

Oracle® Applications

Concepts

Release 11*i* (11.5.10.2)

Part No. B19295-03

May 2006

Oracle Applications Concepts, Release 11i (11.5.10.2)

Part No. B19295-03

Copyright © 2000, 2006, Oracle. All rights reserved.

Primary Author: Robert Farrington

Contributing Author: Steven Chan, Ivo Dujmovic, Henriette Fux, Deyu Hu, Mike Mast, Emily Nordhagen, Sarita Nori, Lyn Pratt

Contributor: Robert Azzopardi, Nadia Bendjedou, Michael Bernstein, Eric Bing, Anne Carlson, Subash Chadalavada, Christina Clark, Carole Eubanks, Eric Fichet, Michael Fiore, Ric Ginsberg, Cliff Godwin, Brad Goodwin, William Greene, John Heimann, Jeff Lunn, Steve Mayze, K.R. Narayanan, Kent Noble, Mladena Novakovic, Lisa Parekh, Andrew Rist, Joan Ryan, Greg Seiden, Yun Shaw, Leslie Studdard, Keith M. Swartz, Mildred Wang

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

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

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software–Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

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

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

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

Contents

Send Us Your Comments

Preface

1 Applications Architecture

Introduction	1-1
The Desktop Tier	1-2
Forms Client Applet	1-4
Oracle JInitiator	1-4
The Application Tier	1-4
Web Server	1-5
Forms Server	1-7
Reports Server	1-10
Discoverer Server	1-11
Concurrent Processing Server	1-12
Admin Server	1-18
Daily Business Intelligence (DBI)	1-18
The Database Tier	1-19
The Oracle Applications Technology Layer	1-20
Oracle Applications DBA (AD)	1-20
Oracle Common Modules (AK)	1-20
Oracle Applications Utilities (AU)	1-21
Oracle Application Object Library (FND)	1-21
Oracle Workflow (OWF)	1-23
Oracle Alert (ALR)	1-24
Oracle XML Publisher (XDO)	1-24

2 Applications File System

Introduction	2-1
The DATA Directory	2-2
DB and ORA Directories	2-2
The COMN Directory	2-3
The APPL Directory	2-4
Product Directories	2-5
Language Files	2-9

Core Technology Directories	2-9
Distributing the APPL_TOP Across Disks	2-10
Environment Settings	2-10
3 Applications Database Organization	
Introduction	3-1
Oracle User IDs	3-1
APPS Schema	3-1
Custom Schema Access	3-3
Data Access	3-3
Space Management	3-3
4 Database Features and Release 11i	
Introduction	4-1
Performance Features	4-1
Business Intelligence Features	4-3
Scalability	4-3
5 AutoConfig	
Introduction	5-1
The Applications Context	5-1
AutoConfig Scripts and Directories	5-2
AutoConfig Operation	5-3
Management Tasks	5-7
6 Cloning	
Introduction	6-1
Cloning Tools	6-1
Cloning Across Platforms	6-1
Cloning Options	6-2
Rapid Clone Operation	6-2
Rapid Clone and Oracle Universal Installer	6-3
New Cloning Features in Release 11.5.10	6-3
Summary	6-4
7 Oracle Applications Manager	
Introduction	7-1
Oracle Applications Manager Features	7-1
Additional Features	7-3
Generic Service Management	7-4
8 Authentication and Integration	
Introduction	8-1

Oracle9i Application Server.	8-1
Oracle Application Server 10g.	8-2
Oracle Portal	8-3
Oracle Discoverer	8-3
Enterprise-Wide Single Sign-On.	8-4
Application Server Integration Options	8-5
Basic Single Sign-On Deployment Scenario	8-6
Advanced Single Sign-On Deployment Scenarios	8-9
Advanced Single Sign-On Options	8-12
9 High Availability	
Introduction	9-1
Patching Hints and Tips	9-1
Maintenance Mode	9-2
Shared APPL_TOP and Shared Application Tier File System	9-2
Staged APPL_TOP.	9-5
Distributed AD	9-7
Nologging Operations	9-8
Disaster Recovery	9-9
10 Load Balancing	
Introduction	10-1
Categories of Load Balancer.	10-2
Load Balancing Options	10-2
11 Security	
Introduction	11-1
Authentication	11-1
Authorization	11-1
Audit Trail	11-2
Network Security	11-2
Oracle User Management.	11-2
Security Strategies	11-3
12 Network Topologies	
Introduction	12-1
Strategies	12-1
Latency	12-1
Satellite Links.	12-1
Wireless LANs	12-2
13 Internationalization Support	
Introduction	13-1

Languages and Character Sets on the Database Tier	13-1
Languages and Character Sets on the Application Tier	13-2
Character Sets on the Desktop Tier	13-2
External Documents	13-3
Globalization and Country-Specific Functionalities	13-3
Dates and Numbers	13-3
Multiple Reporting Currencies	13-4
NLS-Independent Application Servers	13-4
NLS Settings	13-4

14 Multi-Org Architecture

Introduction	14-1
Overview	14-1
Multi-Org Partitioned Objects	14-2
Converting to Multi-Org	14-2

15 Multiple Reporting Currencies

Introduction	15-1
Functional Currencies	15-1
MRC Architecture	15-2

Glossary

Index

Send Us Your Comments

Oracle Applications Concepts, Release 11i (11.5.10.2)

Part No. B19295-03

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on Oracle Metalink and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and enquire about our Oracle University offerings. A list of Oracle offices is available on our website at www.oracle.com.

Preface

Intended Audience

Welcome to Release 11*i* (11.5.10.2) of the *Oracle Applications Concepts*.

This book is intended for all those planning to deploy Oracle E-Business Suite Release 11*i*, or contemplating significant changes to a configuration. It focuses on high-level, strategic topics, to enable system administrators and others to make informed decisions about the various installation and configuration choices open to them.

The book does not attempt to replace or supplant the installation, maintenance, and upgrade manuals. As such, it does not describe procedures in depth, but gives a broad outline of the actions needed to achieve a particular goal. The book is arranged such that it starts with topics that will apply to all installations, then examines areas which, while optional, will be relevant to most sites. Typically, these optional subjects are interrelated, so that a site will most likely be implementing more than one. Finally, several specialized areas are discussed; these are mainly topics that are mentioned primarily for reference.

Note: This book typically uses UNIX nomenclature in specifying files and directories. Windows users should substitute the appropriate Windows terms where applicable. For example, a UNIX .env (environment) file will be a .cmd (command) file on Windows.

See Related Documents on page x for more Oracle Applications product information.

TTY Access to Oracle Support Services

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

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Structure

- 1 Applications Architecture**
- 2 Applications File System**
- 3 Applications Database Organization**
- 4 Database Features and Release 11i**
- 5 AutoConfig**
- 6 Cloning**
- 7 Oracle Applications Manager**
- 8 Authentication and Integration**
- 9 High Availability**
- 10 Load Balancing**
- 11 Security**
- 12 Network Topologies**
- 13 Internationalization Support**
- 14 Multi-Org Architecture**
- 15 Multiple Reporting Currencies**
- Glossary**

Related Documents

This book was up to date at the time it was initially published. It is included on the Oracle Applications Document Library CD, which is supplied in the Release 11i Update CD Pack. You can download soft-copy documentation from <http://otn.oracle.com/documentation>, or you can purchase hard-copy documentation from the Oracle Store at <http://oraclestore.oracle.com>. The Oracle E-Business Suite Documentation Library CD-ROM Release 11.5.10+ contains the latest information, including any documents that have changed significantly between releases. If substantial changes to this book are necessary, a revised version will be made available on the "virtual" documentation CD on OracleMetaLink.

You should have the following key books on hand as you deploy and maintain Oracle Applications. Depending on the requirements of your particular installation, you may also need additional manuals or guides.

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. You should use this guide in conjunction with individual product user's guides and implementation guides.

Upgrading Oracle Applications

This guide describes the upgrade process, and lists database and product-specific upgrade tasks.

Oracle Applications Maintenance Procedures and Oracle Applications Maintenance Utilities

These two guides provide information on maintaining the Oracle Applications file system and database. They contain how-to steps, screenshots, and other guidelines for running the Applications DBA (AD) utilities.

Oracle Applications Product Update Notes

Use this guide as a reference for upgrading an installation of Oracle Applications. It includes new features, enhancements, and changes made to database objects, profile options, and seed data.

Oracle Applications System Administrator's Guide

This guide provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and online help, and manage concurrent processing.

Oracle Workflow Administrator's Guide

This guide explains how to carry out the setup steps necessary for Oracle Applications products that include workflow-enabled processes, as well as how to monitor the progress of runtime workflow processes.

Oracle Workflow Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle Applications embedded workflow processes. It also describes how to define and customize business events and event subscriptions.

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI) available with Oracle Applications. It also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes. You can access this user's guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development teams. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface, as described in the Oracle Applications User Interface Standards for Forms-Based Products. It also provides information to help you build your custom Oracle Forms Developer 6i forms so that they integrate with Oracle Applications.

Oracle Applications User Interface Standards for Forms-Based Products

This guide contains the user interface (UI) standards followed by Oracle Applications development. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built using Oracle Forms.

Multiple Reporting Currencies in Oracle Applications

If you use the Multiple Reporting Currencies feature to record transactions in more than one currency, refer to this manual before implementing your Oracle Applications products. This manual details additional steps required to implement Oracle Applications products with this feature.

Multiple Organizations in Oracle Applications

This guide describes how to set up and use the Oracle Applications Multiple Organization support feature, so you can define and support several different organization structures from a single installation of your Oracle products.

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRMs are available on Oracle MetaLink.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle **STRONGLY RECOMMENDS** that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

Applications Architecture

Introduction

This chapter describes the Oracle Applications architecture and some of the features that this architecture supports. The following topics are included:

- Overview
- The Desktop Tier
- The Application Tier
- The Database Tier
- The Oracle Applications Technology Layer

The *Oracle Applications Architecture* is a framework for multi-tiered, distributed computing that supports Oracle Applications products. In this model, various *servers* are distributed among multiple levels, or *tiers*.

A server is a process or group of processes that runs on a single machine and provides a particular functionality, often referred to as a *service*. For example, the *HTTP server* is a process that listens for and processes HTTP requests, and the *Forms server* is a process that listens for and processes requests for activities related to Oracle Forms.

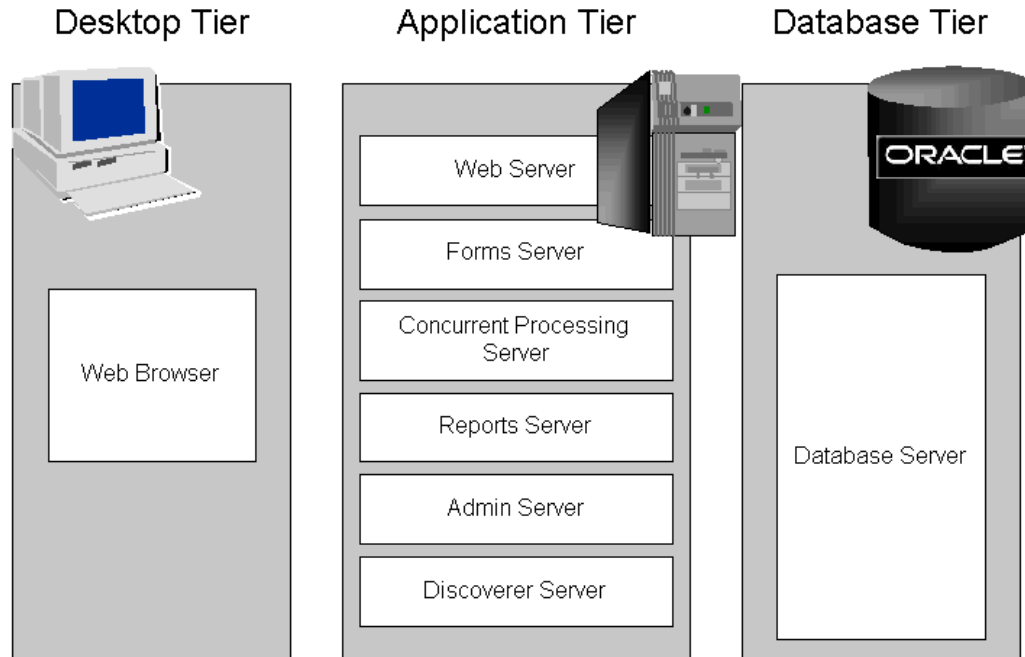
A tier is a logical grouping of services, potentially spread across more than one physical machine. The three-tier architecture that comprises an Oracle E-Business Suite installation is made up of the *database tier*, which supports and manages the Oracle database; the *application tier*, which supports and manages the various Applications components, and is sometimes known as the middle tier; and the *desktop tier*, which provides the user interface via an add-on component to a standard web browser.

A machine may be referred to as a *node*, particularly in the context of a group of computers that work closely together in a *cluster*. Each tier may consist of one or more nodes, and each node can potentially accommodate more than one tier. For example, the database can reside on the same node as one or more application tier components. This may be done for simplicity of administration, or (as in the case of *concurrent processing*) to maximize performance by reducing network traffic.

Centralizing the Oracle Applications software on the application tier eliminates the need to install and maintain application software on each desktop client PC, and also enables Oracle Applications to scale well with an increasing load. Extending this concept further, one of the key benefits of using the *Shared APPL_TOP* model (see Chapter 9) is the need to maintain only a single copy of the relevant Applications code, instead of a copy for every application tier machine.

On the database tier, there is increasing use of *Real Application Clusters (RAC)*, where multiple nodes support a single database instance to give greater availability and scalability.

Figure 1-1 Oracle Applications Architecture

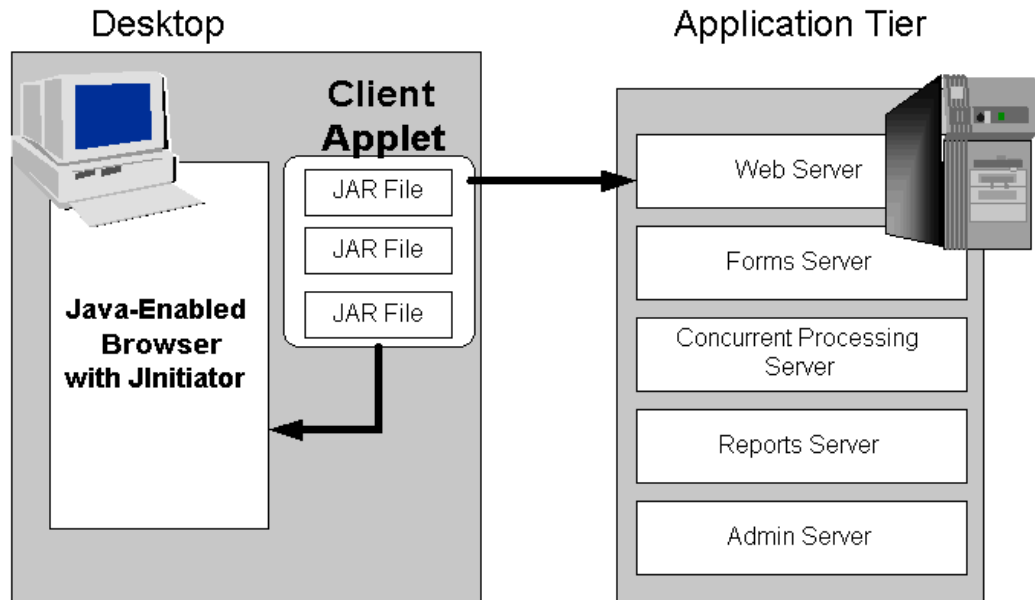


The connection between the application tier and the desktop tier can operate successfully over a Wide Area Network (WAN). This is because the desktop and application tiers exchange a minimum amount of information, for example field value comparison differences. In a global operation with users at diverse locations, requiring less network traffic reduces telecommunications costs and improves response times for users.

