify how `mod_plsql` connects to a database server to fulfill an HTTP request.

```
<Location /pls/plsqlapp>
    SetHandler pls_handler
    ...
</Location>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

dads.conf File

This file contains the configuration parameters for the PL/SQL database access descriptor (DAD). Besides the connection details, a DAD contains important configuration parameters for various operations in the database and for `mod_plsql` in general. Any Web-enabled PL/SQL application that uses the PL/SQL Web Toolkit must create a DAD to invoke the application.

Some typical PL/SQL applications that require DADs are:

- Oracle Application Server Portal
- Oracle Application Server Single Sign-On
- Any Oracle Application Server PL/SQL Cartridge application

Configuring mod_plsql

An example of a typical PL/SQL application DAD:

```
<Location /pls/plsqlapp>
  SetHandler pls_handler
  AllowOverride None

  PlsqlDatabaseUsername      scott
  PlsqlDatabasePassword      tiger
  PlsqlDatabaseConnectionString  host:port:service
  ...
  # PlsqlAuthenticationMode    Basic
</Location>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring mod_plsql

This is a configuration sample provided within the `dads.conf` file. You can access this file by navigating from the Application Server Control Web site to the Oracle HTTP Server Home page, and then by using the Advanced Properties link at the bottom of the page. On the Advanced Properties page, you are able to open the `dads.conf` file from the Configuration Files table.

You already know the `<Location>` container from a previous lesson and what directives are allowed. The following directives are Oracle specific:

- **PlsqlDatabaseUsername:** Specifies the username to use to log in to the database. This is a mandatory parameter, except for a DAD that sets `PlsqlAuthenticationMode` to `Basic` and uses dynamic authentication. For DADs using `SingleSignOn` as authentication mode, this parameter has to be the name of the schema owner.

Configuring mod_plsql (continued)

- **PlsqlDatabasePassword:** Determines the password to use to log in to the database. This is also a mandatory parameter except for a DAD that sets `PlsqlAuthenticationMode` to `Basic` and uses dynamic authentication. For DADs using Single Sign-On authentication, this parameter specifies the name of the schema owner.
- **PlsqlDatabaseConnectString:** Defines the database to which `mod_plsql` will connect to. The following formats are allowed to specify this parameter:
 - **ServiceNameFormat:** `HOST:PORT:SERVICE_NAME` format where `HOST` is the host name running the database, `PORT` is the port number the TNS listener is listening on, and `SERVICE_NAME` is the database service name
 - **SIDFormat:** `HOST:PORT:SID` format where `HOST` is the host name running the database, `PORT` is the port number the TNS listener is listening on, and `SID` is the database SID
 - **TNSFormat:** A valid TNS alias which resolves using Net8 utilities, such as `tnsping` and `SQL*Plus`
 - **NetServiceNameFormat:** A valid net service name which resolves to a connect descriptor. A connect descriptor is a specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.
- If the format argument is not specified, then `mod_plsql` assumes that the specified string is either in the `HOST:PORT:SID` format or resolvable by Net8. The differentiation between the two is made by the presence of the colon in the specified string.
- It is recommended that newer DADs do not use the `SIDFormat` syntax. This exists only for backward compatibility reasons. Use the new two-argument format for newly created DADs.
- **PlsqlAuthenticationMode:** Specifies the authentication mode to be used for allowing access through this DAD. The value can be one of the following:
 - **Basic:** The default value is `Basic`. For `Basic` mode, if you want to perform dynamic authentication, the DAD username/password parameters must be omitted.
 - **SingleSignOn:** For Oracle Application Server Portal, you must set this directive to `SingleSignOn`.
 - **GlobalOwa:** The earlier releases of Oracle applications use `GlobalOwa` mode.
 - **CustomOwa or PerPackageOwa:** The Custom Authentication modes (`GlobalOwa`, `CustomOwa`, `PerPackageOwa`) are used by very few PL/SQL applications.

If the DAD is not using the Basic authentication, then you must include a valid username/password in the DAD configuration.

For more information, refer to the *Oracle HTTP Server Administrator's Guide 10g Release 2 (10.1.2.0.2)*.

Obtaining Information About mod_plsql

Farm > Application Server: portal.edrsr16p1 >

HTTP_Server

Home Virtual Hosts **Administration**

Properties Inheritance

Server Properties
 MIME Languages
 MIME Types
 MIME Encodings
 PL/SQL Properties
 Advanced Server Properties

Home Virtual

mod_plsql Services

Page Refre

General			Related Link
Module Loaded	Yes		SQL Errors
Average Requests Per Hour	0		

Cache			HTTP Response Codes
	Session Cache	Content Cache	HTTP Response and Error Codes
Cache working	Up	Up	200 - OK
Requests Since Startup	0	0	201 - Created
Cache Hits(%)	0	0	202 - Accepted
Cache Misses (New)(%)	0	0	302 - Found
Cache Misses (Stale)(%)	0	0	304 - Not Modified
			400 - Bad Request
			401 - Unauthorized
			403 - Forbidden
			404 - Not Found
			500 - Internal Server Error
			503 - Service Unavailable
			Total

Obtaining Information About mod_plsql

To access the main mod_plsql page, perform the following steps:

1. Navigate to the Oracle HTTP Server Home page in Application Server Control. Click the Administration link to access the Administration page.
2. Click PL/SQL Properties. The mod_plsql Services page is displayed.

You can use this page to monitor the status of mod_plsql. You can also create and edit mod_plsql database access descriptors (DADs), and maintain logging and cache configuration settings from this page.

The page is divided into the following regions:

- **General:** Shows the status of mod_plsql service and the average requests per hour
- **HTTP Response Codes:** Provides categorywise information about the number of errors and response codes
- **Cache:** Provides the status of caching used by mod_plsql
- **Errors and Response Codes:** Links to the HTTP error and SQL error pages

Configuring DADs Using `dads.conf`

Database Connection

[Security, Document, Alias and Session](#)

[Advanced](#)

Edit DAD Database Connection

Database Access Descriptor Name

A unique name for your Database Access Descriptor. The name must not contain any special characters and must not exceed 64 characters.

DAD Name or Location

Database Connectivity Information

Username

Password

Connect String

Connect String Format

NLS Language

NLS Language should match the backend Database. Use the format <NLS_LANGUAGE>_<NLS_TERRITORY>.<NLS_CHARACTERSET>, e.g., American_America.AL32UTF8.

Default Page

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Configuring DADs Using `dads.conf`

To edit the `dads.conf` configuration file, perform the following steps:

1. Navigate to the Oracle HTTP Server Home page on Application Server Control. Click the Administration link to access the Administration page.
2. Click PL/SQL Properties. This opens the `mod_plsql` services page.
3. Click the required link in the DADs table to display the Edit DAD Database Connection page.

In order for changes to take effect, you may need to restart the HTTP Server.

DAD Creation Wizard

Create DAD Type

A Database Access Descriptor is required to connect mod_plsql to a database.

☒ Portal
Pre-configured for use by Portal.

☐ General
For use by any mod_plsql application.

Database Access Descriptor Name
A unique name for your Database Access Descriptor. The name must not contain any special characters.

DAD Name or Location
The Portal DAD Name or Location should be pre-pended with a /pls/

Database Connectivity Information

Username

Password

Connect String

Connect String Format

NLS Language

ORACLE

Copyright © 2005, Oracle. All rights reserved.

DAD Creation Wizard

To create a DAD using the mod_plsql page, perform the following steps:

1. Navigate to the Oracle HTTP Server Home page in Application Server Control. Click the Administration link to access the Administration page.
2. Click PL/SQL Properties. The mod_plsql services page is displayed.
3. Click Create in the DADs table to display the Create DAD page.
4. Select the type of DAD you want, such as Portal, and click Next.
5. Enter values for the following fields:
 - **DAD Name or Location:** Specifies the name of the database access descriptor
 - **Username:** Determines the database user you want to bind with this DAD. This user must exist in your database. If you do not provide a username here, the user trying to access the DAD will be prompted for this information.
 - **Password:** Specifies the password of the given database user
 - **Connection String:** Determines how to contact your database. The entry should be a TNS alias or in the form of *hostname:port:service*.
 - **Connect String Format:** Is used to specify what format you have specified in the Connect String field. You can select from the list of values.
6. Click OK to create the DAD.

Invoking a PL/SQL Application

`http://host:port/path/pack.proc?p1=1&p2=2`

http: Network protocol

Host: Machine name and domain

Port: HTTP listener port number

Path: Virtual path for DAD location

Proc: Stored procedure

Pack: Stored package

P1=1&p2=2: Parameters for procedure

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Invoking a PL/SQL Application

To invoke a PL/SQL stored procedure in a Web browser, the URL typically must be in the following format:

`protocol://host[:port]/path/
[package.]proc_name[?query_string]`

- `protocol` can be either `http` or `https`. For SSL, use `https`.
- `host` is the domain-qualified name of the machine where the Web server is running.
- `port` is the port at which the application server is listening. If the port is omitted, port 80 is assumed.
- `path` is the virtual path to handle PL/SQL requests that are mounted in a `<Location>` container for a specific DAD. This container also includes the connection information.
- `package` is the database PL/SQL package that contains stored procedures. If the package name is omitted, the procedure is a stand-alone procedure.
- `proc_name` specifies the name of the PL/SQL stored procedure to run. This must be a procedure and not a function. It can accept only IN arguments.
- `?query_string` specifies parameters (if any) for the stored procedure.

Invoking a PL/SQL Application: Example 1

Oracle HTTP Server is configured with `plsqlapp` as the DAD location, and the browser sends the following URL:

```
http://myServer:7778/plsqlapp/myproc?p=Hello
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Invoking a PL/SQL Application: Example 1

For example, assume that a Web server is configured to use the virtual path `/plsqlapp` for a specific DAD and the browser sends the following URL:

```
http://mysun.oracle.com:7778/plsqlapp/myproc?p=Hello
```

Then, the Web server running on `mysun.us.oracle.com` and listening at port 7778 handles the request. When the Web server receives the request, it passes the request to `mod_plsql`. The HTTP listener knows to do this because the virtual path `/plsqlapp` is configured by a location directive in the `dads.conf` file. That location is configured to invoke `mod_plsql` by including a `SetHandler` directive within the location directive. Next, `mod_plsql` uses the DAD included in that container, and runs the `myproc` procedure.

In the example in the slide, `p=Hello` will be passed as an input parameter to the `myproc` procedure and will be accepted as “Hello.”

Invoking a PL/SQL Application: Example 2

- Specify a URL without providing a schema, package, or a stored procedure name:

```
http://myServer:7778/pls/plsqlapp
```

- The location container that enables this behavior:

```
<Location /pls/plsqlapp>  
    SetHandler pls_handler  
    ...  
    PlsqlDefaultPage scott.home  
</Location>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Invoking a PL/SQL Application: Example 2

A URL can be successfully retrieved without specifying a schema, package, or a stored procedure name when the location container associated with the virtual path `/pls/plsqlapp` contains the `PlsqlDefaultPage` directive. This specifies the default procedure to call if none is specified in the URL. In the example in the slide, a program unit called `home`, which is stored in the database schema `scott` will be called because it is the default.

OracleAS Portal uses the `PlsqlDefaultPage` as shown in this example.