The Desktop Tier

The client interface is provided through HTML for the newer HTML-based applications, and via a Java applet in a Web browser for the traditional Forms-based interface.

Figure 1-2 Forms-based Desktop Tier Architecture



In Oracle Applications Release 11*i*, each user logs in to Oracle Applications through the E-Business Suite Home Page on a desktop client web browser. The E-Business Suite Home Page provides a single point of access to HTML-based applications, Forms-based applications, and Business Intelligence applications.

Once logged in via the E-Business Suite Home Page, you need not sign on again to access other parts of the system. Oracle Applications does not prompt again for user name and password, even when you navigate to other tools and products. Oracle Applications also retains preferences as you navigate through the system. For example, if you registered in the E-Business Suite Home Page that German is your preferred language, this preference carries over whether you access Forms-based or HTML-based applications.

Figure 1-3 Example Oracle E-Business Suite Home Page



Forms Client Applet

The *Forms client applet* is a general-purpose presentation applet that supports all Oracle Applications Forms-based products, including those with customizations and extensions. The Forms client applet is packaged as a collection of *Java Archive* (JAR) files. The JAR files contain all Java classes required to run the presentation layer of Oracle Applications forms.

Oracle JInitiator

The Forms client applet must run within a Java Virtual Machine (JVM) on the desktop client. The *Oracle JInitiator* component allows use of the Oracle JVM on web clients, instead of the browser's own JVM. JInitiator is implemented as a plug-in (Netscape) or ActiveX component (Microsoft Internet Explorer).

In the traditional, Forms-based Oracle Applications environment, JInitiator was run as part of the standard Applications sign-on process. Now, with the move to a mainly HTML-based environment, JInitiator is only invoked when a user chooses to access functions that require it, such as running a form. If JInitiator has not been installed, the Web browser prompts the user to download the required installation executable.

The Forms client applet and commonly used JAR files are downloaded from the Web server at the beginning of the client's first session. Less commonly used JAR files are downloaded as needed. All downloaded JAR files are cached locally on the client, ready for future sessions. This eliminates the network traffic that would be involved in downloading them whenever they were required. In 11.5.10, with JInitiator 1.3.1.18, the cache directory is of the form "<HOMEDRIVE>\Documents and Settings\<Windows User Name>\Oracle Jar Cache"; for example, "C:\Documents and Settings\jsmith\Oracle Jar Cache".

Note: Selecting "Show console" on the "Basic" JInitiator tab will allow you to observe downloading of JAR files, to confirm they are being downloaded when they should be.

All updates to JAR files are installed on the application tier and downloaded to the client automatically, via the caching mechanism outlined above.

The Application Tier

The *application tier* has a dual role: hosting the various servers that process the business logic, and managing communication between the desktop tier and the database tier. This tier is sometimes referred to as the *middle tier*. Oracle9i Application Server (9iAS) provides the technology used on the application tier. Six servers comprise the application tier for Oracle Applications:

- Web server
- Forms server
- Reports server
- Discoverer server (optional)
- Concurrent Processing server
- Admin server

Try to avoid using a mixture of different platforms on your application tier. This makes maintenance easier, since only one set of patches needs to be downloaded.

Load Balancing

The application tier supports load balancing among many of its servers to provide higher availability, fault tolerance, reliability, and optimal scalability. If you have more than one of any of the following types of server, load balancing can be employed:

- Web server
- Forms server
- Reports server
- Discoverer server
- Concurrent Processing server

Typically, load balancing is most commonly used to spread the load across multiple Web servers, where the load may vary significantly. Chapter 10 discusses the various types of load balancing in more detail.

Web Server

The Oracle HTTP server (powered by Apache) acts as the Web server. It processes the requests received over the network from the desktop clients, and includes additional components such as:

- Web Listener
- Java Servlet Engine
- JavaServer Pages (JSP)

The Web listener component of the Oracle HTTP server accepts incoming HTTP requests (for particular URLs) from client browsers.

If possible, the Web server services the requests itself, for example by returning the HTML to construct a simple Web page. If the page referenced by the URL needs advanced processing, the listener passes the request on to the *servlet engine*, which contacts the database server as needed.

HTML-Based Applications and the Oracle Applications Framework

The Oracle HTML-based (formerly known as Self-Service) Applications:

- Do not use Oracle Forms for the interface
- Are designed in pure HTML and JavaScript
- Dynamically generate HTML pages by executing Java code
- Use a metadata dictionary for flexible layout
- Operate by direct connection to the Web server

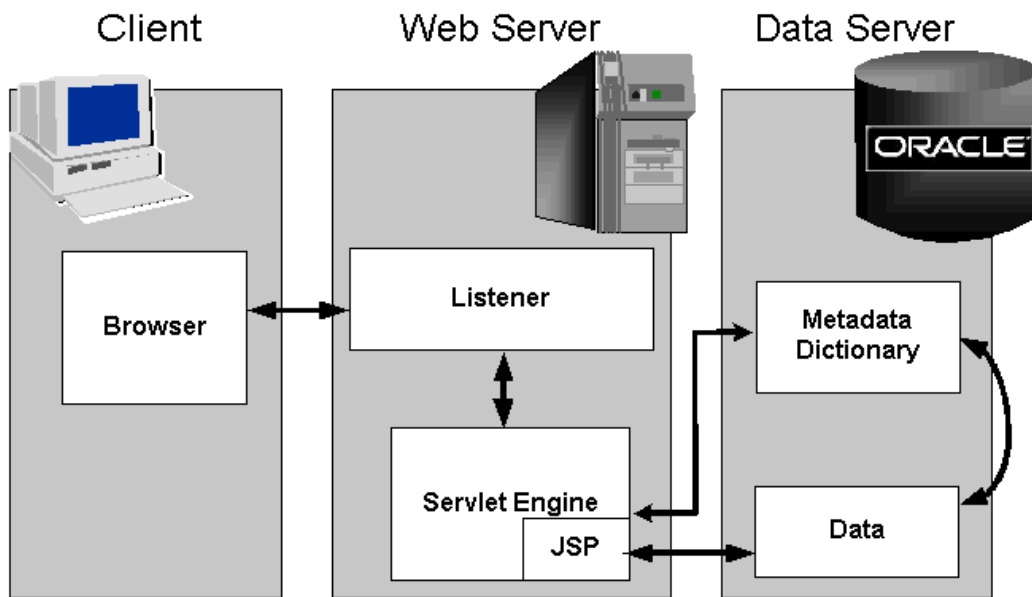
The *Oracle Applications Framework* is the development platform for HTML-based applications. It consists of a Java-based application tier framework and associated services, designed to facilitate the rapid deployment of HTML-based applications.

Notable Oracle Applications Framework components include:

- *Business Components for Java (BC4J)*, included in Oracle JDeveloper, is used to create Java business components for representing business logic. It also provides a mechanism for mapping relational tables to Java objects, and allows the separation of the application business logic from the user interface.
- *AOL/J* supplies the Oracle Applications Framework with underlying security and applications Java services. It provides the Oracle Applications Framework with its connection to the database, and with application-specific functionality such as flexfields.

The Framework-based applications logic is controlled by procedures that execute through the Java servlet engine, which is provided by the Apache JServ module. The servlet engine uses the metadata dictionary in constructing the Framework UI.

Figure 1-4 HTML-Based Applications Architecture

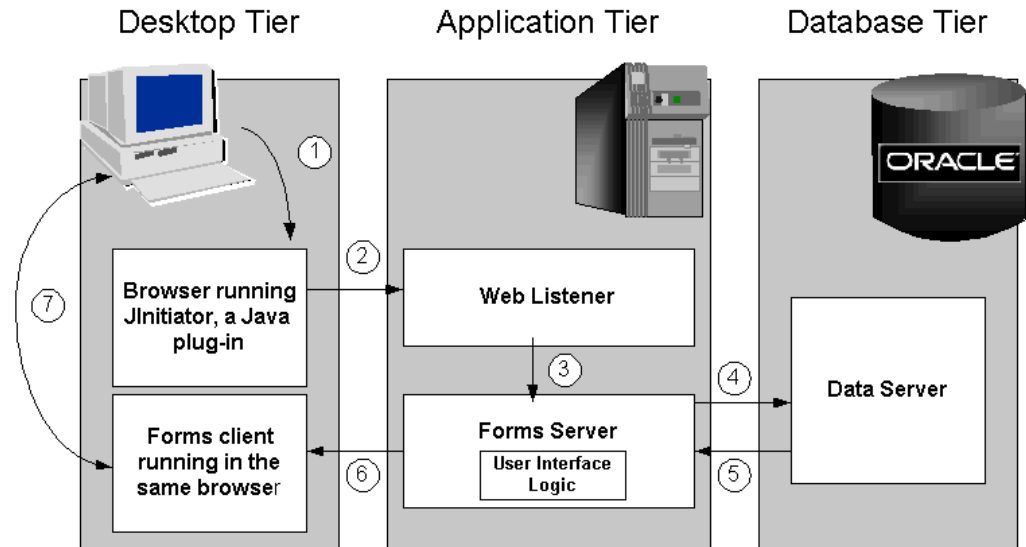


Java Servlet Access with HTML-Based Applications

An HTML-based Applications module uses the following access path:

1. The user clicks the hyperlink of a function from a browser.
2. The browser makes a URL request to the Web listener.
3. The Web listener contacts the Servlet engine (JServ) where it runs a JSP.
4. The JSP obtains the content from the Applications tables and uses information from the metadata dictionary to construct the HTML page.
5. The resulting HTML page is passed back to the browser, via the Web server.

Figure 1-5 Oracle Applications Framework Architecture



Oracle Applications Framework Processing Details

The following is a more detailed explanation of how the JSP obtains the content from the Applications tables and uses information from the metadata dictionary to construct the HTML page.

1. AOL/J validates user access to the page.
2. The page definition (metadata UI definition) is loaded from the metadata repository on the database tier into the application tier.
3. The BC4J objects that contain the application logic and access the database are instantiated.
4. The Java Controller programmatically manipulates the page definition as necessary, based on dynamic UI rules.
5. UIX (HTML UI Generator) interprets the page definition, creates the corresponding HTML in accordance with UI standards, and sends the page to the browser.

Forms Server

The *Forms server* hosts the Oracle Applications forms and associated runtime engine that support the professional interface. It is an Oracle Developer 6i component that mediates the communication between the desktop client and the Oracle database server, displaying client screens and initiating changes in the database according to user actions.

The Forms server caches data and provides it to the client as needed, for example when scrolling through multiple order lines that exceed the limitations of a single screen.

The Forms server communicates with the desktop client using these protocols:

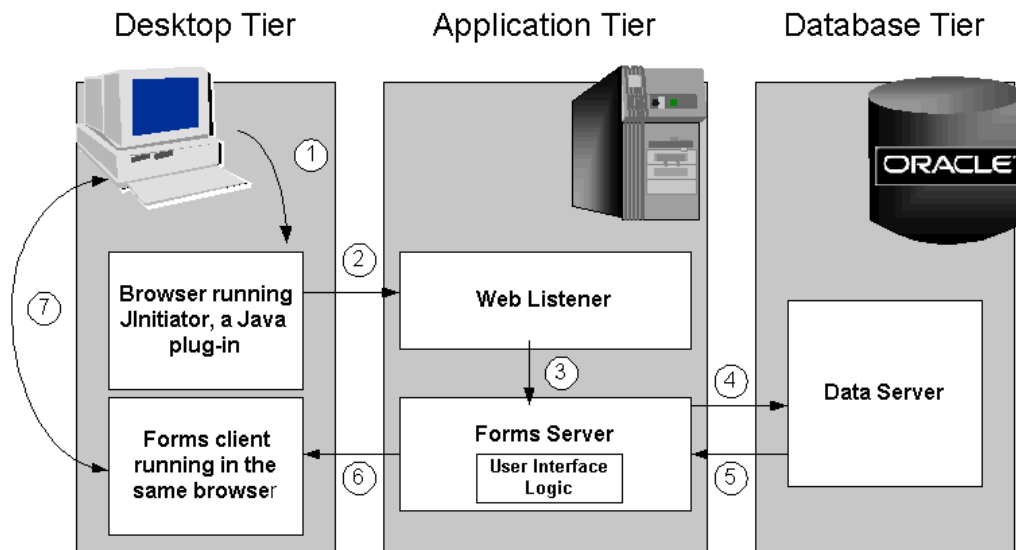
- Standard HTTP network connection
- Secure HTTPS network connection
- TCP/IP connection

Chapter 11 compares HTTP and HTTPS from a security perspective. The Forms server communicates with the Oracle database server using the *Oracle Net* networking infrastructure.

Forms Server Architecture

When a user initiates an action in the Forms client applet, such as entering data into a field or clicking a button, data is passed to the Forms server on the application tier. The user interface logic runs in the Forms server, and determines the appropriate user interface effect based on the user's action. For example, a window may open, or another field value may be populated. If necessary, the database tier is contacted for any data not already cached on the application tier, or for data-intensive processing.

Figure 1-6 Forms Server Architecture



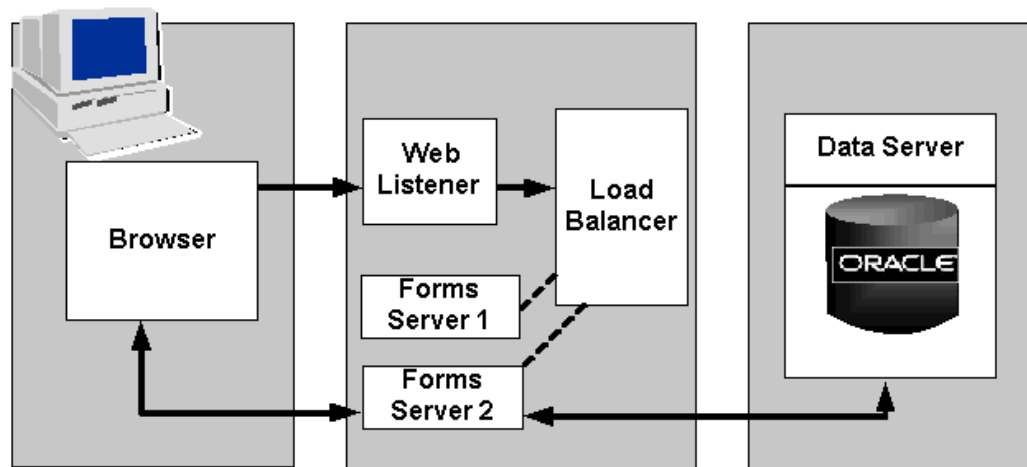
Once a connection has been made, many operations can be performed with little or no further interaction with the Forms server. For example, when a few field values change in response to a user action, there is no need to update the entire screen. In this scenario, only the changed fields are updated with the new values.

Load Balancing Among Forms Servers

Oracle Applications supports automatic load balancing among multiple Forms servers. Such a load balancing configuration utilizes an additional Forms component called the *Metrics server*, which acts as a single point of coordination for all Forms server requests.

The Metrics server is located on one application server. Metrics clients located on the other application servers periodically send load information to the Metrics server so that it can determine which application server has the lightest load.

Figure 1-7 Forms Server Load Balancing



Forms Server load balancing operates as follows.

1. The client requests a Forms session and the request is sent to the load balancer (Metrics Server), via the Web Listener.
2. The load balancer determines the least-loaded Forms server and returns its name via an HTML page.
3. The client can now access the least-loaded Forms server directly.
4. From this point on, all communication is between the client desktop and the designated Forms server.

Note: Using the 11.5.10 Rapid Install, you can if desired set up load balancing for Forms and Web servers during the initial installation.

Forms Listener Servlet Architecture

Oracle E-Business Suite Release 11i supports the use of socket mode for intranet users, HTTPS for Internet users, and HTTP mode for Forms-based (professional user interface) communication. The *Forms Listener Servlet* is a Java servlet that delivers the ability to run Oracle Forms applications over HTTP and HTTPS connections. It manages the creation of a Forms Server Runtime process for each client, as well as network communications between the client and its associated Forms Server Runtime process. The client sends HTTP requests and receives HTTP responses from the Web server, which acts as the network endpoint for the client.

Benefits of using the Forms Listener Servlet include:

- Ability to re-establish dropped network connections
- Fewer machines and ports need to be exposed at the firewall
- Easier firewall/proxy server configuration
- More robust and secure deployment over the Internet

Use of Forms Listener Servlet results in an approximately 40% increase in network traffic, since the HTTP protocol is more "chatty" than sockets. The actual performance overhead of this increase in network traffic will depend on existing network latency:

- LAN users may observe an 8-10% decrease in response time, but this is generally not noticeable given the bandwidth and speed of internal networks.
- WAN users (assuming a 150ms network latency) may observe up to a 30% decrease in response time performance, but this must be set against the extra security and robustness of the ability to run over the Internet using standard protocols.

Requirements for using the Forms Listener Servlet include:

- Installation of Developer 6i Patchset 7 or higher (see OracleMetaLink Note 125767.1, *Upgrading Developer 6i with Oracle Applications 11i*).
- Deployment of the Forms server and the Web server on the same node.

With Forms Listener Servlet, load balancing of Forms-based traffic is accomplished using *JServ Load Balancing*. This requires use of the iAS 1.0.2.2.2 Apache module, *mod_oprocmgr*, which also provides other new infrastructure capabilities such as automatic starting of processes, death detection and restart.

Note: For further details of the Forms Listener Servlet architecture, see <http://technet.oracle.com/products/forms/pdf/p7listenerservlet.pdf>. For steps needed to implement Forms Listener Servlet with Applications, see OracleMetaLink Note 201340.1, *Using Forms Listener Servlet with Oracle Applications 11i*.

Reports Server

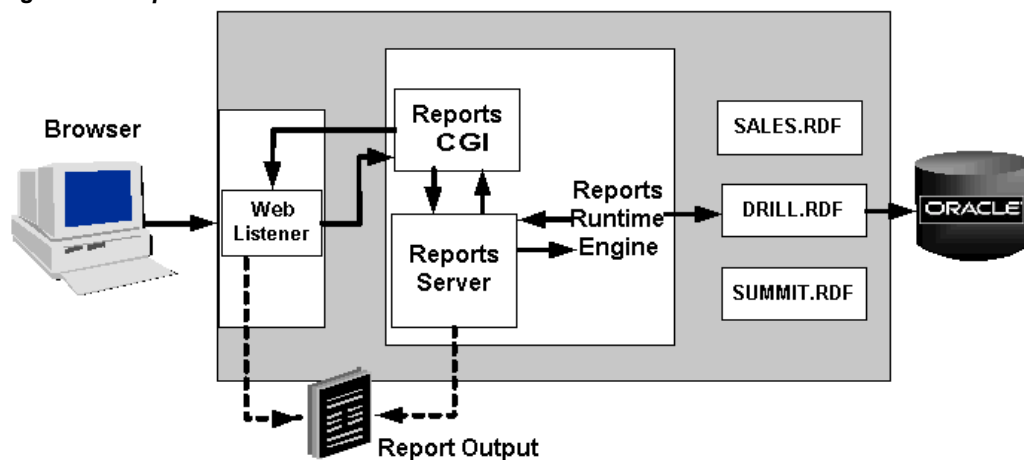
The Reports server is automatically installed on the same node as the Concurrent Processing server, and its reports are contained in the same directory as the concurrent processing reports. However, reports generated by the Reports server are monitored and administered separately from concurrent processing reports.

The Reports server dynamically selects the report language at runtime, so users see their reports in their preferred language.

Reports Server Architecture

The request for an HTML-based report is similar to the flow of any other HTML-based Applications request:

Figure 1-8 Reports Server Architecture



1. The user clicks the hyperlink of a function from a browser.
2. The browser makes a URL request to the Web listener.
3. The Web listener contacts the Reports server through the reports Web CGI. The reports Web CGI allows the user to run reports and see the output through a Web browser.
4. The Reports server starts the Reports runtime engine.
5. The Reports runtime engine locates the necessary reports and connects to the database to query the requested information.
6. The queried information is presented to the user in the form of an HTML page.

Clustering and Load Management

When there are a large number of users, it may be beneficial to have multiple Reports servers. In this configuration, one of the Reports servers is designated as the *master Reports server*. This server receives the initial request and distributes it to one of the other Reports servers, depending on the load each can handle. The master Reports server is capable of determining how many runtime engines each of the Reports servers can support.

Oracle E-Business Suite Release 11*i* supports multiple languages on a single Reports server.

Discoverer Server

The Discoverer server comprises Oracle Discoverer 10g, a key component of the Oracle 10g Application Server. Discoverer is an ad hoc query, reporting, analysis, and publishing tool that allows business users at all levels of an organization to gain immediate access to information from data marts, data warehouses, and online transaction processing (OLTP) systems. Report builders and analysts can easily create, modify, and execute ad hoc queries and reports. Casual users can navigate through predefined reports and graphs via business views that hide the complexity of the underlying data structures.

Oracle Discoverer 10g hides the complexity of the database, presenting it through an intuitive, easy to understand interface. It maps potentially unfamiliar database terms such as tables to familiar terms such as folders, enabling those unfamiliar with database syntax to access key information using standard business terms.

Discoverer 10g is tightly integrated with Oracle E-Business Suite Release 11*i*, allowing E-Business Suite users to employ Discoverer to analyze data from selected business areas in Financials, Operations, Human Resources, Purchasing, Process Manufacturing, Activity Based Management, and other products.

The Discoverer server complements the Reports server, by allowing performance of ad hoc queries and analysis of the resulting query output. It also allows users to perform projections based on possible changes to the business environment or other strategic factors.

Discoverer End User Layer

The Discoverer End User Layer is the key to providing ease of use to the underlying data in the Oracle Applications database. It is a layer of *metadata* (data describing data) that is located in the Oracle Applications database, separate from the actual Applications data.

Three software components are used to access the End User Layer:

- **Discoverer Administrator Edition** - Allows creation and maintenance of the End User Layer and manages the administration of security grants for users and responsibilities.
- **Discoverer Plus** - An applet used to create queries, create and carry out analytic calculations, chart selected data, share query results with users, and export Discoverer workbooks in various formats. Oracle provides predefined workbooks which can be used as bases for ad hoc queries. Discoverer Plus is located on the Discoverer server, to centralize administration.
- **Discoverer Viewer** - This utility allows execution of queries in existing workbooks, displaying the resulting business data in the form of tables and charts in the user's Web browser.

Note: For further details of using Oracle Discoverer with Oracle E-Business Suite Release 11i, see OracleMetaLink Note 313418.1, *Using Discoverer 10.1.2 with Oracle Applications 11i*.

Concurrent Processing Server

As described previously, user interactions with Oracle Applications data can be conducted via HTML-based Applications or the more traditional Forms-based Applications. However, there are also reporting programs and data updating programs that need to run either periodically, or on an ad hoc basis. These programs, which run in the background while users continue to work on other tasks, may require a large number of data-intensive computations, and are run using the *Concurrent Processing* architecture. Concurrent Processing is an Oracle Applications feature that allows these non-interactive and potentially long-running functions to be executed efficiently alongside interactive operations. It uses operating system facilities to facilitate background scheduling of data- or resource-intensive jobs, via a set of programs and forms. To ensure that resource-intensive concurrent processing operations do not interfere with interactive operations, they are run on a specialized server, the *Concurrent Processing server*.

Processes that run on the Concurrent Processing server are called *concurrent requests*. When you submit such a request, either through HTML-based or Forms-based Applications, a row is inserted into a database table specifying the program to be run. A *concurrent manager* then reads the applicable requests in the table, and starts the associated concurrent program.

Concurrent Manager Characteristics

Concurrent managers are fundamental to concurrent processing. Acting as a job scheduling and execution system, a concurrent manager:

- Is an executable that is registered as a program library within Oracle Applications, and which runs in its own operating system process
- Runs operating system processes called *target processes* (often referred to as *workers*), each of which can start one concurrent program at a time
- Can optionally run an *immediate program* that runs as part of the concurrent manager's own operating system process
- Can be allowed to run any concurrent program, or be specialized to run certain programs
- Operates during the days and times defined by a *work shift*

Types of Concurrent Manager

The *Internal Concurrent Manager* (ICM) controls all other concurrent managers. It administers the startup and shutdown of managers as defined by their work shift, monitors for process failure, and cleans up if a failure occurs. The ICM does not process concurrent requests itself (except for queue control requests, such as `ACTIVATE`, `DEACTIVATE`, or `ABORT`).

While the basic ICM definition should not be changed, you can if required modify the *sleep time* (number of seconds the ICM waits between checking for new concurrent requests), *PMON* (*process monitor*) *cycle time* (number of sleep cycles the ICM waits between checking for failed workers), and *queue size* (duration between checks for number of active workers, measured in PMON cycles). If Parallel Concurrent Processing (described below) is being used, you can also set some options for this.

The *Conflict Resolution Manager* (CRM) enforces rules designed to ensure that incompatible concurrent requests do not run in the same *conflict domain* (an abstract representation of the groupings used to partition data). As with the Internal Concurrent Manager, the basic CRM definition should not be changed, but you can modify the sleep time for each work shift, as well as some Parallel Concurrent Processing options.

The *Standard Manager* as shipped with Oracle Applications will accept and run any concurrent requests, as it has no specialization rules that would restrict its activities. Consequently, the definition of the Standard Manager should not be altered, as this might mean some programs could not run at all. Jobs should only be excluded from the Standard Manager after ensuring they can be run by an alternative manager, such as a product-specific manager or user-defined manager.

Transaction Managers support synchronous request processing, whereby a pool of server processes responds to requests from client programs. Instead of polling the concurrent requests table to obtain instructions, a transaction manager waits to be signaled by a client. An example is approval of an order, where execution of the request must take place quickly.

The relevant transaction manager program runs on the server, transparently to the client. All transaction programs for a given manager process run in the same database session. Communication between the client and the server is conducted synchronously via pipes, using the `FND_TRANSACTION.SYNCHRONOUS` function. At the end of program execution, the client program receives a completion message and a return value, for example denoting approval of the order. This strategy of using non-persistent connections between the client and Transaction Manager processes enables a small pool of server processes to service a large number of clients with near real-time response.

Setting Up Concurrent Managers

The *Oracle Applications System Administrator's Guide* gives full details of the steps and options involved in setting up and monitoring concurrent managers. Some of the key steps include:

- Name and description of the manager
- Assignment of a concurrent program library
- Assignment of work shifts to the manager
- Definition of the maximum number of workers (target processes) the manager can run concurrently
- Optionally specializing the manager to run certain types of requests

Tip: It is easier to identify the optimum number of workers by being conservative initially, and defining additional workers later if needed (subject to availability of system resources).

Multiple managers can be run on multiple nodes using *Parallel Concurrent Processing*, as described below.

Concurrent Processing Architecture

In Concurrent Processing, programs are run as operating system background processes. These programs may be written using a variety of Oracle tools, programming languages for executables, or the host operating system scripting language.

As noted above, a concurrent program that runs in the concurrent manager's own operating system process is known as an immediate program. Immediate programs run as a function within the concurrent manager's program library. Examples include PL/SQL programs. In contrast, a concurrent program that runs in a child process of the concurrent manager process is known as a *spawned program*. Examples include SQL programs, SQL Loader programs, Oracle Reports programs, spawned C programs, and host language programs such as UNIX shell scripts or Windows command files.

Note: While C programs can be run as immediate programs, it is advisable to run them as spawned programs. This simplifies maintenance, without introducing any disadvantages.

A concurrent request has a life cycle, which consists of three or possibly four phases:

Table 1-1 Concurrent Request Life Cycle

Phase	Activity
Pending	The request is waiting to be run
Running	The request is running
Completed	The request has finished
Inactive	The request cannot be run

A *concurrent program library* contains concurrent programs that can be called by a concurrent manager. An important example is the Oracle Application Object Library program library (FNDLIBR), which contains Oracle Applications immediate concurrent programs, and is assigned to the standard concurrent manager. Although each concurrent manager can only run immediate concurrent programs from its own concurrent program library, it can also run spawned or Oracle tool concurrent programs.

Various database tables are employed by the concurrent processing architecture:

Table 1-2 Concurrent Processing Database Tables

Table	Content
FND_CONCURRENT_REQUESTS	Details of user requests, including status, start date, and completion date
FND_CONCURRENT_PROGRAMS	Details of concurrent programs, including execution method, whether the program is constrained, and whether it must be run alone.
FND_CONCURRENT_PROCESSES	Cross-references between concurrent requests and queues, and a history of concurrent manager processes
FND_CONCURRENT_QUEUES	Information about each of the concurrent manager queues

Caution: Do not update these tables manually. You can (subject to your organization's archiving requirements) periodically run the "Purge Concurrent Requests and/or manager data" program to prevent these tables growing too large. See the *Oracle Applications System Administrator's Guide* for details.

Concurrent Processing Operations

Because the Internal Concurrent Manager controls all the other managers, it must be running before any other manager can be activated. Once the ICM has been activated, it starts a Service Manager on each node that is enabled for concurrent processing. Acting as an agent of the ICM, the Service Manager starts the concurrent managers on its node, excluding any managers that have been deactivated, or that have no current work shift. The ICM can be activated and deactivated from the operating system prompt, or Oracle Applications Manager. It can also be deactivated (but not activated) from the Administer Concurrent Managers form.

When the ICM is initiated on UNIX, the `$FND_TOP/bin/startmgr` program is invoked. This calls `$FND_TOP/bin/batchmgr`, which then:

1. Starts a shell process
2. Starts the ICM process using the command `FNDLIBR`, with startup parameters `FND`, `CPMGR`, and `FNDCPMBR`
3. Creates log files (`std.mgr` and `wnnn.mgr`) in `$APPLCSF/$APPLLOG`

Normally, `startmgr` is run by the user account that owns the application software (for example, `applmgr`). This account must have write privileges to the log and out directories where the log and output files respectively are written.

The ICM starts up a Service Manager on each node that is enabled for concurrent processing, by instructing the Applications listener on the node to spawn a process running the Service Manager executable (`FNDSM`). The listener must be configured to source the Applications environment file before `FNDSM` is spawned. Following startup, the Service Manager acts as an agent of the ICM to start and stop concurrent managers on that node, according to their defined work shifts.