Preventing the Execution of PL/SQL Procedures

To exclude access to URLs containing specific packages, add the following in the `dads.conf` file:

```
PlsqlExclusionList sys.*
PlsqlExclusionList dbms_*
PlsqlExclusionList utl_*
PlsqlExclusionList owa_*
PlsqlExclusionList owa.*
PlsqlExclusionList http.*
PlsqlExclusionList htf.*
PlsqlExclusionList oracle.private.*
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Preventing the Execution of PL/SQL Procedures

Because some procedures in the `dbms_*` packages allow access to sensitive information, it is generally not a good idea to allow access from a browser. Because these procedures, such as the `dbms_*` packages, `utl_*` packages, and all packages under the `SYS` schema, pose a security risk when they are executed through the Web browser, the default setting prevents effectively the execution of the PL/SQL procedures granted to `public` in the database.

Such packages are intended only for the PL/SQL application developer.

The `PlsqlExclusionList` directive specifies a pattern in `dads.conf` for excluding certain procedures, packages, or schema names from being directly executed from a browser. This is a multiline directive in which each pattern occupies one line. The pattern is not case sensitive, and can accept simple wildcards such as `*`, `?`, and `[a-z]`. The default patterns excluded from direct URL access are `sys.*`, `dbms_*`, `utl_*`, `owa_*`, `owa.*`, `http.*`, and `htf.*`.

Setting this directive to `NONE` will disable all protection. This is not recommended for a production Web site; however, it is occasionally used for debugging purposes.

The example in the slide excludes access to the default pattern and also to URLs containing `oracle.private.*`. This will not allow access to anything within the private package owned by the `oracle` schema.

Preventing the Execution of PL/SQL Procedures (continued)

The setting `PlsqlExclusionList oracle.private.*` will only exclude access to URLs containing `oracle.private.*`. The system defaults will no longer be protected. (This is normally done for backward compatibility only.)

As stated, the default setting prevents access to the above-listed packages without specifying `PlsqlExclusionList` explicitly. So, if you want to exclude access to more packages, procedures, or a schema, as compared with `oracle.private.*`, then you must include a `PlsqlExclusionList` directive for each package, procedure, or schema that you want to exclude.

In addition to URL patterns specified with this directive, `mod_plsql` also excludes any URLs containing special characters such as tabs, new lines, carriage returns, single quotation marks, or the backslash. This cannot be changed.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

mod_plsql Caching

- **mod_plsql can cache repeatedly used SQL statements and credentials to improve performance.**
- **Applications such as OracleAS Portal use this feature.**
- **mod_plsql uses two types of caching:**
 - **PL/SQL cache**
 - **Session Cookie cache**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

mod_plsql Caching

mod_plsql has a mechanism to enable caching of repeatedly used SQL statements and credentials to improve performance. This is different from the caching that Web Cache provides. Applications such as OracleAS Portal benefit from this feature. There are two types of caching being used by mod_plsql.

- **PL/SQL cache:** Used to cache dynamically generated contents that do not change often. Applications using the OWA_CACHE package, such as OracleAS Portal, use this feature to improve performance and take some load off the database.
- **Session Cookie cache:** Used to cache the cookie value generated by a single sign-on server for a particular session. By enabling this feature, a round-trip to the database to obtain a user's credentials is avoided, thereby, improving performance. Only applications that use single sign-on benefit from this feature.

cache.conf file

This file contains the cache settings for mod_plsql:

```
# Turn caching on or off
PlsqlCacheEnable On
# Set directory to write the cache files
PlsqlCacheDirectory /ias20/Apache/modplsql/cache
# Set the total size of the cache, this parameter #
takes bytes as the value, for 25 Megabyte:
PlsqlCacheTotalSize 25600000
PlsqlCacheCleanupTime Everyday 2:00
...
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

cache.conf file

This file specifies the characteristics of the mod_plsql caching system. Directives that are used in the cache.conf file include the following:

- **PlsqlCacheEnable:** Enables mod_plsql caching
- **PlsqlCacheDirectory:** Specifies the directory where cache files are written out. The owner of the httpd child processes must have write permission to this directory. When using with OracleAS Portal, this also points to a directory that is shared by mod_plsql and the OracleAS Portal file system cache. If you are running OracleAS Portal and change PlsqlCacheDirectory, then make sure that the OracleAS Portal configuration is also changed appropriately.
 - For PL/SQL cache, all cache files are created under a directory called plsql relative to a specified caching directory.
 - For Session Cookie cache, all cache files are created under a directory called session relative to a specified caching directory. This directory must exist or Oracle HTTP Server will not start.

cache.conf file (continued)

- **PlsqlCacheTotalSize:** Limits the amount of space the cache is allowed to use. Both PL/SQL cache and Session Cookie cache share this cache space. Note that this setting is not a hard limit. It might exceed the limit temporarily during normal processing.
- **PlsqlCacheCleanupTime:** Specifies the time to start the cleanup of the cache storage. This setting defines the exact day and time in which cleanup should occur. The frequency can be set as daily, weekly, and monthly:
 - To define daily frequency, the keyword “Everyday” is used. The cleanup starts everyday at the time defined. For example, “Everyday 2:00.” This causes the cleanup to happen everyday at 2 a.m. (local time) in the morning.
 - To define weekly frequency, the days of the week such as “Sunday”, “Monday”, “Tuesday”, and so on are used. For example, “Wednesday 15:30.” This causes the cleanup to happen every Wednesday at 3:30 p.m. (local time).
 - To define monthly frequency, the keyword “Everymonth” is used. The cleanup starts at the Saturday of the month at the time defined. For example, “Everymonth 23:00.” This causes the cleanup to happen the first Saturday of every month at 11:00 p.m. (local time).
- **PlsqlCacheMaxSize:** Specifies the maximum possible size of a cache file. This setting is to prevent the case in which one file can fill up the entire cache. In general, it is recommended that this be set to about 3 percent of the total cache size.

Troubleshooting

If you have problems connecting to the database:

- **Ensure that the network connection is working**
- **Ensure that the TNS listener and database are running**
- **Verify that the configured connection goes through using OracleNet, or some other tool to connect directly to the database**
- **Check the username and password information in the DAD**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Troubleshooting

If you have problems connecting to the database:

- Make sure that you can ping the machine hosting the database from a different machine
- Ensure that the TNS listener and the database are running (this information can be obtained using Application Server Control)
- Verify, using OracleNet or some other tool, that you can connect to the database using the configured connection
- Make sure that the username and password information provided for the DAD are correct. Try to establish a database session directly using these credentials, for example, `myserver.us.oracle.com:1521:ORCL`, or use the `tnsping` utility to verify your connection information.

PL/SQL Server Pages

Example: show_emp_simple.psp

```
<%@ page language="PL/SQL" %>
<%@ plsql procedure="show_emp_simple"%>
<HTML>
<HEAD><TITLE>Show Contents of HR.EMPLOYEES (Complete
Dump) </TITLE></HEAD>
<BODY>
<%
    declare
        dummy boolean;
    begin

        dummy:=owa_util.tableprint('employees','border=1');
    end;
%>
</BODY>
</HTML>
```

ORACLE

Copyright © 2005, Oracle. All rights reserved.

PL/SQL Server Pages

Web administrators are typically not involved in creating or uploading PL/SQL Server Pages (PSPs), but they should understand the process for troubleshooting purposes. To create and upload PSPs, perform the following steps:

1. Create a PSP file with the extension `.psp`. It can contain whatever content you like, with text and tags interspersed with PSP directives, declarations, and scriptlets.
 - In the simplest case, this file is nothing more than an HTML file. Compiling it as a PSP produces a stored procedure that outputs the same HTML file as using the PL/SQL Web Toolkit.
 - In the most complex case, it is a PL/SQL procedure that generates all the content of the Web page, including the tags for title, body, and headings.
 - In the typical case, it is a mix of HTML (providing the static parts of the page) and PL/SQL (filling in the dynamic content).
2. Load the PSP into the database as a stored procedure:
`$loadpsp -replace -user hr/hr@t9 show_emp_simple.psp`
where `hr` is the database user who will become the owner of the generated PL/SQL stored procedure called `show_emp_simple`, and `t9` is a TNS alias.
3. Access the page with a URL. For example:
`http://mysun.us.oracle.com:7777/pls/hrdad/show_emp_simple`

Oracle Application Server 10g R2: Administration I D-25

Overview of the `mod_perl` Module

- `mod_perl` is a built-in component of Oracle HTTP Server.
- `mod_perl` integrates a complete PERL interpreter (version 5.004).
- With `mod_perl`, you can run PERL CGI without loading a PERL interpreter every time.
- `mod_perl` works by providing a `perl_script` handler.
- Access control or authentication can be done by a `mod_perl` handler.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Overview of the `mod_perl` Module

The integration of a complete PERL interpreter allows Oracle HTTP Server to run PERL CGI scripts without loading a fresh interpreter each time.

`mod_perl` works by providing a `perl_script` handler that can be associated with directories and file extensions with `SetHandler` and `AddHandler`. This is the way any `mod_perl` handler can be assigned to the `PerlHandler` directive. In addition, any stage of Oracle HTTP Server processing, such as access control or authentication, can be handed over to a `mod_perl` handler.

The disadvantage of `mod_perl` is that it is a large module, because it contains the complete PERL language interpreter; therefore, it causes Oracle HTTP Server to consume more memory. However, the performance gains possible from using `mod_perl` to run CGI scripts, or converting those scripts into more efficient scripts, can be considerable.

The primary `mod_perl` interface is its `Perl*Handler` directives. A default configuration can be found in `httpd.conf`. Each stage of the Oracle HTTP Server processing can be reached with the corresponding handler directive.

Note: For further information about handlers associated with `mod_perl`, visit <http://perl.apache.org/guide/>.

Summary

In this lesson, you should have learned how to:

- Access the configuration files
- Configure `mod_plsql` for PL/SQL applications
- Create a database access descriptor (DAD)
- Specify authentication for PL/SQL applications
- Configure `mod_perl` for the use of PERL

ORACLE

Copyright © 2005, Oracle. All rights reserved.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

Introduction to Linux

ORACLE®

Copyright © 2005, Oracle. All rights reserved.

What Is Linux?

- **Linux is a UNIX-based operating system, created by Linus Torvalds at the University of Helsinki in Finland.**
- **It was developed under the GNU General Public License, allowing source code to be freely available.**
- **Each distribution was developed for a particular purpose.**
- **TUX, the penguin, is the official mascot of Linux.**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is Linux?

Linux is an operating system that was initially created as a hobby by a student, Linus Torvalds, at the University of Helsinki in Finland. Torvalds had an interest in Minix, a small UNIX system, and decided to develop a system that exceeded the Minix standards. He began his work in 1991 when he released version 0.02 and worked steadily until 1994 when version 1.0 of the Linux kernel was released.

Linux is developed under the GNU General Public License and its source code is freely available to everyone. As a result, a number of companies, organizations, and individuals have developed their own “versions” of the Linux operating system, known as distributions.

Each distribution, with associated programs and utilities, was developed for a particular purpose, for example, on computers that receive heavy traffic (such as Web page servers), where security is a priority, or on top of an existing operating system (such as Windows) so that people can try out Linux under familiar conditions.

Although Linux is technically only the kernel, it is commonly considered to be all of the associated programs and utilities of a distribution.

Linux has an online manual containing descriptions for all commands (see the man utility).

What Is Oracle's Strategy on Linux?

The following distributions are certified and supported by Oracle:

- **Red Hat Enterprise Linux AS and ES**
- **UnitedLinux, which includes the following products from Conectiva, SCO, SuSE, and TurboLinux:**
 - **Conectiva Linux Enterprise Edition powered by UnitedLinux**
 - **SCO Linux Server 4.0 powered by UnitedLinux**
 - **SuSE Linux Enterprise Server 8 (SLES 8) powered by UnitedLinux**
 - **TurboLinux Enterprise Server 8 powered by UnitedLinux**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is Oracle's Strategy on Linux?

Oracle is fully committed to supporting the Linux operating system. In fact, Oracle was the first commercial database available on Linux. By supporting Linux with Oracle's industry-leading products, Oracle Corporation enables customers to deploy enterprise-class solutions on the lowest cost hardware and operating system infrastructure. With technical contributions to enhance Linux, with direct support of the key Linux operating systems, and with strategic partnerships, Oracle is offering an Unbreakable Linux platform for customers to safely deploy Linux in a mission-critical environment. Oracle's delivery of a complete solution, including direct technical support of the operating system, is critical to the customer's success.

Red Hat Enterprise Linux AS, version 2.1

Red Hat has been working with Oracle Corporation to provide a more reliable and scalable platform for enterprise Linux users, which resulted in Linux AS, version 2.1. It includes many of the same packages as Red Hat 7.2, but also includes enhancements for enterprise features.

UnitedLinux

UnitedLinux is the result of a consortium of Linux vendors. It is based on the SuSE kernel and supports asynchronous input/output.

Oracle Application Server 10g R2: Administration I E-3

File System and Basic Directory Structure

In Linux, there are directories, subdirectories, and files, but everything is really just a file.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

File System and Basic Directory Structure

Every Linux user has a login username and password. Each user is provided with a separate workspace. In Linux, there are directories, subdirectories, and files, but everything is really just a file. Some key directories are:

- **The /bin directory:** The /bin directory contains programs, also known as binary files.
- **The /boot directory:** The /boot directory contains the Linux kernel.
- **The /dev directory:** The /dev directory contains the devices that your system uses or can use. Everything is considered a file in Linux, so your hard disk is kept track of as a file that sits there. Your hard drive will be known as /dev/hda.
- **The /etc directory:** The /etc directory contains most of the configuration files for Linux.

File System and Basic Directory Structure (continued)

- **The /lib directory:** The /lib directory contains library files. Linux stores library files here for systemwide shared access to libraries.
- **The /root directory:** The /root directory is a restricted area for all users except those with root privileges. Root privilege allows users to perform all system functions. See the su command for instructions on obtaining root privileges.
- **The /sbin directory:** The /sbin directory contains programs (binary files) used by root.
- **The /tmp directory:** The /tmp directory is used to store temporary files.
- **The /usr directory:** The /usr directory contains files and programs that are used by all users on the system.
- **The /var directory:** The /var directory is for certain files that may change their sizes (that is, variable size); for example, databases or incoming e-mail from an e-mail server.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Shell Commands

- **Environment-based commands**
- **Information-based commands**
- **File system commands**
- **Common `vi` editing commands**
- **Common FTP communication commands**
- **Archive utilities**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Shell Commands

All operating systems use a shell to get commands from the keyboard to the computer. The most popular shell used for Linux is the bash shell; bash means “Bourne Again Shell.” It is a free version of the Bourne shell.

For quick reference, the commands are divided as follows:

- Environment-based commands
- Information-based commands
- File system commands
- Common `vi` editing commands
- Common FTP communication commands
- Archive utilities

Environment-Based Commands

- **date**
- **df**
- **du**
- **echo**
- **env**
- **exit**
- **export**
- **free**
- **ifconfig**
- **kill**
- **login**
- **logout**
- **ps**
- **su**
- **top**
- **uname**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Environment-Based Commands

The usage for environment-based commands is as follows:

- **date: Display current date and time**
 - Usage: **date**
- **df: Display disk space used and available for each file system**
 - Usage: **df**
- **du: Display disk space usage for each file of the current directory**
 - Usage: **du**
- **echo: Print a line of text – used to display an environment variable setting**
 - Usage: **echo \$ORACLE_HOME** (Displays the setting for the ORACLE_HOME environment variable)
- **env: Display all environment variable settings**
 - Usage: **env**
- **exit: Log out from a session (see also the su command)**
 - Usage: **exit**

Environment-Based Commands (continued)

- **export:** Set environment variables
 - Usage: `export ORACLE_HOME=/home/oracle/infra` (Sets the `ORACLE_HOME` environment variable)
- **free:** Display amount of free and used memory
 - Usage: `free`
- **ifconfig:** Show the network status
 - Usage: `ifconfig`
- **kill:** Stop a process (see also the `ps` command)
 - Usage: `kill -9 pid` (Where *pid* is the process ID)
- **login:** Log in to a system and change the environment to the login user
 - Usage: `login` (You are prompted for the user name to log in to)
- **logout:** Log out of the system
 - Usage: `logout`
- **ps:** Show currently running processes
 - Usage: `ps -ef | grep keyword` (Displays all processes containing *keyword*)
- **su:** Modify user and group ID
 - Usage: `su root` (You are prompted for the root password to have root privileges)
- **top:** Display top CPU processes
 - Usage: `top` (Use “q” to quit the display)
- **uname:** Print system information
 - Usage: `uname -a` (to print all system information)

Information-Based Commands

- `>`
- `>>`
- `|`
- `cat`
- `diff`
- `file`
- `find`
- `grep`
- `info`
- `less`
- `ls`
- `man`
- `more`
- `pwd`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Information-Based Commands

The usage for information-based commands is as follows:

- **>: Redirect output**
 - Usage: `ls > filename` (Lists all files in a directory and writes them to a file called *filename*. If *filename* already exists, the contents are overwritten, otherwise a new file is created.)
- **>>: Append contents**
 - Usage: `ls >> filename` (The output is written to the end of *filename*. If *filename* does not already exist, it is created.)
- **|: A “pipe” for redirecting the output of a command to another command**
 - Usage: `ps -ef | grep keyword` (Displays all processes containing *keyword*)
- **cat: Concatenate files and print on standard output**
 - Usage: `cat filename` (Displays the contents of *filename* to the screen)
- **diff: Find the differences between two files**
 - Usage: `diff file1 file2` (Displays the difference between *file1* and *file2*)

Information-Based Commands (continued)

- **file:** Determine file type
 - Usage: `file filename` (Displays the file type, for example, text or executable)
- **find:** Find files
 - Usage: `find -name *oracle*` (Finds all files containing “oracle”)
 - `find -mmin -10` (Finds all files created in the last 10 minutes)
- **grep:** Find words in files
 - Usage: `grep -ir 'oracle' filename` (Searches for “oracle” in *filename*, ignoring case (the `-i` directive) and searching directories (the `-r` directive))
- **info:** Provide information on a specified topic
 - Usage: `info ls` (Displays an information page with multiple topic nodes)
 - `Del/Space` (Moves to the previous/next page within the current topic)
 - `n/p` (Moves to the next/previous topic node)
 - `m topicname` (Moves to a specific topic)
- **less:** Display contents of a file, allowing backward and forward scrolling
 - Usage: `less filename` (Use “f” to move forward, “b” to move backward, and “q” to quit.)
 - `diff file1 file2 | less` (Displays the difference between two files and pipe the results through “less”)
- **ls:** List storage (that is, display the contents of the current directory)
 - Usage: `ls -al` (Lists all files in the current directory)
 - `ls -al *.html` (Lists all HTML files in the current directory)
- **man:** Display a manual page
 - Usage: `man find` (Displays the manual for the find command)
- **more:** Display contents of a file
 - Usage: `more filename` (Uses SPACE to move forward and “q” to quit)
 - `ls -al | more` (Lists the directory and pipe the results through more)
- **pwd:** Print working directory
 - Usage: `pwd` (Shows the full path of the current directory)

File System Commands

- `cd`
- `chmod`
- `chown`
- `cp`
- `mkdir`
- `mv`
- `rm`
- `rmdir`

ORACLE

Copyright © 2005, Oracle. All rights reserved.

File System Commands

The usage for file system commands is as follows:

- **cd: Change directory**
 - Usage: `cd /dir/files` (Changes into the `/dir/files` directory)
 - `cd directoryname` (Changes into the *directoryname* directory located under the current directory)
 - `cd ../directoryname` (Changes into the *directoryname* directory located above the current directory)
- **chmod: Change the permissions on a file or directory**
 - Usage: `chmod 7777 filename` (Grants read, write, and execute permission to all users accessing *filename*; computes the octal number as follows: read (4), write (2), execute (1) and identify the user's access positionally, through the letters "ugo" where (u) is the user who owns it, (g) is for other users in the file's group, (o) is for other users not in the file's group, or (a) for all users)

File System Commands (continued)

- **chown: Change the owner or group for a file**
 - Usage: `chown owner:group filename` (Changes the owner and group of *filename* to the current user)
- **cp: Copy files**
 - Usage: `cp file ../newFile` (Copies *file* into the directory above the current directory and renames it as *newFile*)
- **mkdir: Make directory**
 - Usage: `mkdir newdir` (Makes a new directory *newdir* under the current directory)
 - `mkdir /usr/newdir` (Makes a new directory *newdir* under the `/usr` directory)
- **mv: Move (rename) files**
 - Usage: `mv oldName newName` (Renames *oldName* as *newName*)
- **rm: Remove files**
 - Usage: `rm filename` (Removes *filename* from the current directory)
 - `rm *old` (removes all files ending in “old”)
- **rmdir: Remove directories**
 - Usage: `rmdir directoryname` (Removes *directoryname* from the current directory)

Common `vi` Editing Commands

`vi` is a full-screen text editor with two modes:

- **Input mode:** Text is entered in the document by inserting or appending.
- **Command mode:** You can move within the document and merge, search, and cut lines.

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Common `vi` Editing Commands

The `vi` program is a full-screen text editor, which has two modes:

- **Input mode:** Text is entered in the document by inserting or appending.
- **Command mode:** You can move within the document and merge, search, and cut lines.

Common `vi` Commands

- `ESC`: Exits input mode and puts you in command mode
- `h, j, k, l`: Left, down, up, right (or use the arrow keys)
- `w, W, b, B`: Forward, backward by word
- `0, $`: First, last position of current line
- `/pattern`: Search forward for pattern
- `?pattern`: Search backward for pattern
- `n,N`: Repeat last search in same, opposite direction
- `x`: Delete character
- `dd`: Delete current line
- `D`: Delete to end of line

Common vi Editing Commands (continued)

- `dw`: Delete word
- `p`, `P`: Put deleted text before, after cursor
- `u`: Undo last command
- `.`: Repeat the last command
- `i`, `a`: Insert text before, after cursor [Puts you into input mode]
- `o`, `O`: Open new line for text below, above cursor [Puts you into input mode]
- `ZZ`: Save file and quit
- `:w`: Save file
- `:q!`: Quit, without saving changes

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Common FTP Communication Commands

Command	Description
<code>ftp hostname.com</code>	To connect to <code>hostname.com</code>
<code>type binary</code>	To set the type for binary files
<code>type ascii</code>	To set the type for ASCII files
<code>get filename</code>	To get a file from the FTP site
<code>put filename</code>	To put a file on the FTP site
<code>mget *jar</code>	To get all JAR files from FTP site
<code>mput *war</code>	To put all WAR files on FTP site
<code>prompt</code>	To shut off/turn on prompting

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Complete FTP Communication Commands

The complete list of FTP commands is as follows:

- `!`: Escape to the shell
- `?`: Print local help information
- `append`: Append to a file
- `ascii`: Set ASCII transfer type
- `bell`: Beep when command completed
- `binary`: Set binary transfer type
- `bye`: Terminate FTP session and exit
- `cd`: Change remote working directory
- `close`: Terminate FTP session
- `delete`: Delete remote file
- `debug`: Toggle debugging mode
- `dir`: List contents of remote directory
- `disconnect`: Terminate FTP session

Complete FTP Communication Commands (continued)

- `get`: Receive file
- `glob`: Toggle metacharacter expansion of local file names
- `hash`: Toggle printing `#' for each buffer transferred
- `help`: Print local help information
- `lcd`: Change local working directory
- `literal`: Send arbitrary FTP command
- `ls`: List contents of remote directory
- `mdelete`: Delete multiple files
- `mdir`: List contents of multiple remote directories
- `mget`: Get multiple files
- `mkdir`: Make directory on the remote machine
- `mls`: List contents of multiple remote directories
- `mput`: Send multiple files
- `open`: Connect to remote TFTP
- `prompt`: Force interactive prompting on multiple commands
- `put`: Send one file
- `pwd`: Print working directory on remote machine
- `quit`: Terminate FTP session and exit
- `recv`: Receive file
- `remotehelp`: Get help from remote server
- `rename`: Rename file
- `rmdir`: Remove directory on the remote machine
- `send`: Send one file
- `status`: Show current status
- `trace`: Toggle packet tracing
- `type`: Set file transfer type
- `user`: Send new user information
- `verbose`: Toggle verbose mode

Archive Utilities

The following archive utilities are available for Linux:

- **tar**
- **gzip and gunzip**
- **bzip2 and bunzip2**
- **zip and unzip**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Archive Utilities

The following archive utilities are available for Linux:

tar

tar stands for tape archive and was originally designed for tape backups, but is used to create a tar file anywhere on the file system. The **tar** utility creates one “tar file” (also known as a “tarball”) out of several files and directories. A tar file is not compressed. It is just a heap of files assembled together in “one container.” So, the tar file takes up the same amount of space as all the individual files combined, plus a little extra. A tar file can be compressed by using **gzip** or **bzip2**.