Note: The Service Manager is a component of the Generic Service Management (GSM) architecture rather than Concurrent

Processing, although GSM and Concurrent Processing are closely integrated. GSM must be enabled in any AutoConfig-enabled Applications system.

Concurrent manager processes on a specific node can be seen by running the UNIX commands:

```
ps -ef | grep FNDLIBR
```

```
ps -ef | grep FNDSM
```

The Service Manager PID seen in the output of the second command can then, if desired, be used to locate all concurrent manager and service processes on the node, since the Service Manager is the parent process for them:

```
ps -ef | grep <sm_pid>
```

On Windows, the Task Manager can be used to locate concurrent manager processes. An FNDLIBR process runs for the Internal Concurrent Manager and each standard manager. The ICM can be distinguished by additional details being displayed, including some of the parameters it was started with.

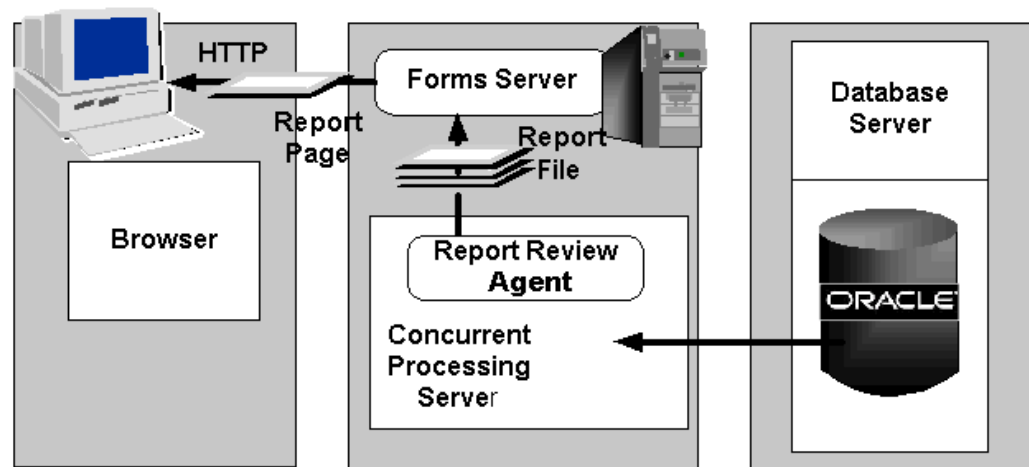
For every process that was successfully started at operating system level, the ICM inserts a row into FND_CONCURRENT_PROCESSES. It then updates the RUNNING_PROCESSES column to reflect the actual running processes as shown in FND_CONCURRENT_QUEUES.

Viewing Concurrent Processing Output

The output from a concurrent processing job goes through several stages before being displayed to the user.

1. The Concurrent Processing server communicates with the database server via Oracle Net.
2. The log or output file associated with a concurrent request is passed back to the *Report Review Agent*, also known as the *Web Review Agent*.
3. The Report Review Agent passes a file containing the entire report to the Forms server.
4. The Forms server passes the report back to the user's browser one page at a time.

Figure 1-9 Viewing Concurrent Processing Output



You can cater for your network capacity and report volume by using profile options to specify the maximum size of the files and pages that can be passed through the system.

Parallel Concurrent Processing

Parallel Concurrent Processing (PCP) allows concurrent processing activities to be distributed across multiple nodes in a Real Application Clusters (RAC) environment or similar cluster system. By distributing concurrent processing in this way, hardware resources can be fully utilized, maximizing throughput and providing resilience to node failure, while retaining a central point of control.

Parallel concurrent processing enables you to:

- Run concurrent processes on multiple nodes to improve concurrent processing throughput
- Continue running concurrent processes on the remaining nodes when one or more nodes fail
- Administer concurrent managers running on multiple nodes from any node in the cluster

One or more concurrent managers can be specified to run on one or more nodes, to best suit your processing needs and fully utilize available hardware resources. Parallel Concurrent Processing is described further in Chapter 9.

Managing Concurrent Processing

From the command line, two commands can be entered to control the Internal Concurrent Manager: `startmgr`, which starts the ICM; and `consub`, which is used to stop or abort the ICM, or request the ICM to check on the operating system process for each manager. In addition, an AutoConfig-enabled environment provides a number of scripts for starting and stopping application tier services from the command line. The script for concurrent processing startup and shutdown is `COMMON_TOP/admin/scripts/<CONTEXT_NAME>adcmctl.sh`.

The various components of the concurrent processing system can be managed from forms such as the Administer Concurrent Managers form, or from Oracle Applications Manager (OAM). In OAM, you can drill down to the required screen: for

example, Applications Dashboard > Concurrent Processing > Conflict Resolution Manager > View Processes.

Admin Server

The Admin (Administration) server is located on the node where you maintain the data model and data in your Oracle Applications database. You carry out the following operations from this server:

- **Upgrading Oracle Applications**

This process is conducted only when you are upgrading to a new release from an earlier one, such as 10.7 or 11.0. You use the *AutoUpgrade* utility (*adaimgr*) to upgrade Oracle Applications.

- **Applying database patches to Oracle Applications**

In general, *Applications patches* consist of files and scripts that update the file system and database objects. Most patches consist of a number of *patch driver files* that perform different functions. The *copy (c) driver* modifies Applications files, the *database (d) driver* modifies Applications database objects, and the *generate (g) driver* generates forms, reports, graphics, or message files. Newer patches combine the actions of the copy, database, and generate driver files into a single *unified (u) driver* file.

You use the *AutoPatch* utility (*adpatch*) to perform these updates. AutoPatch may also be used to apply cumulative patches such as *mini-packs* (for example, 11i.AD.H) and *maintenance packs* (for example, 11.5.10).

- **Maintaining Oracle Applications data**

Some features, such as Multiple Reporting Currencies, require regular maintenance to ensure that updates are propagated to the additional tables and schemas they use. The *AD Administration* utility (*adadmin*) enables you to carry out this and various other file system and database maintenance tasks.

Daily Business Intelligence (DBI)

Daily Business Intelligence (DBI) is a reporting framework that is integrated with Oracle E-Business Suite. It replaces the Business Intelligence System (BIS), and includes a new set of materialized views that pre-summarize transaction data. Using Daily Business Intelligence *overview pages*, managers can view summarized information across multiple organizations, drilling down to specific transaction details on a daily basis.

For example, the *Profit and Loss page* provides an overview of revenue, cost of goods sold, expenses, and gross margin by line of business. Managers can use this page to view revenue to date, track it against forecast, and compare it to previous periods. If actual revenue is trailing forecast revenue, managers can investigate the root cause, drilling down to specific lines of business, specific managers, or even specific customer invoices in Oracle Receivables.

Daily Business Intelligence resides in the same instance as the transactional system. This single instance architecture reduces the need for a separate maintenance and administration team, and optimizes reporting performance. It also utilizes the materialized view and incremental refresh capabilities of the Oracle database, thereby enabling organizations to refresh data daily, hourly, or at any required frequency.

Note: The old BIS views are still available for use, although they may occasionally be altered or replaced to reflect changes to underlying table definitions.

The Database Tier

The database tier contains the Oracle database server, which stores all the data maintained by Oracle Applications. The database also stores the Oracle Applications online help information.

More specifically, the database tier contains the Oracle data server files and Oracle Applications database executables that physically store the tables, indexes, and other database objects for your system.

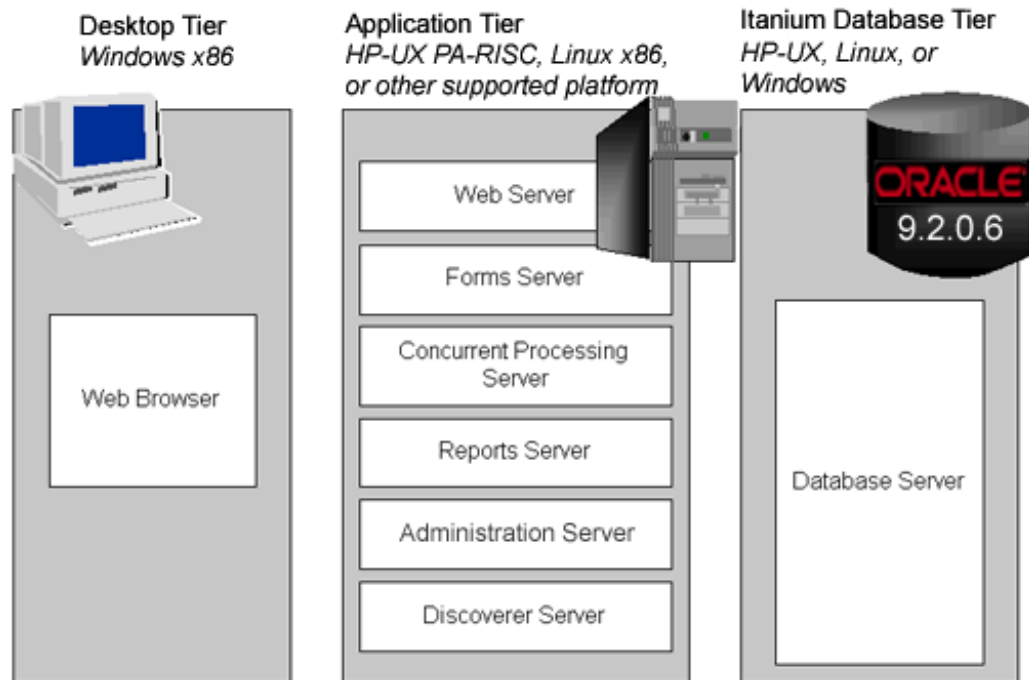
In general, the database server does not communicate directly with the desktop clients, but rather with the servers on the application tier, which mediate the communications between the database server and the clients.

Using a Split Configuration

The Oracle database server is sometimes available on platforms where Oracle Applications is not currently certified. In such a case, it may be possible to utilize a *split configuration*, where the database is installed on one platform and the application tier on another.

For example, the database tier can be deployed on an HP Itanium platform, and the application tier on a platform such as Intel x86.

Figure 1-10 Example of Deploying Oracle Applications on Heterogeneous Platforms



This enables the database to utilize the advanced performance features offered by the 64-bit Itanium architecture. It can also allow the application tier to be managed in a more cost-effective way.

The Oracle Applications Technology Layer

The Oracle Applications technology layer lies between the Oracle Applications technology stack and the Oracle Applications product-specific modules. It provides features common to all Oracle Applications products.

Products in the Oracle Applications technology layer include:

- Oracle Applications DBA (AD)
- Oracle Applications Object Library (FND)
- Oracle Applications Utilities (AU)
- Oracle Common Modules (AK)
- Oracle Workflow (WF)
- Oracle Alert (ALR)
- Oracle Applications Framework (FWK)
- Oracle XML Publisher (XDO)

Oracle Applications DBA (AD)

The Applications DBA product provides a set of tools for administration of the Oracle Applications file system and database. AD tools are used for installing, upgrading, maintaining, and patching the Oracle Applications system.

The AD utilities include:

- **AD Administration** - Performs general maintenance tasks for Oracle Applications.
- **AD Merge Patch** - Merges multiple patches into a single, integrated patch.
- **AutoConfig** - Manages configuration changes in an Oracle Applications system.
- **AutoPatch** - Applies patches and adds new languages and products to an Oracle Applications system.
- **AutoUpgrade** - Upgrades a system to the latest version of Oracle Applications.
- **Rapid Clone** - Used to copy (clone) an Oracle Applications system.
- **Rapid Install** - Sets up a fully configured Oracle Applications system, including the latest certified technology stack and all patches, mini-packs, and other updates.

Oracle Common Modules (AK)

AK is an active data dictionary that enables you to define Oracle Applications components for the HTML-based Applications, and generate many of the characteristics of Oracle Applications at runtime.

The *Oracle Common Modules* can be used to develop inquiry applications for the HTML-based Applications, without the need for any programming. They allow storage of language-translated labels for all the attributes on a transaction page, thus assisting with the provision of support for multiple languages.

For example, the AK Runtime Dictionary may be used to define an attribute or reusable component such as the customer name attribute, which can be reused any time a customer name field is displayed on an HTML page.

Note: AK is superseded by JDeveloper's *MetaData Services* (MDS) repository in Framework 5.7 (with Applications Release 11.5.8).

Oracle Applications Utilities (AU)

The Applications Utilities (AU) component is used to maintain the Oracle Applications system.

AU hosts a collection of files copied from other products. This allows generating on-site classes of files such as Forms and reports, or running reports invoked by the Reports server from a standard, shared location. Generating forms or reports may require access to shared PL/SQL libraries, so these files are copied to AU_TOP as well.

The Oracle Applications Java files are stored in AU_TOP, as well as in JAVA_TOP and <PROD>_TOP.

Oracle Application Object Library (FND)

The Oracle Application Object Library is a key component of the Applications technology layer. It consists of a collection of reusable code, programs, and database objects that provides common functionality across all products.

Oracle Application Object Library offers many features to make system administration easier, such as security setup and maintenance, and management of concurrent processing. Using Application Object Library ensures that the processing of flexfields or the procedure for report submission, for example, does not vary from one product to another. Oracle Application Object Library also provides capabilities for developers to extend the operation of Oracle Applications by allowing the creation of custom programs that interact with the base modules.

End User Features

Oracle Application Object Library includes several features that help provide uniformity of function across the various Applications products.

Standard User Interface

Oracle Application Object Library supports the integration of Oracle Applications by providing standardized functionality and capabilities across all products so that the look and feel remains the same from product to product.

Shared Flexfield value sets

Flexfields allow the entry of certain important information to be standardized across all products. One example is the Accounting Flexfield, which is used by Financials products and Manufacturing products.

Standard Report Submission (SRS)

The procedure to submit a background report to the concurrent manager using SRS is the same, regardless of the product that owns the report. SRS takes advantage of shared flexfield value sets.

Applications Online Help

The presentation of Applications Online Help is also standardized across all products.

Developer Features

Oracle Application Object Library provides many features for developers creating custom forms, reports, or programs that interface with Oracle Applications:

- The same coding and Graphical User Interface (GUI) standards used by Oracle Applications developers are available for custom development.
- Custom reports can be integrated into Standard Report Submission so that they can be submitted and monitored using the same procedures as other Oracle Applications reports, and developers can set up certain menus and responsibilities to access custom reports or standard objects.
- Flexfields used on custom forms can take advantage of existing flexfield capabilities such as value sets, validation, and security rules.
- Custom menus and responsibilities can be seamlessly integrated with Oracle Applications.

Features for System Administrators

Oracle Application Object Library provides many features to simplify administration of Oracle Applications, enabling the system administrator to carry out routine tasks quickly and easily. These features include:

- Registering new Oracle Applications users, and giving them access to only those Forms, functions, and reports they need to do their jobs.
- Deciding which users have access to each product, and within a product, which Forms, functions, and reports a user can access.
- Monitoring what users do, and when, via comprehensive auditing capabilities.
- Setting user and system *profiles* to modify the look and behavior of Oracle Applications products; profiles can be set at site, application, responsibility, and user levels.
- Monitoring and controlling concurrent processing using interfaces such as Oracle Applications Manager (OAM).

Oracle Application Object Library Security

Oracle Application Object Library controls access to the data in Oracle Applications via user sign-ons and responsibilities. Each user must have a valid user name and password to gain access to Oracle Applications.

A *responsibility* is a level of authority in Oracle Applications that lets Applications users access only those functions and data appropriate to their roles in the organization. Responsibilities allow access to a specific product, operating unit, set of books, and a restricted list of windows, functions, reports, and groups of products, or data groups.

Note that the Forms available from the navigation menus vary by responsibility. For example, the Purchasing User navigation menu does not include all the forms that are available to the Purchasing Superuser navigation menu.

When you install Oracle Applications, a standard Applications user called SYSADMIN is created for you. Several default responsibilities are also created. Since the SYSADMIN sign-on is automatically assigned to the System Administrator responsibility, you can use SYSADMIN to create new user signons and assign them to responsibilities. You can also create any custom responsibilities you need.

Oracle Workflow (OWF)

Oracle Workflow provides an infrastructure for the enterprise-wide communication of data related to defined business events, providing the capabilities needed to:

- Manage enterprise business processes that may span trading partners
- Support standard and personalized business rules
- Streamline and automate transaction flows
- Manage exceptions without manual intervention

Oracle Workflow lets you model and maintain your business processes using a graphical workflow builder. You can model and automate sophisticated business processes, defining processes that can loop, branch into parallel flows and rendezvous, decompose into sub-flows, branch on task results, time out, and more.

Acting as a *system integration hub*, Oracle Workflow can apply business rules to control objects and route them between applications and systems. It extends the reach of business process automation throughout an enterprise and beyond, to include any email user, web user, or system, enabling people to receive, analyze, and respond to *notifications* needing their attention. Users can respond to a notification via any standard email system or standard Web browser.

Components

Oracle Workflow Builder provides a graphical drag and drop process designer. You can create and evolve business processes to incorporate existing business practices between your organization and customers or suppliers, without modifying existing business processes and without changing applications code.

The Oracle Workflow *Business Event System* provides a workflow-enabled solution for your enterprise application integration requirements. The Business Event System is an application service delivered with Oracle Workflow that uses Oracle Advanced Queuing technology to communicate business events between systems. The Business Event System supports the following types of integration:

- Message-based point-to-point system integration
- System integration messaging hubs
- Distributed applications messaging

The Business Event System uses Oracle Advanced Queuing to propagate messages between communication points on systems, called *agents*, using a specified protocol. Events received from external systems are processed by an agent listener that runs on that agent's queue.

The *Oracle Workflow Event Manager* enables registration of significant business events for selected applications, including functions that generate the XML event messages. Users of those applications can register *subscriptions* on events that are significant to their systems, to take actions such as triggering custom code.

Operation

When a business event occurs, the Workflow Event Manager executes any subscriptions registered on the event. For local events, the subscribing code can be executed synchronously, in the same database transaction as the code that raised the event, or asynchronously, deferring costly subscription processing to a later time, and thus allowing control to be returned more quickly to the calling application. Events can

also be received asynchronously from external systems. Before producing the XML event message, the Event Manager minimizes processing by checking whether event information is required by subscriptions on the event.

You can review and respond to your business process notifications from one central window, known as the *worklist*, using a standard Web browser. This offers the flexibility to prioritize tasks and to define sort criteria, giving you the flexibility to organize your work the way you wish. For example, you can group notifications by type or subject, to avoid having to jump from one context to another. Alternatively, you can focus on time critical tasks first, sorting by priority or due date. Oracle Workflow is fully integrated with the Oracle E-Business Suite, providing the ability to drill down to any Oracle E-Business Suite or associated URL to view or complete a transaction.

Additional Capabilities

The flexibility of the powerful *Workflow Engine* event activities enable you to model business events within workflow processes. Event activities can be used to model content-based routing, transformations, error handling, and so on. A workflow process can be started or processed by an inbound message, and can send an outbound message or raise an event to the Event Manager. XML function activities give you access to event content data within workflow processes. Workflow processes based on business events give the greatest flexibility when implementing an integration solution. However, the Business Event System can also run independently of the Workflow Engine, to enable point-to-point messaging to be utilized.

You can perform complex transformations between different formats required for your business documents. Oracle Workflow allows you to apply a stylesheet to an XML event message. In addition, when queues are defined within the Business Event System, you specify the logic used to enqueue and dequeue messages. This logic, called a *queue handler*, can include transformations.

Oracle Workflow also allows you to take advantage of XML support in the Oracle database. Oracle9i and later releases deliver native support for XML via a new XML data type, which makes the manipulation of XML data and documents in applications seamless and straightforward. From Oracle9i, the database server offers the capability to generate, massage, and transform XML data and documents inside the runtime engine itself, giving excellent scalability and performance.

Oracle Alert (ALR)

Oracle Alert (ALR) allows you to email system notifications to users when an exception or event has occurred. Some products are delivered with predefined alerts, which can be used to notify users about specified database exceptions as they occur, and perform routine tasks automatically according to a schedule you define.

For example, you can configure Oracle Alert to send an email to key database administrators when a tablespace in the Oracle Applications database does not have adequate free space.

Oracle XML Publisher (XDO)

Oracle XML Publisher is a Java-based product based on the World Wide Web Consortium (W3C) *Extensible Stylesheet Language* (XSL). Specifically, XML Publisher utilizes the *XSL-FO* standard to transform XML data into a *formatting object* (FO). A formatting object contains both data and formatting information, and can be further transformed to an output format such as Portable Document Format (PDF).

XML Publisher uses *data definitions* and *templates* to produce output reports in the desired format. A data definition is a data source (or a combination of data sources) that either is XML or can produce XML. Examples include output from concurrent programs and Web services. A template is a report definition, which sets out how a report should look. The template layout can be user-specified. Supported templates currently include RTF, PDF Forms, and XSL.

Key features of XML Publisher include:

- Provides a template-based, easy-to-use publishing solution that enables end-users to develop and maintain report formats to suit specific business needs.
- Allows users to employ familiar desktop tools such as Adobe Acrobat, Microsoft Excel, and Microsoft Word to create and maintain reports in their preferred format, and then use XML Publisher to convert these documents to the XSL-FO format.
- Offers a variety of options for published documents, such as multiple output formats, multiple languages, and multiple delivery options.

Core Components

The core components of XML Publisher are a Java-based set of publishing tools accessible via Java APIs from Oracle Applications or from any Java-based application.

- **FO Processor** - The formatting object that results from the application of the XSL-FO template to the XML data is used by the FO Processor to generate the required output document. Currently the FO Processor only supports PDF as an output format; other formats such as HTML and RTF will be supported in future releases of XML Publisher.
- **PDF Document Merger** - The PDF Document Merger accepts XML data and a PDF document as input, and uses a mapping between the XML and the fields in the document to merge the data into the PDF document.
- **PDF Form Processor** - The PDF Form Processor is used to add attributes such as watermarks to a merged document.
- **RTF Template Parser** - Report templates can be created in the *Rich Text Format* (RTF) document format, and converted to an XSL-FO format that can be applied to XML data.

Template Manager

The Template Manager enables you to upload and maintain your templates and data sources. Key features include:

- **Data Source Registration** - Data sources that generate XML data can be registered with XML Publisher. These can be concurrent programs such as Oracle Reports, or BC4J objects that are able to generate XML data.
- **Template Registration** - Templates used to format report data must be registered with XML Publisher. They can then be applied to the report data to create the final published output. Supported template types include PDF, RTF, and XSL-FO.
- **XML Publisher APIs** - XML Publisher provides APIs to allow other Oracle Applications products to communicate directly with the underlying processes.

Applications File System

Introduction

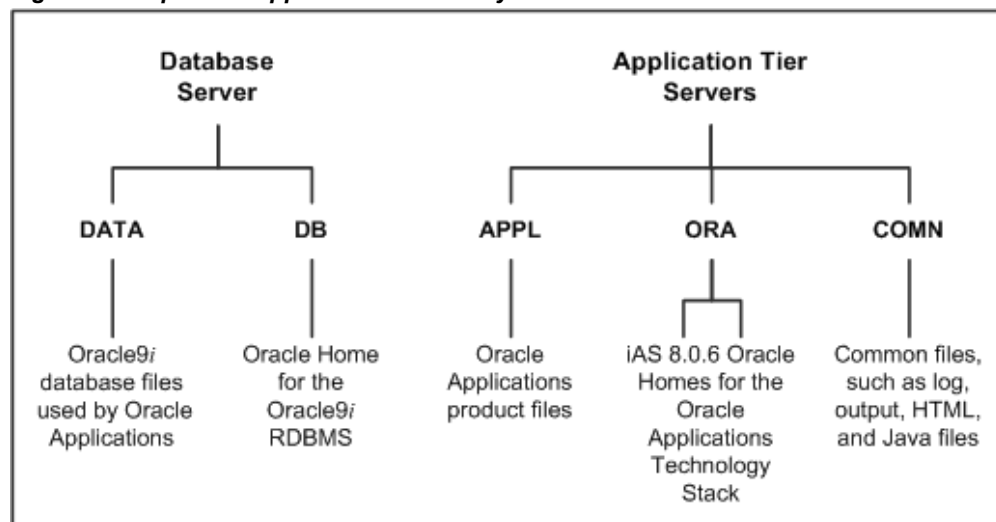
An Oracle Applications Release 11i system utilizes components from many Oracle products. These product files are stored below a number of key top-level directories on the database and application server machines.

Note: No Applications files are installed on desktop client machines, although JAR files and their supporting utilities are downloaded as required.

Depending on how you chose to install Applications, these product directories may be located on a single machine (the simplest case) or on multiple machines (the most common type of deployment). Operating system *environment settings* indicate the location of the various files in the file systems of the database and application server machines. This chapter discusses the association between these environment settings and the corresponding files and directories.

Note: *<dbname>* is the name of your system determined through Rapid Install at the time of installation. For example, PROD.

Figure 2-1 Top-Level Applications Directory Structure



- The `<dbname>DATA` or `DATA_TOP` directory is located on the database server machine, and contains the system tablespaces, redo log files, data tablespaces, index tablespaces, and database files.
- The `<dbname>DB` directory is located on the database server machine, and contains the `ORACLE_HOME` for the Oracle9i database.
- The `<dbname>APPL` or `APPL_TOP` directory contains the product directories and files for Oracle Applications.
- The `<dbname>ORA` directory contains the `ORACLE_HOMEs` for the Applications technology stack components.
- The `<dbname>COMN` or `COMN_TOP` (or `COMMON_TOP`) directory contains directories and files used across products.

Oracle Applications Environment

Oracle Applications makes extensive use of environment settings to locate executable programs and other files essential to Applications operation. These environment settings are defined when you install Oracle Applications. Many of the settings are defined by information you provide when running Rapid Install, while others have the same values in all installations.

The environment settings and their associated values are stored in *environment files*, which have a `.env` suffix (`.cmd` on Windows). Environment files and settings are discussed in more detail later in this chapter.

The DATA Directory

The `<dbname>DATA` file system contains the data (`.dbf`) files of the Oracle database. Rapid Install installs the system, data, and index files in directories below several mount points on the database server. You can specify the directory names of the mount points on the database server during installation.

DB and ORA Directories

Oracle Applications supports employing an Applications database of one version, while linking Applications programs using the tools from a second or third version of the database. This *multiple ORACLE_HOMEs* configuration allows new features of the database to be supported, while maintaining compatibility with earlier releases.

Release 11i utilizes three `ORACLE_HOMEs`:

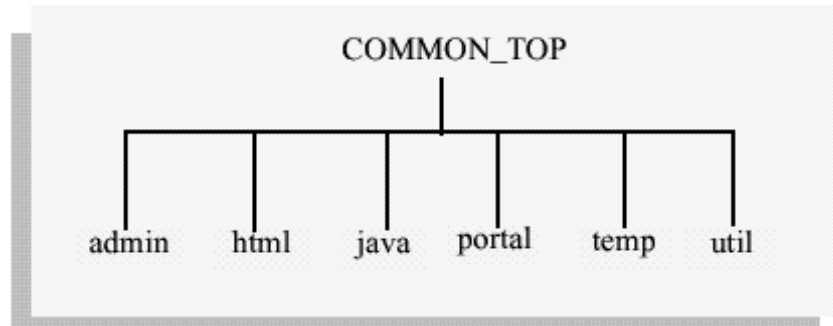
- The 9.2.0 `ORACLE_HOME` (Applications database home) is located in the `<dbname>DB` directory. It contains the files needed for running and maintaining the Oracle Applications database.
- The 8.0.6 directory contains the `ORACLE_HOME` for the Developer 6i products (Forms, Reports, and Graphics). The product libraries in the 8.0.6 `ORACLE_HOME` are used to relink Oracle Applications executables.
- The iAS directory, also under the `<dbname>ORA`, contains the `ORACLE_HOME` for Oracle9i Application Server.

Oracle E-Business Suite is always certified with database server patchsets (minor maintenance releases). For example, an E-Business Suite release that is certified with the Oracle9i Release 2 database will also be certified with database patchset 9.2.0.5.

The COMN Directory

The `<dbname>COMN` or `COMMON_TOP` directory contains files used by many different Oracle Applications products, and which may also be used with third-party products.

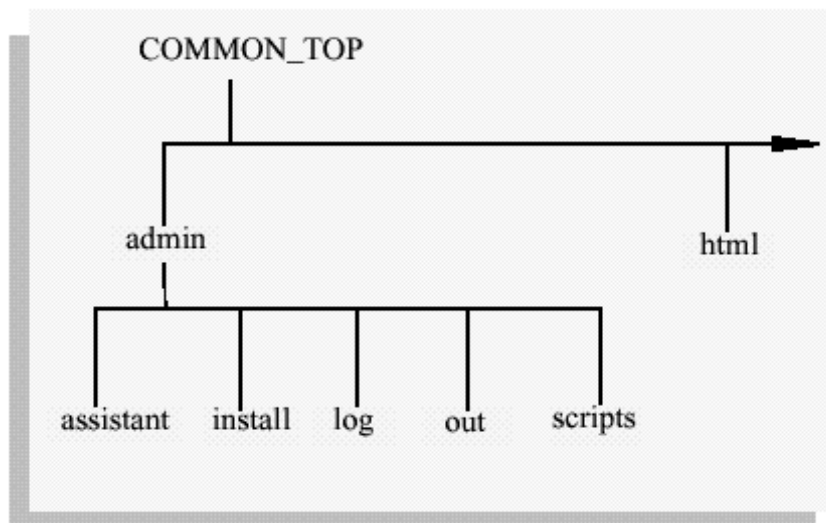
Figure 2-2 COMMON_TOP Directory Structure



The admin directory

The `admin` directory, under the `COMMON_TOP` directory, is the default location for the concurrent manager log and output directories. When the concurrent managers run Oracle Applications reports, they write the log files and temporary files to the `log` subdirectory of the `admin` directory, and the output files to the `out` subdirectory of the `admin` directory.

Figure 2-3 COMMON_TOP admin Directory Structure



You can change the location the concurrent managers write these files to, so that, for example, the log and output files are written to directories in each `<PROD>_TOP` directory. This may be more desirable in terms of disk space management, or the need to avoid a possible performance bottleneck on a system that has a high concurrent processing throughput.

Note: For further details, see *Overview of Concurrent Processing, Oracle Applications System Administrator's Guide*.

The *install* subdirectory of the admin directory contains scripts and log files used by Rapid Install. The *scripts* subdirectory of admin contains scripts used to start and stop services such as listeners and concurrent managers.