The following are some examples:

- **tar -cf backup.tar /home/ftp/pub**: Creates a tar file named **backup.tar** from the contents of the **/home/ftp/pub** directory
- **tar -tvf example.tar**: Lists the contents of **example.tar** to the screen
- **tar -xvf example.tar**: Extracts the contents of **example.tar** and displays the files as they are extracted
- **tar -zxvpf my_tar_file.tar.gz**: Unzips the tar file and then extracts the contents

Archive Utilities (continued)

The options are defined as follows:

- `-c`: Create a new archive
- `-f`: Use the file in question (required option)
- `-p`: Preserves dates, permissions of the original files
- `-t`: List the contents of an archive
- `-v`: Verbose (that is, tar tells you what files it is extracting)
- `-x`: Extract the files from the tarball
- `-z`: Unzip the file first

gzip and gunzip

The `gzip` utility compresses a tar file, reducing the amount of space required to store the archived tar file. The `gunzip` utility (or `gzip -d`) expands (decompresses) the `gzip` file. `gunzip` recognizes the special extensions `.tgz` and `.taz` as shorthands for `.tar.gz` and `.tar.Z`, respectively. The following are some examples:

- `gzip my_tar_file.tar`: Compresses the tar file and renames it with a `.gz` extension
- `gzip -d my_tar_file.tar.gz`: Unzips the tar file
- `gunzip my_tar_file.tar.gz`: Unzips the tar file

bzip2 and bunzip2

The `bzip2` utility compresses files using the Burrows-Wheeler block sorting text compression algorithm and Huffman coding. Compression is generally considerably better than that achieved by more conventional compressors. The `bunzip2` utility (or `bzip2 -d`) decompresses a `bzip2` file.

The following are some examples:

- `bzip2 *`: Compresses each file in the current directory and renames the file with a `.bz2` extension
- `bunzip2 my_file.bz2`: Decompresses the `my_file.bz2` file
- `bzip2 -d my_file.bz2`: Decompresses the `my_file.bz2` file

zip and unzip

`zip` is a compression and file-packaging utility for UNIX, VMS, MS-DOS, OS/2, Windows NT, Minix, Atari and Macintosh, Amiga, and Acorn RISC OS. It is compatible with PKZIP (Phil Katz's ZIP for MS-DOS systems). The companion program "unzip" unpacks zip archives. The `zip` and `unzip` utilities can work with archives produced by PKZIP, and PKZIP and PKUNZIP can work with archives produced by `zip`.

Archive Utilities (continued)

The zip program puts one or more compressed files into a single zip archive, along with information about the files (name, path, date, time of last modification, protection, and check information to verify file integrity). An entire directory structure can be packed into a zip archive with a single command. Compression ratios of 2:1 to 3:1 are common for text files.

The following are some examples:

- `zip my_files *`: Creates a compressed file named `my_files.zip`, containing all of the files in the current directory
- `unzip my_files.zip`: Expands the zip file within the current directory

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Shortcuts and Tips

- **Case sensitivity**
- **The clear utility**
- **[Shift] + [Page Up]/[Page Down]**
- **Tab**
- **Color coding**
- **The touch utility**
- **Web sites**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Shortcuts and Tips

- Linux is a case-sensitive operating system. You must enter the case correctly.
- In the terminal window, to clear the contents, enter `clear`.
- To scroll up, press [Shift] + [Page Up].
- To scroll down, press [Shift] + [Page Down].
- Press [Tab] to complete the remainder of the text. (Linux beeps to let you know that is as far as it can complete the text; you now need to add more characters to resolve ambiguities.)
- When you enter `ls -al`, the result is color coded. Blue is for directories.
- To create a file, enter `touch filename`. If file name does not exist, it gets created. If file name already exists, touch alters its timestamp to the current time. Please note, in Linux, we cannot easily name files with spaces in them, therefore, words should use underscores or a capital letter to separate them. For example, “touch my file” does not work. You must write either “touch myFile” or “touch my_file.” This applies to creating directories as well.
- The following are helpful Linux Web sites:
 - www.oracle.com/linux
 - www.linux.org
 - www.linux.com
 - www.redhat.com

Oracle Application Server 10g R2: Administration I E-20

Introduction to OracleAS Portal 10.1.4

ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is OracleAS Portal?

OracleAS Portal:

- **Is a component of Oracle Application Server**
- **Can be accessed from a Web browser**
- **Offers organized and personalized views of Web content through portal pages**
- **Provides a secure and manageable framework for accessing distributed software services and information resources**
- **Supports data-driven portlets and content publishing**
- **Provides deployment architecture**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is OracleAS Portal?

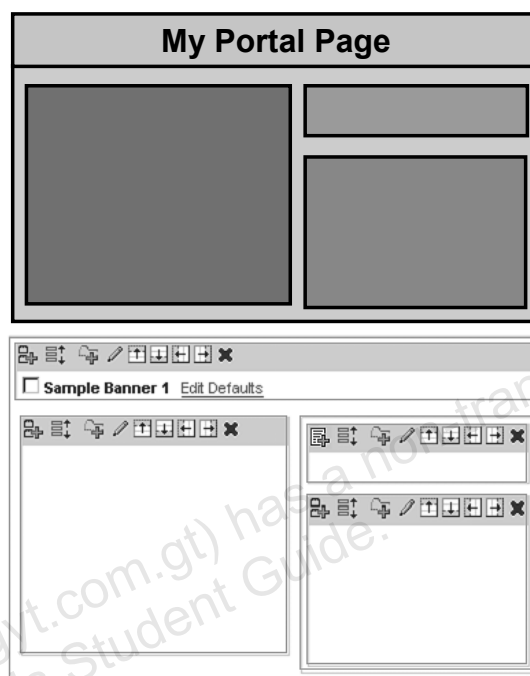
OracleAS Portal, a component of Oracle Application Server (OracleAS), is a complete solution for building, deploying, and maintaining self-service, integrated portals. Using OracleAS Portal, you can connect to applications, retrieve the information that you need, and customize the display of that information in the required manner. You can also use OracleAS Portal to organize and integrate new and existing Web applications and sites from the intranet and extranet consistently.

OracleAS Portal incorporates a portal creation and deployment framework that defines Web information sources as information components, assembles these components within a page, and supports their customization by individual users or groups of users. Web pages that are created and deployed with OracleAS Portal are called portal pages.

By using OracleAS Portal, you can provide secure access to existing information, regardless of where the information resides. OracleAS Portal supports personalized views of information resources.

What Is a Portal Page?

- A portal page is an interface that brings information sources together in one place and, thus, serves as a starting point for Web applications.
- The layout of portal pages is defined through regions.
- The regions contain portlets or items.



ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is a Portal Page?

OracleAS Portal displays information in the form of a portal page. A portal page enables users to aggregate the information that they need to perform their core job functions.

Portal pages are created and maintained by users or by OracleAS Portal page developers on behalf of users. Users can create, edit, and customize a portal page by using simple Web browser-based wizards. Although many pages are created by page creators, the wizards are simple enough to allow end users to create their own pages. Content can be modified by any user who has been given the privilege to manage the content.

The key elements that define an OracleAS Portal page include page layout, page content that is presented through portlets and items, page style, and page security. The page layout is defined through regions that hold either portlets or items. Page style defines the look-and-feel of the page.

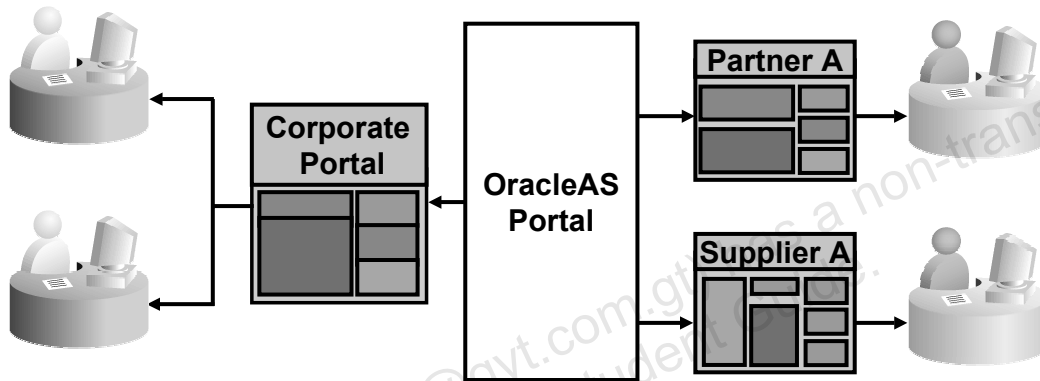
Accessing Portal Pages

Public users:

- No login required
- Common view

Authenticated users:

- Login required
- Personalized views



ORACLE

Copyright © 2005, Oracle. All rights reserved.

Accessing Portal Pages

A portal page can be accessed by public and authenticated users.

Public Users

A public user is any user who has not been authenticated to portal. Public users can view any portal page that has been marked as public by the object owner, but they cannot edit or customize the page.

Authenticated Users

A user who is allowed to log in to OracleAS Portal is considered an authenticated user. After authentication, a user can be authorized to create pages, layouts, and styles, as well as edit and delete the objects that he or she has created. An authenticated user also appears in the list of users to whom privileges may be granted for pages that are created by other users.

What Is a Portlet?