The html directory

The OA_HTML environment setting points to the *html* directory. The Oracle Applications HTML-based sign-on screen and Oracle HTML-based Applications HTML files are installed here. The *html* directory also contains other files used by the HTML-based products, such as JavaServer Page (JSP) files, Java scripts, XML files, and style sheets. Rapid Install and the AD utilities copy the HTML-based product files from each <PROD>_TOP directory to subdirectories in the *html* directory.

The java directory

The JAVA_TOP environment setting points to the *java* directory. Rapid Install installs all Oracle Applications JAR files in the Oracle namespace of this JAVA_TOP directory. The *java* directory also holds third-party Java files used by Oracle Applications, as well as other zip files.

The portal directory

The *portal* directory contains the Rapid Install Portal files. The Rapid Install Portal is a web page that provides access to post-install tasks that may be necessary for your installation, plus server administration scripts, installation documentation, and online help. Using a browser, you can view the Rapid Install Portal after you run Rapid Install.

The temp directory

The *temp* directory is used for caching by some products such as Oracle Reports.

The util directory

The *util* directory contains the third-party utilities licensed to ship with Oracle Applications. These include, for example, the Java Runtime Environment (JRE), Java Development Kit (JDK), and the Zip utility.

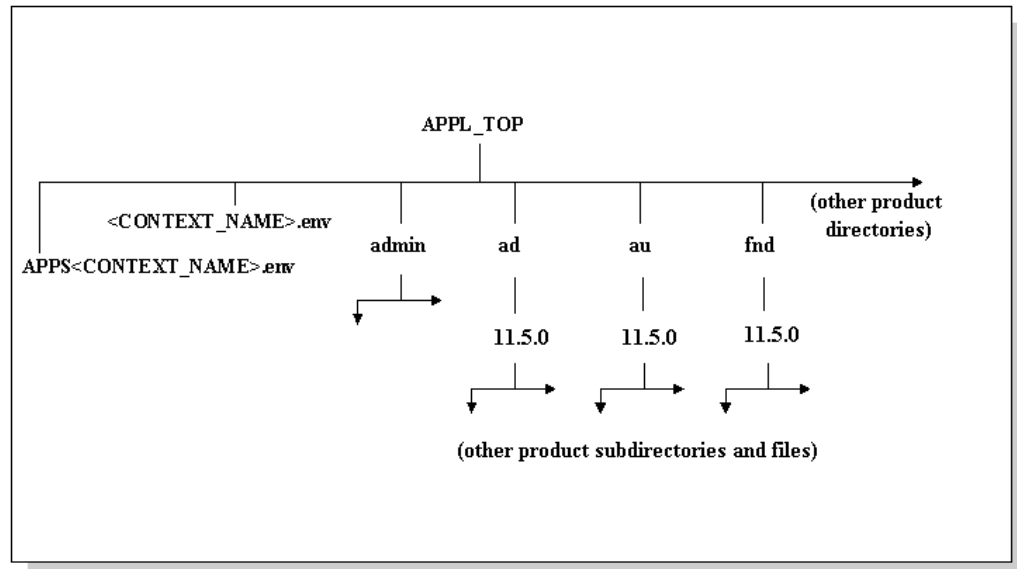
The scripts directory

The *scripts* directory contains application tier control scripts such as *adstrtal.sh* and *adstpall.sh*, which are located in the <CONTEXT_NAME> subdirectory.

The APPL Directory

Oracle Applications files are stored in the <dbname>APPL directory, which is known as the *APPL_TOP* directory.

Figure 2-4 APPL_TOP Directory Structure



The APPL_TOP directory contains:

- The core technology files and directories.
- The product files and directories (for all products).
- The main applications environment file, called `<CONTEXT_NAME>.env` on UNIX, and `<CONTEXT_NAME>.cmd` on Windows.
- The consolidated environment file, called `APPS<CONTEXT_NAME>.env` on UNIX, and `APPS<CONTEXT_NAME>.cmd` on Windows.

Note: CONTEXT_NAME is the *Applications context*, described further in Chapter 5. Its default value is `<SID>_<hostname>`.

Rapid Install creates a directory tree for every Oracle Applications product in this APPL_TOP directory, whether licensed or not.

Warning: All Oracle Applications products are installed in the database and the file system, regardless of registration. Do not attempt to manually remove files for unregistered products.

Rapid Install installs a new Applications top directory when you upgrade. Rapid Install does not delete any existing product files from earlier releases, but unloads new product files into a new `<dbname>APPL` directory tree.

Each Applications top directory is associated with a single Oracle Applications database on the Oracle database server. If you install both a Vision Demo system and a test system, you must use Rapid Install to lay down two file systems, one for each system.

Product Directories

Each product has its own subdirectory under APPL_TOP. The subdirectories are named in accordance with the product's standard abbreviation, such as *gl* for Oracle General Ledger. Within each product directory is a subdirectory that is named using the base

Oracle Applications release number, such as 11.5.0. This directory contains the various subdirectories for the product files.

<PROD>_TOP Directory

The <APPL_TOP>/<prod>/<version> path is known as the *product top directory* (<PROD>_TOP), and its value is stored in the <PROD>_TOP environment variable.

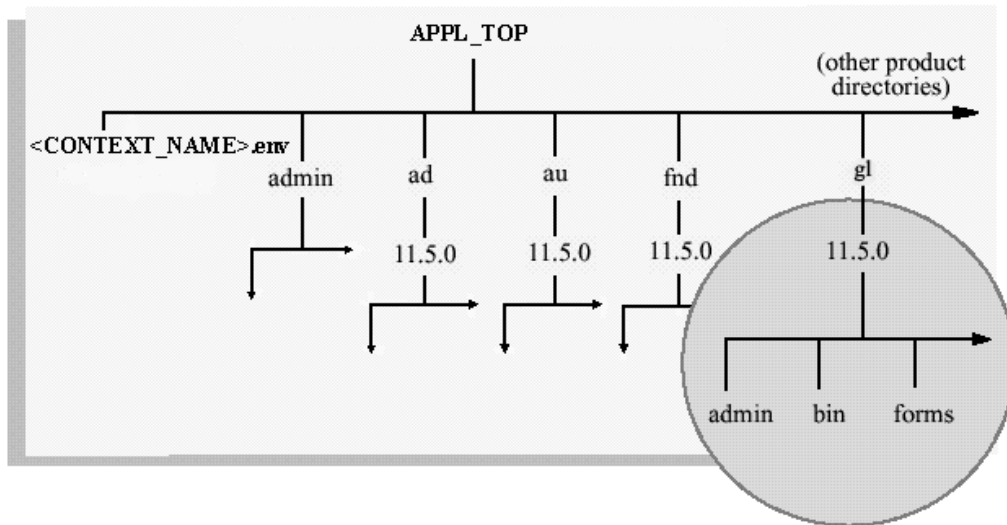
For example, if APPL_TOP=/d01/oracle/prodapps, then the value contained in the AD_TOP environment variable is /d01/oracle/prodapps/ad/11.5.0, and the AD_TOP environment variable points to the <APPL_TOP>/ad/11.5.0 directory.

For the same APPL_TOP, the value of AU_TOP is /d01/oracle/prodapps/au/11.5.0, and the AU_TOP environment variable points to the <APPL_TOP>/au/11.5.0 directory. The same principle applies to all directories, apart for the admin directory.

Product Files

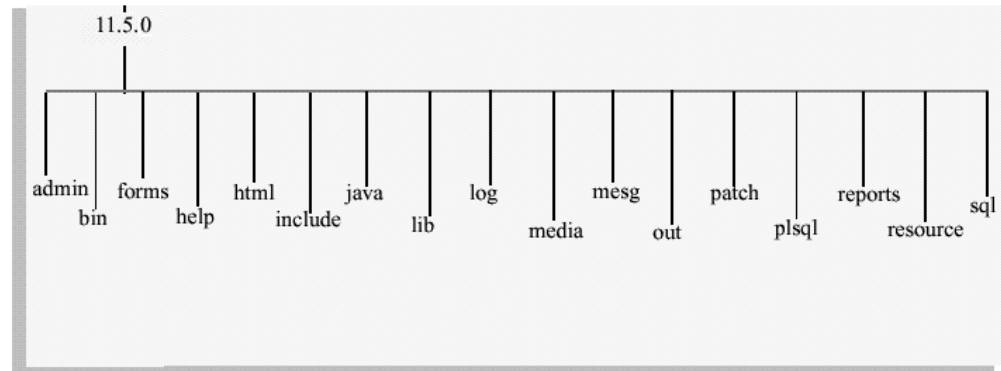
Each <PROD>_TOP directory, such as <APPL_TOP>/gl/11.5.0, contains subdirectories for product files. Product files include forms files, reports files, and files used to upgrade the database. To display data entry forms for Oracle General Ledger, for example, Oracle Applications accesses files in the forms subdirectory under the 11.5.0 directory.

Figure 2-5 APPL_TOP Directory Structure



Within each <PROD>_TOP directory, the product's files are grouped into subdirectories according to file type and function. The next figure expands the inset to show the full directory structure for gl.

Figure 2-6 Detail of gl Directory Structure



The following table summarizes the product subdirectories and the types of files each one may contain. Note that not all products contain all the subdirectories listed in this table.

Table 2-1 Applications Directory and File Types

Subdirectory Name	Description
admin	The <PROD>_TOP/admin directory contains <i>product-specific</i> files used by AutoUpgrade to upgrade each product. This is in distinction to the <APPL_TOP>/admin directory, which contains upgrade-related files for <i>all</i> products.
driver	Contains driver files (.drv files). AutoUpgrade processes in several phases, and each phase is controlled by a driver file.
import	Contains DataMerge files used to upgrade seed data.
odf	Contains object description files (.odf files) used to create tables and other database objects.
sql	Contains SQL*Plus scripts used to upgrade data, and .pkh, .pkb, and .pls scripts to create PL/SQL stored procedures.
bin	Contains concurrent programs, other C language programs and shell scripts for each product.
forms	Contains Oracle Forms generated runtime (.fmx) files (Oracle Forms form files).
help	Contains the online help source files. Within this directory are subdirectories for each language installed.
html	Contains HTML, JavaScript, and Java Server Page files, primarily for Self Service Applications products.

Subdirectory Name	Description
include	Contains C language header (.h) files that may be linked with files in the lib directory. Not all products require this directory.
java	Contains JAR files (Java Archive files) and Java dependency files. Copies of JAR files are also located in JAVA_TOP.
lib	<p>Contains files used to relink concurrent programs with the Oracle server libraries. These files include:</p> <ul style="list-style-type: none"> • object files (.o on UNIX, .OBJ on Windows), with compiled code specific to one of the product's programs. • library files (.a on UNIX, various including .DLL on Windows), with compiled code common to the product's programs. • make files (.mk) that specify how to create executables from object files and library files.
log and out	<p>Contains output files for concurrent programs:</p> <ul style="list-style-type: none"> • .mgr (master log file for concurrent manager) • .req (log file for a concurrent process) <p>Note that <i>log</i> and <i>out</i> subdirectories under a product directory are not used if you choose to set up a common directory for log and output files (FND_TOP is the only exception to this).</p>
media	Contains .gif files used in the display of text and graphics on the desktop client.
mesg	Concurrent programs also print messages in the log and output files. This directory contains the .msb files (binary message files used at runtime), and language-specific message files (such as a US.msb file for American English and a D.msb file for German). The files contain the forms messages that are displayed at the bottom of the screen or in popup windows.
patch	Updates to the data or data model utilize this directory to store the patch files.
plsql	Location where .pll files (PL/SQL library files for Oracle Reports) are unloaded. Later in the installation, they are moved to the <i>plsql</i> subdirectory in the AU_TOP directory.
reports	Contains Oracle Reports platform-specific rdf binary report files for each product. Reports for each language are stored in subdirectories of the <i>reports</i> directory.

Subdirectory Name	Description
resource	Contains .pll files (PL/SQL library files for Oracle Forms), which, like the <i>plsql</i> directory files, are later copied to AU_TOP.
sql	Contains .sql files (SQL*Plus scripts) for concurrent processing.

Language Files

When you install Oracle Applications in a language other than American English, each product tree includes directories that use the NLS language code. These directories hold translated data, forms, message, and reports files. For example, the language directory named D designates German. The data loader files in the D subdirectory of *admin* contain the German translation of the product seed data. The D subdirectory of reports holds Oracle Reports files translated into German.

The US subdirectory in the forms directory holds Oracle Forms forms in American English. The D directory in the *forms* directory holds the same forms translated into German. The *mesg* directory holds message files in both American English and German.

Note: For further details, see the *Oracle Globalization Support Guide*.

Core Technology Directories

The *admin*, *ad*, *au*, and *fnf* directories are the core technology directories.

The admin directory

This directory contains files and scripts used by the AD utilities during the upgrade and maintenance processes. These include:

- The *adovars.env* environment file, which defines certain file and directory locations.
- Text files read by AutoUpgrade.
- Scripts run during the upgrade.
- A *preupg* directory containing product-specific pre-upgrade scripts.
- *<SID>/log* and *<SID>/out* directories for upgrade, log, and output files respectively.
- A *<SID>/restart* directory where AD programs create restart files.
- The Applications context file, *<CONTEXT_NAME>.xml*.

The ad (Applications DBA) directory

This directory contains installation and maintenance utilities, including:

- AD Administration (adadmin)
- AutoUpgrade (adaimgr)
- AutoPatch (adpatch)
- AutoConfig (adconfig.sh)

The au (Applications Utilities) directory

This directory contains product files that are consolidated in a single location for optimal processing. These files include:

- PL/SQL libraries used by Oracle Reports, in the *plsql* subdirectory.
- PL/SQL libraries used by Oracle Forms, in the *resource* subdirectory.
- Oracle Forms source files, in the *forms* subdirectory.
- A copy of all Java files used by JInitiator when regenerating the desktop client JAR files, in the *java* subdirectory.
- Certain reports needed by products such as Discoverer, in the *reports* subdirectory.

The **fn**d (Application Object Library) directory

This directory contains the scripts and programs that are used as the foundation for all Applications products to build data dictionaries, forms and C object libraries.

Distributing the APPL_TOP Across Disks

The Oracle Applications file system on the application tier requires a significant amount of disk space. If desired, you can distribute the APPL_TOP files across multiple disks, by choosing multiple mount points when running Rapid Install. You can also install the OA_HTML, JAVA_TOP, and COMMON_TOP on mount points outside the APPL_TOP location.

For example, you could install Oracle General Ledger (GL) product files in one APPL_TOP, with Oracle Purchasing (PO) and Oracle Payables (AP) files stored in a second APPL_TOP directory on a different file system. The system knows the location of each product top because Rapid Install defines the full path to the directory in the <PROD>_TOP parameter in the <CONTEXT_NAME>.env file.

Note: When distributing the files across disks, the four core technology directories (admin, ad, au, and fnd) must always be on the same disk, and must all share the same APPL_TOP.

Environment Settings

Rapid Install creates several environment files that set up the Oracle database, the Oracle technology stack, the Oracle HTTP server, and the Oracle Applications environments.

The location of these environment files is shown in the following table:

Table 2-2 Locations of Applications Environment Files

Filename	Location	Environment
<CONTEXT_NAME>.env or <CONTEXT_NAME>.cmd	9.2.0 ORACLE_HOME	Oracle Server Enterprise Edition
<CONTEXT_NAME>.env or <CONTEXT_NAME>.cmd	iAS ORACLE_HOME	HTTP Server
<CONTEXT_NAME>.env or <CONTEXT_NAME>.cmd	8.0.6 ORACLE_HOME	Oracle technology stack
<CONTEXT_NAME>.env or <CONTEXT_NAME>.cmd	APPL_TOP	Applications
APPS<CONTEXT_NAME>.env or APPS<CONTEXT_NAME>.cmd	APPL_TOP	Consolidated setup file

On UNIX, Oracle Applications includes a consolidated environment file called *APPS<CONTEXT_NAME>.env*, which sets up both the Oracle Applications and Oracle technology stack environments. When you install Oracle Applications, Rapid Install creates this script in the APPL_TOP directory. Many of the parameters are specified during the install process.

On Windows, the equivalent consolidated environment file is called *%APPL_TOP%\envshell.cmd*. Running it creates a command window with the required environment settings for Oracle Applications. All subsequent operations on the APPL_TOP (for example, running adadmin or adpatch) must be carried out from this window.

The following table lists the key environment settings in *APPS<CONTEXT_NAME>.env*.

Table 2-3 Key Environment Settings

Parameter	Description
APPLFENV	The name of this environment file, <CONTEXT_NAME>.env. If you rename the environment file, change this parameter.
PLATFORM	The execution platform. The value should match the value in the APPL_TOP/admin/adpltfm.txt.
APPL_TOP	The top-level directory for this Oracle Applications installation.
FNDNAM	The name of the ORACLE schema to which the System Administration responsibility connects. The default is APPS.
GWYUID	The public ORACLE username and password that grants access to the Oracle Applications initial sign-on form. The default is APPLSYS PUB/PUB.

Parameter	Description
FND_TOP	The path to the Application Object Library directory.
AU_TOP	The path to the Applications Utilities directory.
<PROD>_TOP	The path to a product's top directory. There is one entry for each Oracle Applications product.
PATH	Sets the directory search path, primarily FND_TOP and AD_TOP.
APPLDCP	Specifies whether distributed concurrent processing is being used. Distributed concurrent processing distributes processing load across multiple concurrent processing nodes.
APPCPNAM	Indicates whether the format of the concurrent manager log and output files follow 8.3 file name conventions (maximum of 8 characters to the left of the dot and 3 to the right, for example, alogfile.log). If this parameter is set to "REQID", the concurrent manager uses file names that meet 8.3 naming requirements.
APPLCSF	Identifies the top-level directory for concurrent manager log and output files if they are consolidated into a single directory across all products. For example, if the path is <COMN_TOP>/admin, all log and output files are placed in a subdirectory under this defined directory.
APPLLOG	The subdirectory for concurrent manager log files. The default is <i>log</i> .
APPLOUT	The subdirectory for concurrent manager output files. The default is <i>out</i> .
APPLTMP	Identifies the directory for Oracle Applications temporary files. The default is <i>/tmp</i> on UNIX and <i>C:\temp</i> on Windows.
APPLPTMP	Identifies the directory for temporary PL/SQL output files. The possible directory options must be listed in the init.ora parameter <i>utl_file_dir</i> .
NLS_LANG	The language, territory, and character set installed in the database. The default for a fresh install is "AMERICAN_AMERICA.US7ASCII".
NLS_DATE_FORMAT	The National Language Support date format. The default is "DD-MON-RR", e.g. 18-MAY-04.
NLS_NUMERIC_CHARACTERS	The National Language Support numeric separators. The default is ".,".

Parameter	Description
FORMS60_MAPPING	Specifies the Oracle Applications forms Web server host name and the port on which it is running. For example: <code>http://server1.surby.com:8000/OA_TEMP</code>
REPORTS60_TMP	The directory for temporary files used by Oracle Reports.
GRAPHICS60_PATH	Specifies the path to the Oracle Graphics files.

Most temporary files are written to the location specified by the APPLTMP environment setting, which is set by Rapid Install. If you so specify, Oracle Reports temporary files can be directed to a different location, determined by the REPORTS60_TMP environment setting.

Applications also produces temporary PL/SQL output files used in concurrent processing. These files are written to a location on the database server node specified by the APPLPTMP environment setting. The APPLPTMP directory must be the same directory as specified by the `utl_file_dir` parameter in your database initialization file.

Rapid Install sets both APPLPTMP and the `utl_file_dir` parameter to the same default directory.

Some Oracle Applications utilities use your operating system's default temporary directory even if you define the environment settings listed in the previous paragraph. You should therefore ensure there is adequate free disk space in this directory, as well as in those denoted by APPLTMP, REPORTS60_TMP, and APPLPTMP. On a multi-node system, the directory defined by APPLPTMP does not need to exist on the application tier servers.

Note: The temporary files placed in the `utl_file_dir` directory can be secured against unauthorized access by ensuring that this directory has read and write access for the Oracle database account only.

Other Environment Files

Several other key environment files are used in an Oracle Applications system.

The `adovars.env` file

The `adovars.env` file, located in `$APPL_TOP/admin`, specifies the location of various files such as Java files, HTML files, and JRE (Java Runtime Environment) files. It is called from the main applications environment file, `<CONTEXT_NAME>.env`. The `adovars.env` file includes comments on the purpose and recommended setting of each variable. In a Release 11.5.10 environment, `adovars.env` is maintained by AutoConfig and should not be edited manually, except in specific cases where customizations have to be used.

The `adovars.env` file includes the following parameters:

Table 2-4 Parameters Specified in the *adovars.env* File

Parameter	Description
JAVA_TOP	Indicates the top-level directory to which all Java files are copied.
OA_JRE_TOP	Indicates the location where JRE is installed.
OA_HTML_TOP	Defines the location to which HTML files are copied.
OAD_TOP	Defines the locations to which context-sensitive documentation files are copied.
LD_LIBRARY_PATH	Used on many UNIX platforms to list the directories scanned for dynamic library files needed at runtime.
CLASSPATH	Lists the directories and zip files scanned for Java class files needed at runtime.

The *adconfig.txt* file

AD utility programs perform a variety of database and file management tasks. These utilities need to know certain configuration information to run successfully. This configuration information is specified when Oracle Applications is installed, and subsequently stored in the *adconfig.txt* file in the <APPL_TOP>/admin directory. Once it has been created, this file is used by other Oracle Applications utilities.

Note: *adconfig.txt* is created with the APPL_TOP file system, and it shows the tiers that have been configured on a particular node. It is distinct from the *config.txt* file created by Rapid Install.

The *fndenv.env* file

This file sets additional environment variables used by the Application Object Library. For example, it sets APPLBIN as the name of the subdirectory where product executable programs and shell scripts are stored (*bin*). This file should not be modified; the default values are applicable for all customers. The file is located in the FND_TOP directory.

The *devenv.env* file

This file sets variables that let you link third-party software and your own custom-developed applications with Oracle Applications. This script is located in FND_TOP, and is automatically called by *fndenv.env*. This allows you to compile and link custom Oracle Forms and Reports user exits and concurrent programs with Oracle Applications.

Applications Database Organization

Introduction

This chapter describes the Oracle Applications data model. It includes a discussion of the basic Applications data model and the corresponding Oracle database server requirements.

A given Oracle database can store the objects associated with a single installation of the Oracle E-Business Suite. Generally, product *data* objects are stored in the associated product schemas, whereas product *code* objects are all stored in the APPS schema.

Oracle User IDs

Each Oracle E-Business Suite product has a default Oracle user ID, with the product abbreviation as both the schema name and password. For example, the default Oracle user ID/password for Oracle General Ledger is GL/GL.

Important: As a security precaution, you should change the passwords immediately after installation. Oracle recommends that you do not change the default user IDs.

A product's schema determines the ownership of the product's data objects, such as sequences, tables, and indexes. If two products are installed under the same schema, that schema owns the data objects for both products.

Since a product's data objects are created in their own schema (such as the GL schema) but the user accesses all data objects through the APPS schema, appropriate grants and synonyms are required between the APPS schema and the base product schemas.

APPS Schema

The APPS schema has access to the complete Oracle E-Business Suite data model. It is analogous to the SYSTEM schema, which has access to the entire database. Oracle Applications responsibilities connect to an APPS schema, and the environment variable FNDNAM is set to the name of the APPS schema.

APPS and Base Product Schemas

All data objects for a product are owned by a schema for that product, known as a *base product schema*. One schema, called *APPS*, owns all the code objects for the Oracle E-Business Suite, and has access to all data objects. There is one *APPS* schema for every product installation group.

The following code objects are installed in the APPS schema:

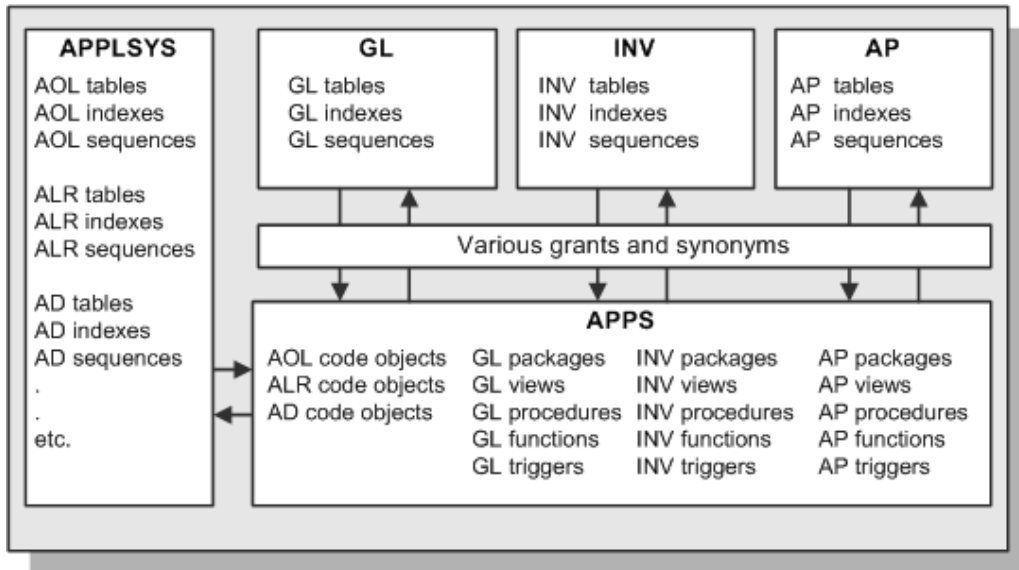
- Packages
- Procedures
- Functions
- Triggers
- Views
- Materialized views
- Java classes
- Queues

The following objects are installed in the base product schemas:

- Tables
- Sequences
- Indexes
- Constraints
- Queues

In addition, there are grants from the tables and sequences to the APPS schema, as well as synonyms from the APPS schema to those objects.

Figure 3-1 APPS Schema and Base Product Schemas



APPS Schema Benefits

The APPS schema improves the reliability of and reduces the time needed for installation, upgrade, and patching by eliminating the need for cross-product grants and synonyms. Utilizing a single schema that has access to all objects eliminates cross-product dependencies, and creates a hub-and-spokes access model rather than a spider's web.

Custom Schema Access

In some circumstances, you may wish to create a schema that has limited or read-only access to Oracle Applications data.

Warning: Since the APPS schema has all privileges to all Oracle Applications objects, you should never give users direct access to this schema.

You will need to grant access on objects to the user schema from the base product schema.

Note: You may need to re-grant access if the underlying object is dropped and recreated.

Data Access

Some views access packages or functions, where the value returned by the package or function may depend on the environment having been set up properly. The environment is initialized automatically when accessing Oracle Applications through the Sign-On screen, or when using concurrent processing with Oracle Reports or SQL scripts.

If you connect directly to a schema, the rows returned by the view may therefore be different from those returned if you are running in an Oracle Applications environment. For example, a view may reference profile options. Consequently, when accessed from SQL*Plus, the site value of the profile option will be used, rather than the setting for a particular Applications user.

Space Management

This section discusses how the Oracle database is set up to meet the space management needs of Oracle Applications. It provides information on tablespaces, firstly outlining the basic tablespaces required, then discussing the traditional tablespace structure used to support Applications products, and finally describing the new tablespace model introduced as standard with Oracle Applications Release 11.5.10 (but also optionally available for use with earlier releases).

Note: Oracle Applications Release 11*i* requires an Oracle database block size of 8K. No other block size may be used.

Introduction to Tablespaces

The Oracle9*i* database server always requires the following tablespaces to be available:

- **System Tablespace** - This tablespace holds data dictionary tables owned by the SYS account, and is created when the database is installed.
- **Undo Tablespace** - This tablespace holds undo (rollback) information that is used to track database changes until they are either committed or undone (rolled back).
- **Temporary Tablespace** - Temporary tablespaces are used to sort data while it is being processed. It is possible to use a single temporary tablespace, typically called TEMP, for all Oracle Applications products. Alternatively, separate temporary tablespaces can, if desired, be created for individual products. Since users access Applications objects through the APPS schema, the temporary tablespace for that schema (initially the same as that for the Oracle Application Object Library) is used by all products.

The traditional Oracle Applications tablespace model employed separate tablespaces for a product's tables and indexes. The resulting tablespaces were named by appending 'D' for data or 'X' for an index to the product's short name or Oracle schema name. For example, APD was the tablespace for Oracle Payables data, and APX was the tablespaces for Oracle Payables indexes.

Note: For further information about tablespaces, see the *Oracle9i Database Administrator's Guide*.

Employing separate table and index tablespaces for each product made it easier maintain products, and helped to improve database performance. However, with an increasing number of products, this model could easily require several hundred product tablespaces, plus a system tablespace, undo (rollback) tablespace, and temporary tablespace.

In addition, the traditional tablespace model used a database *sizing factor* to set the extent sizes for an Oracle Applications product's tables and indexes. The value of this factor was a percentage of the typical estimated growth rate for Applications database objects. The sizing factor affected only the size of subsequent extents, as determined by the NEXT database object creation parameter. Most objects were defined with small first extents and larger additional extents.

During installation, Rapid Install provides the option of distributing tablespaces across different disks, to reduce disk head contention and improve overall system performance. In addition to this, many production systems utilize sophisticated disk and volume management technologies at operating system level to further enhance performance.

Note: For further details, see *Upgrading Oracle Applications*.

Oracle Applications Tablespace Model

Oracle Applications Release 11.5.10 introduces a new infrastructure for tablespace management, the *Oracle Applications Tablespace Model* (OATM). The Oracle Applications Tablespace Model is similar to the traditional model in retaining the system, undo, and temporary tablespaces. The key difference is that Applications products in an OATM environment share a much smaller number of tablespaces, rather than having their own dedicated tablespaces.

Note: The information in this section also applies to earlier releases of Applications that have been migrated to the Oracle Applications Tablespace Model. See OracleMetaLink Note 248857.1 for details.

Applications schema objects are allocated to the shared tablespaces based on two main factors: the type of data they contain, and I/O characteristics such as size, life span, access methods, and locking granularity. For example, tables that contain *seed data* are allocated to a different tablespace from the tables that contain *transactional data*. In addition, while most indexes are held in the same tablespace as the base table, indexes on transaction tables are held in a single tablespace dedicated to such indexes.

The Oracle Applications Tablespace Model provides a variety of benefits, summarized in the list below and discussed in more detail later:

- Simplifies maintenance and recovery by using far fewer tablespaces than the older model.