Items in an item region

Example Form on EMP Table

Company Name/Logo

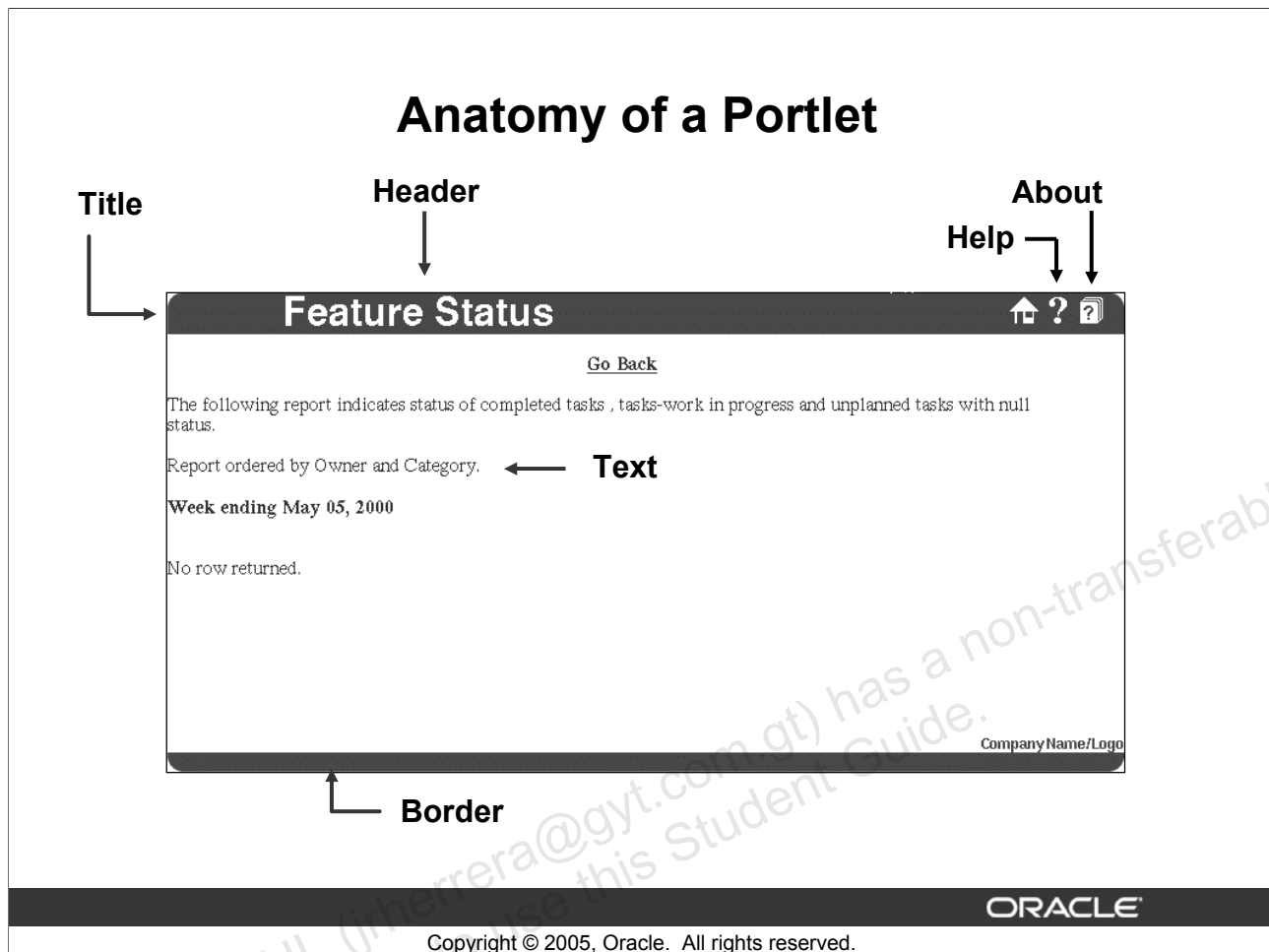
Update	Delete	Reset
Empno	7839	
Name	KING	
Job	PRESIDENT	
Manager		
Hire Date	17-NOV-81	
Salary	5,000.0	
Commision		
Deptno	<input type="radio"/> ACCOUNTING (10) <input type="radio"/> OPERATIONS (40) <input type="radio"/> RESEARCH (20) <input type="radio"/> SALES (30)	
Update	Delete	Reset

What Is a Portlet?

Portlets represent their content in areas within a portlet region in the portal page. You can think of portlets as reusable and pluggable Web components that display portions of Web content. Because of their dynamic nature, portlets are often used to summarize key data, highlight important information, and alert users to new developments.

Portlets are as varied as the interests of the user. For example, a user may need to systematically enter employee salary information into the corporate database. This user is also interested in obtaining stock quotes and accessing e-mails. The content source of the portlet can be a document, an application, or a commonly requested piece of information, such as a report, an employee list, or external information from the Internet. The portlet code runs on the server and delivers real-time HTML to the portal page.

OracleAS Portal offers a large library of prebuilt portlets that you can use in enterprise portals. You can use these portlets for Web publishing and portal administration. In addition, Oracle has created a partner initiative to support a growing community of independent software vendors and Internet content providers who create standard, supported portlets to their applications and services. For more information about those portlets, see the Portal Studio Web site at <http://portalstudio.oracle.com>.



Anatomy of a Portlet

A portlet on a portal page is rendered in an HTML table cell. The portlet can display almost any HTML content. It can contain any type of formatted text, images, or any elements of an HTML form.

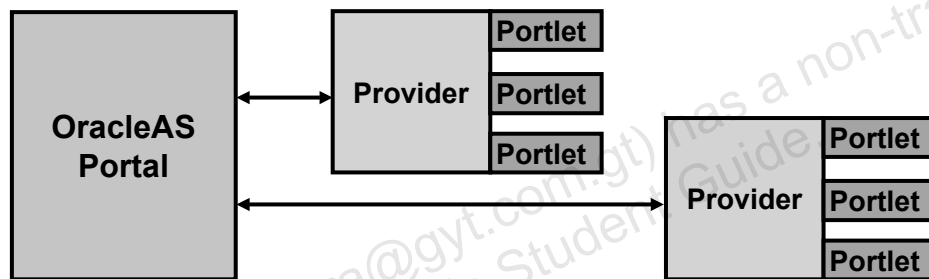
Portlets have a header and a border. The page designer can choose to hide either of them. In the header section, you find the portlet title. Portlets can provide a hyperlink in the portlet title. By clicking the title, the portlet is rendered in full-page mode, where the portlet is displayed in the browser exclusively.

The portlet header also contains links to the Customize, Help, and About windows. By using the Customize window, end users can personalize the portlet to their needs.

The look-and-feel of the portlets on a portal page is controlled by styles, either the region style or the page style.

What Is a Portlet Provider?

- **A portlet provider is an entity that:**
 - Provides a communication link between OracleAS Portal and portlets
 - Is registered in OracleAS Portal
- **OracleAS Portal offers built-in providers and tools to create providers and portlets.**



ORACLE

Copyright © 2005, Oracle. All rights reserved.

What Is a Portlet Provider?

In the OracleAS Portal architecture, a portal never communicates with a portlet directly. Instead, it communicates with the provider, an entity that provides a communication link between the portal and the portlets. When the portal renders a portal page, it calls the provider of each portlet on the page, which, in turn, executes the portlet and returns the results.

Each portlet is contained by one and only one provider. A provider contains one or more portlets that expose an underlying application or information source.

Each provider is registered with the portal either through a browser-based provider administration window or through a registration script. Registration provides the portal with the name and location of the provider code.

OracleAS Portal classifies portlet providers as follows:

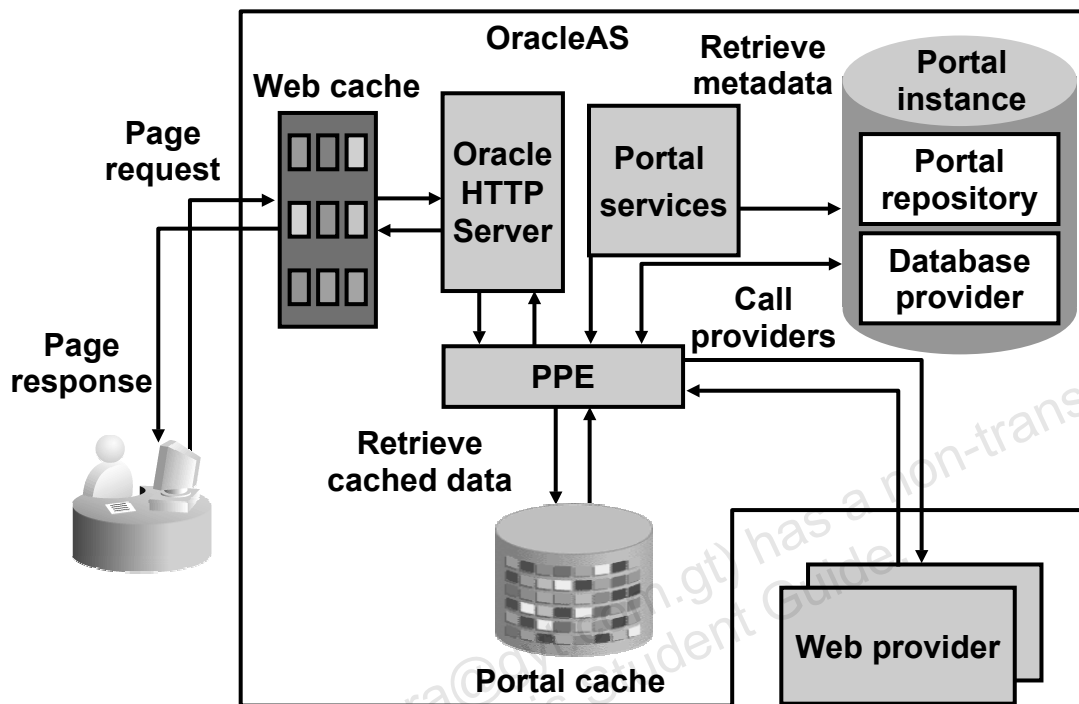
- **Built-in providers:** All major functions within OracleAS Portal are accessed by way of portlets. OracleAS Portal ships with a set of prebuilt portlets that are installed and configured for portal administration, portal development, and general use by portal users.

What Is a Portlet Provider? (continued)

- **Custom providers:** OracleAS Portal provides the facilities to create and maintain portlets that access customer-specific content or applications. The following two approaches for creating custom portlets are supported in OracleAS Portal:
 - **Declarative:** OracleAS Portal's declarative portlet building services are a simple-to-use wizard and dialog box-based mechanism that are used to create Portal DB providers and data-driven portlets, such as forms, reports, or charts.
 - **Programmatic:** To integrate new and legacy applications, and data into OracleAS Portal, customers can develop executable code that implements specific logic or user interface controls. OracleAS Portal supports programmatic portlet creation through a set of open public Java and PL/SQL APIs that are documented in the OracleAS Portal Developer Kit (PDK).

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable license to use this Student Guide.

Requesting a Portal Page



Copyright © 2005, Oracle. All rights reserved.

Requesting a Portal Page

The diagram in the slide illustrates the architecture of the requests that are made to OracleAS Portal.

- The user requests a page from OracleAS Portal with a browser that issues an HTTP request to the specified server and port.
- The request is received by OracleAS Web Cache. If the requested page is available in the Web cache, it is returned to the client. Otherwise, the request is forwarded to Oracle HTTP Server, which translates the request into a call to the Parallel Page Engine (PPE) that is a multithreaded Java servlet.
- The PPE sends a request to the portal services to retrieve the page definition metadata, such as page structure and list of portlets that are published on the page, and user-defined customizations from the portal repository.
- The PPE accepts and inspects the page definition and issues one or more portlet requests, in parallel, to the various portlet providers. The contents of portlets that have been cached earlier and is still valid will be retrieved from the Portal cache.
- Providers return requested portlet content to the PPE. The engine accepts the portlet content, composes a portal page that merges the results, and sends the composed page back to the user.

Built-in Portal Pages

- **Portal Builder page:**
 - **Welcome tab:** Home for public user
 - **Build tab:** Tools and services for authenticated users
 - **Administer tab:** Tools and services for portal administrators
- **Sign-in page: Authentication**
- **Portal Navigator page:**
 - **Page Groups tab**
 - **Providers tab**
 - **Database Objects tab**

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Built-in Portal Pages

OracleAS Portal has a number of built-in portal pages that help in navigation and in performing your tasks:

- **Portal Builder page:** This page is by default a starting page for portal users. It contains three tabs: Welcome, Build, and Administer. The Welcome tab is by default a home page for public users. It contains information about OracleAS Portal and links to information sources where you can learn more about OracleAS Portal. The other two tabs are used by authenticated portal users to perform their tasks.
- **Sign-in page:** You can log in to the portal by providing your credentials on the Sign-in page.
- **Portal Navigator page:** The Portal Navigator is a tool that provides access to OracleAS Portal objects. It enables you to manage your page groups, providers, and database objects.

Oracle Instant Portal

The screenshot shows a web browser window titled "Oracle Instant Portal". It contains two main sections. The left section, "Create an Oracle Instant Portal", instructs the user to enter a name and optionally an XML file. It includes input fields for "Name" (containing "My Instant Portal") and "XML File", with "Create" and "Browse..." buttons respectively. The right section, "Delete or Go To an Oracle Instant Portal", instructs the user to select a portal and click "Go" or "Delete". It includes dropdown menus for "Name" and buttons for "Go" and "Delete". A "Help" link is also present.

The screenshot shows the "My Instant Portal" home page. At the top, it says "Welcome, PORTAL" with links for ">Logout" and ">Change Profile". Below this is a navigation bar with tabs: "Home", "Company", "Sales", "Marketing", "Finance", and "Human Resources". The "Home" tab is selected. The main content area is divided into four sections: "Search My Instant Portal" (with a search icon), "News", "Announcements", "New Content", and "Favorite Content".

Copyright © 2005, Oracle. All rights reserved.

Oracle Instant Portal

Oracle Instant Portal provides an instant out-of-the-box portal application for secure publishing and content sharing. This type of solution is ideal for enterprises with a need to share information on their intranet or internal communications hub. By offering a subset of the Oracle Portal features with enhanced capabilities for ease of use, Oracle Instant Portal enables you to share information quickly and easily without the effort of building a portal.

To create an Oracle Instant Portal, you need to perform the following steps:

1. Log in to OracleAS Portal.
2. Click the Build tab on the Portal Builder page.
3. In the Oracle Instant Portal section, enter a name as My Instant Portal, and click Create. By default, the portal created would be based on a sample company portal structure where individual pages would be created for each department.

Notice that the home page for My Instant Portal is displayed.

Note: During an Instant Portal creation, you can also choose an XML file to define a personalized structure for your portal.

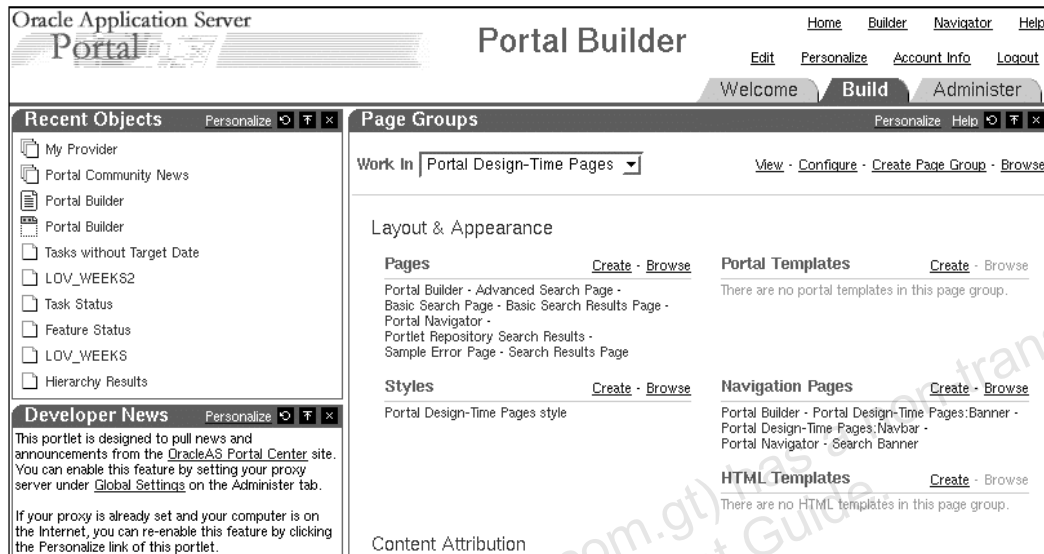
Oracle Instant Portal (continued)

Organizations with limited hardware and software resources can get a portal functional with little development effort. The layered user interface provides context-sensitive toolbars and menus to manipulate the page and its contents. Editing features include point-and-click portal branding and styling, page management, and content management.

For additional information about Oracle Instant Portal, refer to the *Oracle Instant Portal Getting Started 10g Release 2 (10.1.2)*.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

OracleAS Portal Builder Page



OracleAS Portal Builder Page

After you have logged in, your personalized home page is displayed. Depending on your role as a portal user, the home page can be a page of your corporate portal or the Portal Builder page that enables you to access the portal building environment.

The Portal Builder page is divided into three tabbed pages that group the related features together. The tabs that are available to you on the Portal Builder page depend on your portal privileges; therefore, you may not see all of the tabs. Each tabbed page is made up of portlets that provide access to OracleAS Portal tools that enable you to perform development or administrative tasks, such as creating page groups and pages, registering providers, or creating users and groups. The portlets that you can see depend on your privileges. For example, you will be able to see the Remote Providers portlet only if you have privileges to manage providers.

The shortcut bar at the top right of the Portal Builder page provides quick access to important OracleAS Portal pages and services, such as the Portal Navigator page, the Portal Builder page, your home page, and the online Help of OracleAS Portal.

OracleAS Portal Navigator

Oracle Application Server Portal

Portal Navigator

Home Builder Navigator Help

Personalize Account Info Logout

Page Groups Providers Database Objects

Here are the pages and associated actions available to you in this page group.

Create New... [Page](#)

Perform [actions](#) on multiple objects simultaneously.

Path: [Page Groups](#) > [Portal Design-Time Pages](#) > [Pages](#)

Type ▲▼	Name ▲▼	Actions	Creator ▲▼	Last Modified ▲▼ ?
Standard	Advanced Search Page	Create Sub-Page , Edit , Properties , Delete , Move , Copy , Convert to Template	PORTAL	06-OCT-2005 11:54 PM
Standard	Basic Search Page	Create Sub-Page , Edit , Properties , Delete , Move , Copy , Convert to Template	PORTAL	06-OCT-2005 11:54 PM
Standard	Basic Search Results Page	Create Sub-Page , Edit , Properties , Delete , Move , Copy , Convert to Template	PORTAL	06-OCT-2005 11:54 PM
Standard	Portal Navigator	Create Sub-Page , Edit , Properties , Copy , Convert to Template	PORTAL	06-OCT-2005 11:54 PM
Standard	Portlet Repository Search Results	Create Sub-Page , Edit , Properties , Copy , Convert to Template	PORTAL	06-OCT-2005 11:54 PM
Standard	Sample Error Page	Create Sub-Page , Edit , Properties , Delete , Move , Copy , Convert to Template	PORTAL	06-OCT-2005 11:54 PM
Standard	Search Results Page	Create Sub-Page , Edit , Properties , Copy , Convert to Template	PORTAL	06-OCT-2005 11:54 PM

ORACLE

Copyright © 2005, Oracle. All rights reserved.

OracleAS Portal Navigator

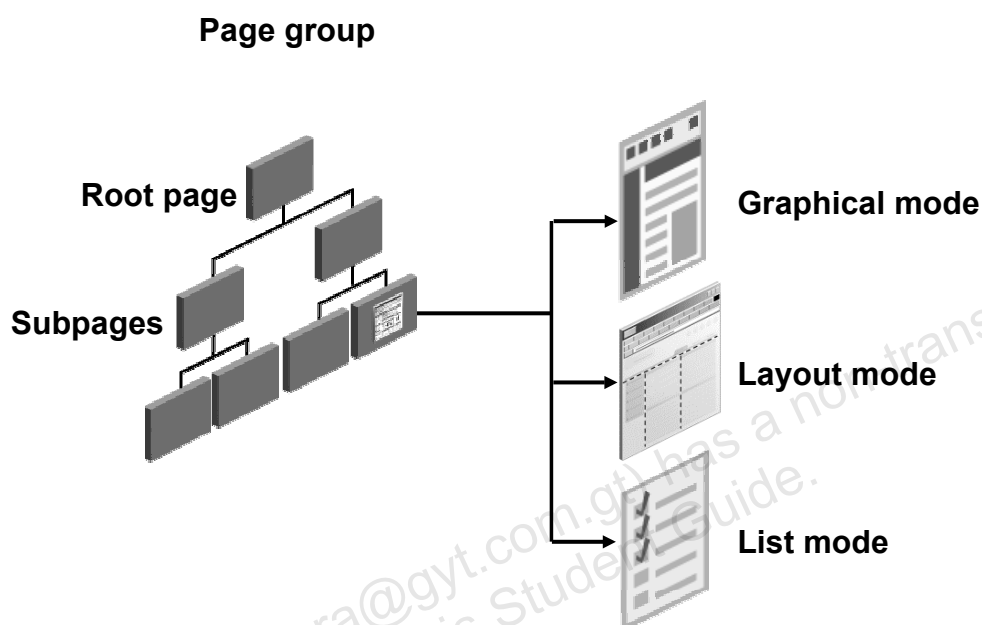
The Navigator is a very powerful tool that provides access to OracleAS Portal objects, which are grouped into three main groups, represented in the Portal Navigator page as tabs. The three tabs are Page Groups, Providers, and Database Objects. You can click the relevant tab, depending on what type of object you are looking for. During an OracleAS Portal session, the Navigator remembers the tab that you used last.

OracleAS Portal objects are organized in hierarchical fashion in each group. The Navigator lists the objects of the same group and level in the form of a table. Each row of the table represents an object type, the object name, a list of actions that you can perform on the object, the object owner name, and the time when the object was last modified. The list of actions varies depending on your privileges. Depending on the object type, you view an object or drill down within the object by clicking its name. Use links in the Path to quickly navigate to the parent objects.

You can use the arrow buttons in the table header to sort the objects in the order that will enable you to locate the object that you are looking for easily. The green arrow indicates the current sort order.

If you know the name, or part of the name of the object that you are looking for, then you can use the Find field to search for that object in the current tab of the Navigator.

Page Group and Portal Page Modes



Copyright © 2005, Oracle. All rights reserved.

Page Group and Portal Page Modes

A page group is an entity that holds related portal objects. A page group comprises a set of pages that are organized in hierarchical fashion. The hierarchy consists of a root page and subpages.

OracleAS Portal provides you with a set of modes that enhances the development of portal pages. You can view and manage your pages in three modes: Graphical, Layout, and List.

- **Graphical mode:** Enables you to view the page content rendered with editing tools, such as the page toolbar and action icons. Using the editing tools, you can edit the page content rendered in place.
- **Layout mode:** Enables you to add and configure the regions on the page and to perform some tasks on multiple objects on the page.
- **List mode:** Enables you to see listings of nonportlet content of the page, such as items, tabs, and subpages. You can also move content between pages, or copy content from or to pages, within the page group.

Getting Help About OracleAS Portal

- **Built-in Help system:**
 - Tutorials
 - Glossary
 - Task-oriented help
 - Tips and troubleshooting
 - Help categories and perspectives
- **Context-sensitive Help**
- **OracleAS Portal Center on OTN**
 - Portal Studio
 - Portlet Catalog
 - OTN discussion forums

ORACLE

Copyright © 2005, Oracle. All rights reserved.

Getting Help About OracleAS Portal

OracleAS Portal provides the following ways of getting help:

- The Help system of the product provides detailed step-by-step instructions and reference information, as well as an introduction to OracleAS Portal and troubleshooting information. You can access the online Help by clicking the Help icon on the shortcut bar at the top of any portal page.
- The context-sensitive Help provides specific information about the fields on the current page. You access the context-sensitive Help by clicking the Help icon on the page.
- There are also several online resources at the OracleAS Portal Center, which is part of the Oracle Technology Network (OTN). Here, you can find information about new features and news, technical help and information, OracleAS Portal Developer Kit, Oracle Partners, discussion forums, and so on.

Index

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

A

Abstract Window Toolkit 9-5
Access Control List 11-24, 12-30, 13-48
Apache Portable Runtime library 6-36
Apache Web Server 2-29, 6-3, 6-5, 6-38
Apache Web server module 6-5, 6-38
Application Server Console 4-28, 4-31
Application Server Control Console 2-18, 3-20, 3-31, 3-34, 4-29,
5-12, 5-13, 5-20, 5-31, 5-34, 5-35, 5-38, 5-40, 5-41, 5-45, 5-46,
8-10, 8-11, 8-13, 8-23, 14-18, 15-8, 15-24, 16-10, 16-11
Application Service Level Management 5-6, 5-7
Automatic Storage Management 5-4

C

Canonical Server name 6-19
certificate revocation list 17-19, 17-23, 17-24, 18-8, 18-33
Common Log Format 6-26, 8-45
Configuration Management metadata 2-22, 3-4
container-managed persistence 9-33, 10-19, 10-21
content delivery networks 8-37

D

Data Encryption Standard 11-9, 17-6
database access descriptor 15-4, D-2, D-4, D-6, D-7, D-9,
D-10, D-13, D-15, D-27
Database Configuration Assistant 3-12, 3-28
Database Control 2-25, 3-18, 3-34, 3-35, 5-4, 19-21
Database Control Console 3-34, 3-35
Delegated Administration Service 1-11, 2-21, 2-23, 3-3, 3-11,
3-20, 11-7, 11-8, 11-20, 11-21, 11-23, 11-24, 12-6, 12-30, 12-47, 12-51,
13-8, 13-14, 13-21, 13-22, 16-1, 16-2, 16-3, 16-4, 16-5, 16-6, 16-7,
16-8, 16-9, 16-10, 16-11, 16-12, 16-13, 16-15, 16-17, 16-18, 16-19, 16-20,
16-21, 16-22, 16-24, 16-25, 16-30, 16-31, 16-33, 16-34, 16-35, 18-37, 18-41

D

Delegated Administration Services 1-11, 2-21, 2-23, 3-3, 3-11,
11-7, 11-20, 11-21, 11-23, 11-24, 12-30, 12-47, 12-51, 13-22, 16-1, 16-2,
16-3, 16-4, 16-5, 16-6, 16-7, 16-8, 16-9, 16-10, 16-11, 16-12, 16-13,
16-15, 16-17, 16-18, 16-19, 16-20, 16-21, 16-22, 16-24, 16-30, 16-31, 16-33,
16-34, 16-35, 18-37
Discovery and Integration 2-35
distinguished name 3-23, 3-24, 12-14, 16-8, 17-10, 18-6, 18-21, 18-22
Distributed Authoring and Versioning 2-30, 6-5, 6-13, 14-2, 14-7
Distributed Configuration Management 2-13, 2-19, 4-15, 5-12, 5-48
domain name system 8-5
Dynamic Monitoring Service 5-12, 5-13, 6-4

E

Edge Side Includes 2-5, 4-25, 8-3, 8-6, 8-7, 8-37
Edge Side Includes for Java 2-5, 8-6
Enterprise Archives 9-9
Enterprise Edition 2-3, 2-29, 9-3, E-3
enterprise information portals 2-37
Enterprise Information System 9-3, 9-4
Enterprise JavaBeans 9-4, 9-8, 9-9, 9-15, 10-5, 11-18
Enterprise Manager 2-12, 2-13, 2-18, 2-25, 3-31, 3-34, 4-24, 5-2,
5-3, 5-4, 5-5, 5-7, 5-16, 5-18, 5-29, 5-30, 5-39, 5-40, 5-41,
5-48, 5-54, 6-33, 8-10, 10-10, 13-6, 14-15, 14-19, 15-24, 17-15
Enterprise Resource Planning 2-14
Extensible Markup Language 9-6

I

Identity Management 1-10, 2-6, 2-13, 2-20, 2-21, 2-22, 2-25,
2-26, 3-3, 3-4, 3-5, 3-11, 3-12, 3-18, 3-20, 3-23, 3-24, 4-11,
4-13, 4-15, 4-18, 4-29, 11-7, 11-8, 11-20, 11-21, 11-22, 12-3, 12-4,
12-5, 12-6, 12-11, 12-12, 12-13, 12-14, 12-15, 12-42, 12-47, 12-48, 12-49,
12-54, 15-4, 16-3, 16-13, 16-14, 16-26, 16-28, 16-29, 16-35, 19-7, 19-29,
19-35, 19-40
Identity Management Realm 3-11, 3-12, 3-23, 3-24, 12-11, 12-12,
12-13, 12-14, 12-15, 12-42, 12-48, 12-49, 16-3, 16-13, 16-14, 16-26, 16-28,
16-29, 16-35

I

Internet Protocol version 6 6-36

J

J2EE Connector Architecture 2-15

Java 2 Platform 2-3, 2-29, 9-3, 11-18

Java Application Archives 9-9

Java Authentication and Authorization Service 2-12, 5-48, 5-49,
9-21, 11-7, 11-18

Java Community Process 9-3

Java Naming and Directory Interface 10-4

Java Runtime Environment 11-7

Java servlets 8-42, 9-4, 9-6

Java Virtual Machine 2-29, 2-34, 9-5, 9-11

javac command line 9-29

JDK javac 9-29

L

Lightweight Directory Access Protocol 2-41, 11-19, 12-5

Log Repository 5-13

M

message authentication codes 17-5

mobileXML 2-40

N

network file system 6-14

O

OmniPortlet Web Provider portlets 14-32

online transaction processing 2-11

OPMN servers 5-42

Oracle ADF Business components 9-4

Oracle Application Development Framework 2-11

O

Oracle Application Server 1-2, 1-3, 1-4, 1-5, 1-6, 1-7,
1-8, 1-9, 1-10, 1-11, 1-12, 2-2, 2-4, 2-5, 2-7, 2-8, 2-9,
2-11, 2-12, 2-13, 2-14, 2-15, 2-16, 2-17, 2-18, 2-19, 2-20, 2-21,
2-22, 2-24, 2-27, 2-28, 2-29, 2-30, 2-34, 2-39, 2-41, 2-42, 3-3,
3-4, 3-6, 3-7, 3-10, 3-11, 3-12, 3-13, 3-14, 3-17, 3-18, 3-20,
3-21, 3-22, 3-24, 3-27, 3-31, 4-2, 4-4, 4-6, 4-8, 4-9, 4-10,
4-11, 4-14, 4-17, 4-20, 4-24, 4-27, 4-28, 4-30, 4-33, 5-1, 5-2,
5-4, 5-5, 5-6, 5-8, 5-10, 5-11, 5-12, 5-13, 5-14, 5-15, 5-16,
5-17, 5-18, 5-20, 5-23, 5-24, 5-25, 5-26, 5-27, 5-28, 5-29, 5-30,
5-33, 5-34, 5-35, 5-36, 5-37, 5-40, 5-41, 5-42, 5-43, 5-44, 5-47,
5-48, 5-49, 5-50, 5-51, 5-53, 6-2, 6-3, 6-4, 6-8, 6-9, 6-10,
6-16, 6-20, 6-21, 6-35, 6-36, 6-42, 8-4, 8-10, 8-15, 8-28, 8-31,
9-8, 9-11, 9-12, 9-23, 9-24, 9-29, 9-31, 10-2, 10-7, 10-8, 10-9,
10-27, 10-30, 11-2, 11-6, 11-7, 11-9, 11-16, 11-17, 11-18, 11-20, 11-21,
11-22, 11-23, 11-24, 11-25, 12-1, 12-6, 12-7, 12-9, 12-10, 12-11, 12-16,
12-17, 12-30, 12-31, 12-34, 12-35, 12-36, 12-37, 12-38, 12-45, 12-46, 13-5,
13-6, 13-8, 13-9, 13-12, 13-14, 13-21, 13-22, 13-40, 13-41, 13-42, 14-3,
14-7, 14-8, 14-10, 14-12, 14-13, 14-17, 14-20, 14-22, 15-3, 15-4, 15-5,
15-14, 15-15, 16-11, 17-2, 17-3, 17-4, 17-7, 17-10, 17-11, 18-10, 18-30,
18-37, 19-1, 19-2, 19-3, 19-5, 19-6, 19-7, 19-8, 19-10, 19-12, 19-13,
19-14, 19-16, 19-17, 19-23, 19-25, 19-26, 19-27, 19-28, 19-29, 19-33, 19-34,
19-35, 19-36, 19-37, 19-43, 19-47, D-4, D-8, D-10, D-12, F-2

Oracle Application Server 10g 2-7, 2-8, 3-13, 3-22, 4-2, 4-6,
4-11, 5-5, 5-8, 5-34, 6-4, 8-10, 8-31, 9-8, 9-11, 11-7, 11-21,
12-11, 13-22, 13-40, 19-3

Oracle Application Server 10g Containers for J2EE 4-11

Oracle Application Server 10g Web Cache 4-11

Oracle Application Server Adapter 2-14, 2-15

Oracle Application Server Administration Service 6-35

Oracle Application Server Backup and Recovery Tool 19-2, 19-47

Oracle Application Server business intelligence features 2-5

Oracle Application Server Certificate Authority 1-3, 1-11, 2-12,
2-21, 3-3, 3-11, 3-12, 3-18, 3-20, 3-24, 17-2, 17-3, 17-4

Oracle Application Server Cluster 2-18, 5-10, 5-17, 5-18, 6-8

O

Oracle Application Server configuration repository 5-10
Oracle Application Server Control Console 5-20
Oracle Application Server Farm 5-34, 5-35
Oracle Application Server Forms Services 2-12
Oracle Application Server Home 5-41, 19-33
Oracle Application Server Infrastructure 3-17, 4-27
Oracle Application Server Instance 1-7, 1-12, 2-5, 2-18, 2-20,
3-18, 3-27, 3-31, 5-2, 5-5, 5-10, 5-14, 5-15, 5-16, 5-17, 5-23,
5-24, 5-25, 5-27, 5-28, 5-29, 5-33, 5-34, 5-40, 5-48, 5-49, 5-51,
5-53, 6-16, 12-46, 16-11
Oracle Application Server Integration B2B 2-14, 2-15
Oracle Application Server Integration InterConnect 2-14, 2-15
Oracle Application Server Log Loader component 5-13
Oracle Application Server Portal 4-27, 4-28, 4-30, 13-41, 14-3,
14-10, 14-17, 14-22, D-10, D-12
Oracle Application Server Reports Services 2-11, 2-12
Oracle BPEL Designer 2-14
Oracle Business Intelligence Discoverer 2-11, 2-29
Oracle clustering 6-3
Oracle Collaboration Suite 2-6, 5-5, 12-6
Oracle Delegated Administration Services 1-11, 2-21, 2-23, 3-3,
3-11, 11-20, 11-21, 11-23, 11-24, 13-22, 16-1, 16-2, 16-3, 16-4, 16-5,
16-6, 16-7, 16-8, 16-10, 16-11, 16-12, 16-13, 16-15, 16-17, 16-18, 16-19,
16-20, 16-21, 16-22, 16-24, 16-30, 16-31, 16-33, 16-34, 16-35
Oracle Directory Integration and Provisioning 2-21, 3-3, 3-11
Oracle Directory Integration Platform 12-52
Oracle Directory Provisioning Integration service 12-52, 12-53
Oracle E-Business Suite 2-6, 2-29, 12-6
Oracle Enterprise Manager 10g 2-13, 2-18, 3-31, 3-34, 5-2, 5-3,
5-4, 5-7, 5-18, 5-40, 5-41, 5-54, 13-6, 14-19
Oracle Enterprise Manager 10g Application Server Control 2-13,
2-18, 3-31, 5-2, 5-18, 13-6
Oracle Enterprise Manager 10g Database Control 3-34, 5-4
Oracle Enterprise Manager 10g Grid Control 2-13, 2-18, 5-7

O

Oracle HTTP Server 1-7, 1-11, 2-4, 2-7, 2-11, 2-16, 2-17, 2-21,
2-29, 2-30, 2-37, 3-3, 3-20, 3-29, 3-31, 4-9, 4-11, 4-12, 4-24,
5-4, 5-14, 5-16, 5-23, 5-26, 5-48, 5-49, 6-1, 6-2, 6-3, 6-4,
6-5, 6-6, 6-7, 6-8, 6-9, 6-10, 6-11, 6-12, 6-15, 6-16, 6-17,
6-18, 6-19, 6-20, 6-21, 6-22, 6-23, 6-24, 6-28, 6-31, 6-32, 6-33,
6-34, 6-35, 6-36, 6-37, 6-39, 6-41, 6-42, 7-3, 7-4, 7-7, 7-12,
7-13, 7-16, 7-17, 7-18, 7-22, 7-23, 7-24, 7-25, 7-26, 7-27, 7-28,
7-29, 8-20, 9-20, 10-26, 11-7, 11-16, 11-17, 11-22, 13-4, 14-7, 14-8,
14-19, 14-24, 14-25, 14-32, 15-4, 15-5, 15-8, 15-9, 15-12, 15-15, 16-7,
16-11, 16-12, 17-11, 17-15, 17-24, 18-2, 18-29, 18-32, 18-33, 18-37, 18-42,
19-35, C-3, C-10, D-3, D-6, D-7, D-8, D-9, D-11, D-12, D-13,
D-14, D-15, D-22, D-26, F-9

Oracle HTTP Server modules 6-2, 6-42

Oracle HTTP Server processing model 6-2, 6-42

Oracle Internet Directory Integration Service 12-5

Oracle Internet Directory Provisioning Integration Service 12-5

Oracle Internet Directory Self-Service Console 12-44, 12-46, 16-2,
16-4, 16-5, 16-6, 16-27, 16-28, 16-30, 16-32, 16-35

Oracle Management Agent 5-8, 5-12, 5-13

Oracle Management Repository 5-8

Oracle Management Service 5-8

Oracle Management Watchdog Process 5-12, 5-13

Oracle Net Configuration Assistant 3-28

Oracle Notification Server 5-42

Oracle Portal Configuration Assistant 18-37

Oracle Process Manager and Notification Server 2-13, 4-28, 4-31,
5-2, 5-12, 5-13, 5-53

Oracle Remote Method Invocation 9-8

Oracle Sensor Edge Server 2-11

Oracle Single Sign-On 2-8

Oracle Thin driver 10-17

Oracle Ultra Search 2-29, 13-8

Oracle wallet 1-11, 11-11, 11-22, 17-3, 17-4, 18-2, 18-10

Oracle Web Cache 8-39, 14-31, 14-33

O

Oracle9i Database Studio 3-18

OracleAS Certificate Authority 11-8, 11-22, 12-6, 17-1, 17-4, 17-7,
17-8, 17-9, 17-12, 17-13, 17-14, 17-15, 17-16, 17-17, 17-19, 17-20, 17-21,
17-23, 17-26, 18-2, 18-3, 18-4

OracleAS Containers for J2EE 2-11, 2-21, 2-29, 2-30, 2-34, 6-12

OracleAS Farm 2-18, 4-14, 5-16, 5-17, 5-20

OracleAS File-based Farm 4-10, 4-13, 4-14, 4-16, 5-17

OracleAS Infrastructure 1-2, 1-3, 1-4, 1-6, 1-7, 2-13, 2-16,
2-17, 2-20, 2-25, 2-26, 2-27, 2-28, 2-29, 3-1, 3-2, 3-4, 3-5,
3-6, 3-7, 3-8, 3-10, 3-11, 3-12, 3-15, 3-16, 3-17, 3-20, 3-29,
3-30, 3-31, 3-32, 3-33, 3-36, 3-37, 3-38, 4-3, 4-11, 4-13, 4-18,
4-23, 5-17, 5-45, 5-46, 5-47, 8-37, 12-19, 12-21, 12-27, 13-9, 14-29,
15-4, 15-10, 15-24, 16-10, 17-10, 17-16, 18-41, 19-2, 19-5, 19-16, 19-17,
19-27, 19-30, 19-31, 19-38, 19-39, 19-40, 19-41, 19-42, 19-47

OracleAS Instance 9-16, 9-17, 9-18, 9-19, 10-13, 10-29, 13-4, 13-13

OracleAS Integration Adapters 2-12

OracleAS Integration B2B 2-12

OracleAS MapViewer 2-11

OracleAS Metadata Repository 2-16, 2-21, 2-22, 2-25, 2-26, 3-3,
3-4, 3-5, 3-11, 3-18, 3-19, 3-20, 3-34, 3-35, 4-11, 4-14, 4-15,
4-19, 4-29, 4-30, 5-42, 5-45, 5-47, 5-49, 11-24, 13-5, 14-9, 14-10,
14-12, 14-15, 14-16, 14-20, 19-13, 19-42

OracleAS Middle Tier 1-2, 1-3, 1-4, 1-6, 2-16, 3-4, 4-1,
4-2, 4-4, 4-5, 4-18, 4-32, 4-33, 5-45, 5-46, 5-47, 13-11, 13-42,
14-9, 14-12, 15-26

OracleAS PKI 17-2, 17-3, 17-26

OracleAS Portal Configuration Assistant 12-52

OracleAS Portal middle tier 4-27, 14-24, 14-25, 18-39

OracleAS Single Sign-On 1-3, 1-10, 2-16, 2-21, 2-23, 2-24, 2-26,
2-30, 3-3, 3-11, 3-18, 3-20, 4-11, 4-13, 4-25, 5-4, 11-8, 11-22,
11-23, 11-24, 12-6, 12-9, 13-10, 14-19, 14-20, 15-2, 15-3, 15-4, 15-5,
15-6, 15-7, 15-8, 15-9, 15-10, 15-11, 15-12, 15-13, 15-14, 15-17, 15-19,
15-20, 15-21, 15-22, 15-23, 15-24, 15-25, 15-28, 17-7, 17-9, 17-12, 17-15,
17-23, 18-4, 18-5, 18-40

O

OracleAS Single Sign-On server 1-10, 15-2, 15-3, 15-4, 15-5, 15-6,
15-8, 15-10, 15-12, 15-13, 15-14, 15-17, 15-19, 15-20, 15-21, 15-22, 15-23,
15-24, 15-25, 15-28, 17-7, 17-9, 17-12, 17-15

OracleAS SSO server 12-9

OracleAS TopLink 2-11

OracleAS Web Cache 1-7, 1-11, 2-5, 2-7, 2-8, 2-11, 2-13, 2-16,
2-17, 2-29, 2-31, 2-32, 2-33, 2-37, 4-9, 4-12, 4-25, 5-4, 8-1,
8-2, 8-3, 8-4, 8-5, 8-6, 8-7, 8-8, 8-9, 8-10, 8-11, 8-12,
8-13, 8-14, 8-15, 8-16, 8-17, 8-18, 8-19, 8-20, 8-21, 8-22, 8-24,
8-26, 8-27, 8-28, 8-29, 8-30, 8-31, 8-33, 8-34, 8-35, 8-36, 8-37,
8-38, 8-39, 8-40, 8-41, 8-42, 8-43, 8-45, 8-46, 8-47, 8-48, 8-49,
8-50, 8-51, 8-52, 8-53, 11-7, 11-16, 11-23, 13-8, 14-12, 14-15, 14-18,
14-19, 14-20, 14-23, 14-24, 14-25, 16-33, 17-11, 18-36, 18-37, 18-38, 18-41,
F-9

OracleAS Web Cache Manager 8-10, 8-12, 8-41

OracleAS Web Services 2-11, 2-29, 2-35

OracleAS Wireless 2-5, 2-11, 2-13, 2-16, 2-29, 2-38, 2-39, 2-40,
4-9, 5-6, 8-28

P

PL/SQL Gateway 6-5

Portal middle tier 4-27, 14-24, 14-25, 18-39

Public Key Certificate 11-13, 11-15

public key infrastructure 11-6, 17-3

R

Real Application Clusters 2-7, 12-7

Remote Procedure Call 2-35

RMI/Internet Inter-ORB Protocol 9-8

S

secure sockets layer 2-5, 2-23, 2-24, 11-6, 11-9, 11-16, 11-22,
12-7, 12-28, 17-3, 17-6, 17-7, 18-4, 18-29, 18-36

Service-Oriented Architecture 2-14, 2-29

Short Message Service 2-38

S

single sign-on 1-3, 1-10, 2-5, 2-8, 2-12, 2-16, 2-21, 2-23,
2-24, 2-26, 2-28, 2-30, 3-3, 3-11, 3-18, 3-20, 3-36, 4-11, 4-13,
4-25, 5-4, 5-45, 6-3, 6-5, 11-7, 11-8, 11-17, 11-18, 11-20, 11-22,
11-23, 11-24, 12-6, 12-9, 13-3, 13-10, 13-14, 13-15, 14-19, 14-20, 14-21,
15-2, 15-3, 15-4, 15-5, 15-6, 15-7, 15-8, 15-9, 15-10, 15-11, 15-12,
15-13, 15-14, 15-15, 15-16, 15-17, 15-19, 15-20, 15-21, 15-22, 15-23, 15-24,
15-25, 15-28, 17-3, 17-7, 17-9, 17-12, 17-15, 17-23, 18-4, 18-5, 18-40,
D-10, D-12, D-21

T

Transport Layer Security 11-9, 11-16, 11-22

U

Universal Description 2-35

W

Wallet Managers 11-15

Web Archive 9-9

Web Cache 1-7, 1-11, 2-5, 2-7, 2-8, 2-11, 2-12, 2-13, 2-16,
2-17, 2-26, 2-28, 2-29, 2-31, 2-32, 2-33, 2-37, 3-4, 4-9, 4-10,
4-11, 4-12, 4-13, 4-14, 4-15, 4-16, 4-20, 4-22, 4-25, 5-4, 6-3,
6-19, 8-1, 8-2, 8-3, 8-4, 8-5, 8-6, 8-7, 8-8, 8-9, 8-10,
8-11, 8-12, 8-13, 8-14, 8-15, 8-16, 8-17, 8-18, 8-19, 8-20, 8-21,
8-22, 8-23, 8-24, 8-26, 8-27, 8-28, 8-29, 8-30, 8-31, 8-32, 8-33,
8-34, 8-35, 8-36, 8-37, 8-38, 8-39, 8-40, 8-41, 8-42, 8-43, 8-45,
8-46, 8-47, 8-48, 8-49, 8-50, 8-51, 8-52, 8-53, 11-7, 11-16, 11-21,
11-23, 13-4, 13-5, 13-8, 13-16, 13-17, 13-27, 13-32, 14-12, 14-15, 14-18,
14-19, 14-20, 14-23, 14-24, 14-25, 14-31, 14-33, 16-33, 17-11, 18-2, 18-36,
18-37, 18-38, 18-41, 18-42, 19-7, 19-26, 19-35, D-21, F-9

Web Cache Listener 4-22

Web Cache Log Format 8-45

Web Clipping provider 14-28, 14-29, 14-30, 14-31

Web Clipping Repository 14-28, 14-29, 14-33

Web Services Description Language 2-35, 13-39

Web Services for Remote Portlets 4-26, 13-38

Web-based Distributed Authoring and Versioning 14-2, 14-7

WebDAV 1-7, 6-5, 14-2, 14-7, 14-8, 14-34

W

Wireless Markup Language 9-6

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.

HERRERA RAUL (jrherrera@gyt.com.gt) has a non-transferable
license to use this Student Guide.