- Makes best use of the restricted number of raw devices available in Real Application Cluster (RAC) and other environments, where every tablespace requires its own raw device.
- Utilizes locally managed tablespaces, enabling more precise control over unused space and hence reducing fragmentation.
- Takes advantage of automatic segment space management, eliminating the need for manual space management tasks.
- Increases block-packing compared to the older model, reducing the overall number of buffer gets and improving runtime performance.
- Maximizes usefulness of wide disk stripe configurations.

The Oracle Applications Tablespace Model uses *locally managed tablespaces*, which enables extent sizes either to be determined automatically (*autoallocate*), or for all extents to be made the same, user-specified size (*uniform*). This choice of extent management types means that locally managed tablespaces offer greater flexibility than the *dictionary-managed tablespaces* used in the traditional tablespace model. However, when using uniform extents with locally managed tablespaces, the extent size must be chosen with care: too small a size can have an adverse effect on space management and performance.

A further benefit of locally managed tablespaces, and hence use of OATM, is the introduction of *automatic segment space management*, a simpler and more efficient way of managing space within a segment. It can require more space, but eliminates the need for traditional manual segment space management tasks such as specifying and tuning schema object storage parameters such as PCTUSED. This and related storage parameters are only used to determine space allocation for objects in dictionary-managed tablespaces, and have no meaning in the context of locally managed tablespaces.

Automatic segment space management is self-tuning, so can take into account an increase in the number of users. A further benefit in Real Application Cluster (RAC) environments is *dynamic affinity of space to instances*, which avoids the hard partitioning of space inherent with the traditional use of free list groups.

Table 3-1 OATM Tablespace Types and Contents

Tablespace Type	Tablespace Contents
Transaction Tables	Tables that contain transactional data.
Transaction Indexes	Indexes on transaction tables.
Reference	Reference and setup data and indexes.
Interface	Interface and temporary data and indexes.
Summary	Summary management objects, such as materialized views, and other objects that record summary information.
Nologging	Materialized views that are not used for summary management and temporary objects.
Advanced Queuing	Advanced Queuing (AQ) tables and indexes.
Media	Multimedia objects, such as text, video, sound, graphics, and spatial data.
Archive	Tables that contain archived purge-related data.
Undo	Automatic Undo Management (AUM) tablespace. Undo segments are equivalent to rollback segments when AUM is enabled. See note below.
Temp	Temporary tablespace for global temporary table, sorts, and hash joins.
System	System tablespace used by the Oracle database.

In Oracle database server releases prior to Oracle9i, undo space management was performed using rollback segments. For clarity, this method is now referred to as *manual undo management*. Its successor, *automatic undo management* is based on the use of a small number of *undo tablespaces*, in contrast to the larger number of variously-sized rollback segments typically used in manual undo management.

Oracle Applications Tablespace Migration Utility

Conversion of pre-11.5.10 Applications releases to the Oracle Applications Tablespace Model is performed as a standalone operation. The *OATM Migration Utility* migrates objects from the existing dictionary-managed tablespaces to locally managed tablespaces that (as mentioned earlier) use automatic segment space management to optimize space utilization. During the process of migrating data and index segments from the old tablespaces to OATM, the Migration Utility reclaims any unused space. This is especially valuable where fragmentation has occurred, for example in the case of indexes where columns are frequently inserted, updated, or deleted.

The Migration Utility is an interactive, menu-based Perl program that allows for migration of either all schemas at once, or for one or more selected schemas at a time, depending on acceptable downtime and available disk space for doing the conversion.

Note: It is more efficient to perform a single comprehensive migration of all schemas, rather than a phased schema-by-schema migration.

The OATM Migration Utility's actions fall into the following functional areas:

- Produces reports in advance of the actual migration to help estimate database sizing requirements
- Carries out the migration to the new tablespace structure
- Writes log files used to verify success of the migration

To minimize production downtime, one or more test runs should be carried out to determine the amount of downtime likely to be required.

The Tablespace Migration Utility main menu shows the six required sequential steps used to migrate a database's objects to OATM. These steps can be divided into three phases.

1. Preparatory steps for migrating the database objects to OATM
 1. Generate Migration Sizing Reports
 2. Create New Tablespaces
 3. Generate Migration Commands
2. Migration of the database objects to OATM
 1. Execute Migration Commands
 2. Run Migration Status Reports
3. Post-migration steps
 1. Run Post Migration Steps

Step 2.1 must be executed without any users logged on to Oracle Applications. Steps 2.2 and 3.1 must be completed before making Oracle Applications available to users again.

The Migration Utility has a final menu item, Run Customization Steps, which can optionally be used to customize tablespaces, tablespace types, and object classifications. Custom database objects (such as tables, indexes, materialized views, and materialized view logs) that have been created in standard Oracle Applications schemas (such as GL) will be moved either to their default tablespace, or to the tablespace type determined by the rules for the particular object type.

Custom database objects created in custom schemas registered with Oracle Applications are not affected: if they reside in standard Oracle Applications tablespaces, they will be ignored during the migration, and no errors will be reported. Possible examples include database objects in other Oracle product schemas (such as Oracle Portal or Oracle Discoverer), or in third-party or custom schemas.

Note: Any customization steps should be run before the other migration steps are started.

The Oracle Tablespace Migration Utility supports the following customization tasks:

- Registering custom tablespace types, for cases where the required type is not supplied by default with the Oracle Applications Tablespace Model.
- Changing the names of any default OATM tablespaces or registered custom tablespaces.
- Registering custom objects with their corresponding tablespace classifications (relevant only for objects that require explicit classification, such as tables).

- Changing any existing object-tablespace classifications.

Warning: An object can only be moved back to the old tablespace model by restoring from a database backup, or using a suitable SQL statement (or the equivalent function in Oracle Enterprise Manager).

The Migration Utility is primarily designed to enable the use of OATM with database objects such as tables, indexes, and materialized views stored in standard Oracle Applications product schemas. The Migration Utility does not handle custom or third party application schemas by default: however, they can be handled by it if they are registered with Oracle Applications, with suitable access privileges.

Note: See OracleMetaLink Note 269291.1 for further details of OATM and the Migration Utility.

Database Features and Release 11i

Introduction

Many features in Oracle E-Business Suite Release 11i are built on the advanced capabilities of the underlying Oracle database technology. Release 11i utilizes various Oracle database features to optimize performance, scalability, and business intelligence capacity.

Performance Features

Database performance features include optimization, resource usage, space management, and access rights.

Cost-Based Optimization

The Oracle optimizer evaluates many factors to calculate the most efficient way to execute a SQL statement. It is capable of using either a *rule-based* or *cost-based* approach for execution of the statement. Rule-based optimization was used in earlier Applications releases, but since the SQL used in Release 11i has been extensively tuned for cost-based optimization, Release 11i requires the optimizer to use cost-based optimization (CBO).

Using CBO, the optimizer considers the available access paths, factoring in statistical information for the tables and indexes that the SQL statement will access. CBO also considers *hints*, which are optimization suggestions placed in a comment of the SQL statement.

First, the optimizer creates a set of potential execution plans for the SQL statement, based on its available access paths and hints. The optimizer then estimates the *cost* of each execution plan, based on data dictionary statistics for the data distribution and storage characteristics of the tables, indexes, and partitions. The optimizer compares the costs of the execution plans and chooses the one with the smallest cost, i.e. optimum execution characteristics.

For some operations, such as batch processing, Release 11i uses CBO to achieve the most efficient means of processing *all rows* that are accessed by the statement. For other operations, such as accessing forms or communication with the desktop client, Release 11i uses CBO to achieve the best response time for processing the *first rows* that are accessed by the statement.

Several other Oracle database performance enhancements used in Release 11i, such as partitioned tables, also require use of the cost-based optimizer.

Note: For further details, see: The Optimizer Overview, *Oracle9i Concepts*; Introduction to the Optimizer, *Oracle9i Database Performance*

Database Resource Manager

The gives the system administrator extensive control over processing resources on the database node. The administrator can distribute server CPU based on business rules, ensuring that the highest priority activities always have sufficient CPU resources. The administrator could, for example, guarantee Order Entry users 40% of CPU resources during business hours, regardless of the load or number of users in other groups on the system.

System administrators can also use the Database Resource Manager to limit the impact of any inefficient ad hoc queries. For example, a limit of 5% of CPU resources could be placed on ad hoc queries against the database.

Partitioned Tables

Partitioning helps support very large tables and indexes by dividing them into smaller, more manageable pieces called *partitions*. Once the desired partitions have been defined, SQL statements can access them instead of the original tables or indexes.

Note: Custom partitioning of standard Applications tables in Release 11i is fully supported.

Partitioning reduces access time, and partitions are especially useful in data warehouse applications, which often store and analyze large amounts of historical data. For example, operations that involve copying or deleting data are improved by use of partitioned tables. Creating and deleting all rows of a partitioned table is a much faster operation than selectively inserting rows into an existing table, then selectively deleting rows from the table. Some operations that might have taken hours can now be completed in seconds.

Most Applications tables do not have a natural partitioning key that would apply to all installations, because of differences in data distribution and access paths in different implementations. Tables should therefore be partitioned in a logical manner, to meet your specific requirements. For example, *period_name* and *set_of_books_id* are likely candidates for partitioning the GL_BALANCES table.

Important: Custom partitioning should be planned carefully. After it has been implemented, you should test that the desired performance benefits have been achieved; it is possible for performance to be degraded if partitioning is not planned properly.

Invoker Rights

In earlier releases, if you used Multiple Reporting Currencies (MRC) or Multiple Sets of Books Architecture (MSOBA), several copies of some Oracle Applications packages needed to exist in the database. As well as taking more space, this required extra time and effort to maintain.

Applications Release 11i uses the *Invoker Rights* functionality of the Oracle database server to ensure that in most cases, packages need be installed in the APPS schema only. PL/SQL routines use Invoker Rights to access packages in the APPS schema.

Note: There have been significant changes to the MRC architecture in recent releases of Oracle Applications, and the MRC schema no longer exists in Release 11.5.10. See Chapter 15 for more details.

Business Intelligence Features

To meet the increasing demand for up-to-date details of business activities, Oracle Applications utilizes Oracle database features that help to optimize the types of query typically required in such environments.

Materialized Views

Materialized views are schema objects that can be used to summarize, precompute, replicate, and distribute data. They can markedly increase the speed of queries on very large databases when used to precompute and store aggregated data such as sums and averages. Materialized views can therefore improve performance of Oracle Applications products, such as Daily Business Intelligence, that perform many queries on summary data.

Cost-based optimization can use materialized views to improve query performance by automatically recognizing when one can be used to satisfy a request. The optimizer transparently rewrites the request to use the materialized view. Queries are then directed to the materialized view, and not to the underlying detail tables or views.

In distributed environments, materialized views can be used to replicate data at remote sites, providing local access to data that would otherwise have to be accessed from the main site, with any network delays this might introduce.

Scalability

As well as providing more computing power, multi-node systems facilitate the addition of machines to meet increases in demand. They also provide resilience in the event of failures of individual components.

Real Application Clusters

Real Application Clusters (RAC) harness the processing power of multiple interconnected computers. RAC software and a collection of computers (known as a *cluster*) harness the processing power of each component to create a robust and powerful computing environment. A large task divided into subtasks and distributed among multiple nodes is completed more quickly and efficiently than if the entire task was processed on one node. Cluster processing also facilitates deployment of additional hardware resources for larger workloads and rapidly growing user populations.

In RAC environments, all active instances can concurrently execute transactions against a shared database. RAC coordinates each instance's access to the shared data, to provide data consistency and data integrity. From a developer's point of view, RAC enables applications to be scaled to meet increasing data processing demands, without the need to change the application code.

All E-Business Suite modules can be successfully deployed against a RAC-enabled Oracle database. Using Parallel Concurrent Processing (see Chapter 1), concurrent managers on separate application tier machines can be configured to direct requests to different database servers in a RAC cluster.

Note: The AutoConfig configuration management tool (described in Chapter 5) supports RAC environments from Release 11.5.10 onwards.

Introduction

Configuring a new installation of Oracle Applications includes a number of stages:

- Collection of information required to create the desired configuration
- Storage of configuration information in the correct locations on the relevant machines
- Creation of technology stack configuration files with the appropriate details
- Creation of Applications configuration files with the appropriate details
- Starting of all required processes in the correct order

AutoConfig is a tool that simplifies and standardizes configuration management tasks in an Oracle Applications environment. A fresh installation of Release 11.5.10 includes AutoConfig as a standard (and required) configuration management tool. AutoConfig can also be used with earlier releases of Oracle Applications.

The Applications Context

Where the *System Identifier* (SID) has traditionally been used to identify a file as belonging to a particular Oracle Applications environment, an *Applications context* is used in an AutoConfig-managed Applications environment such as 11.5.10. The default *context name*, also referred to in this chapter as `<CONTEXT_NAME>`, is `<SID>_<hostname>`.

Using an Applications context has a number of advantages:

- Simplifies overall management of an Applications system
- Allows easier startup and shutdown of Applications services
- Permits services to be installed or deinstalled independently of others
- Integrates seamlessly with a shared APPL_TOP environment (see Chapter 9)
- Enables use of Oracle Applications Manager for configuration management (see Chapter 7)
- Facilitates support for Real Application Clusters (see Chapter 4)

Before the Applications context and AutoConfig were introduced, configuration management tasks could be time-consuming and prone to error, in some cases requiring manual changes to be made to several configuration files. While individual configuration files are still used in an AutoConfig-enabled environment, they play a secondary role to an XML-based repository of Applications environment information, called the *context file*.

By centralizing the configuration information, AutoConfig simplifies procedures for activities that range from upgrading a technology stack component to starting and stopping Applications services. Another benefit is that the various files AutoConfig employs can be updated via standard Applications patches.

There are separate context files for the application and database tiers of an Applications system.

Applications Context File - The Applications context file, `APPL_TOP/admin/<CONTEXT_NAME>.xml`, is a repository for environment-specific details used by AutoConfig to configure the application tier. Information from this file is used to generate Applications configuration files and update relevant database profiles.

Information stored includes:

- Name and location of the database
- Port numbers for Forms and Web servers
- Product-specific port numbers
- Information about application tier services controlled by AutoConfig

The values of the *context variables* that make up the context file are in part determined by the choices you make when you run Rapid Install. For example, when you specify that a particular application tier node is to be used as a concurrent processing and Reports server, the relevant variable will be set to create the necessary service scripts.

Database Context File - The database context file, `<RDBMS_ORACLE_HOME>/appsutil/<CONTEXT_NAME>.xml`, performs an equivalent role on the database tier. Information from this file is used to generate configuration files used on the database tier when AutoConfig is next run.

AutoConfig Scripts and Directories

A number of key configuration and control scripts are employed in an AutoConfig environment. AutoConfig creates several directories for these scripts and their associated files.

AutoConfig Scripts

Key AutoConfig configuration scripts (command files on Windows) include:

- **adautoconfig.sh** - Wrapper script that passes the name of the specific environment context file to `adconfig.sh`.
- **adconfig.sh** - Invoked by `adautoconfig.sh`, this script is a wrapper for `adconfig.pl`.
- **adconfig.pl** - Invoked by `adconfig.sh`, this Perl script calls the Java API to carry out the actual configuration tasks. The relevant Java code is located in the `<JAVA_TOP>` directory, either `<COMMON_TOP>/java` (on the application tier) or `RDBMS_ORACLE_HOME/appsutil/java` (on the database tier).

In addition to these configuration scripts, several additional AutoConfig scripts are used to help manage the Applications system. These are described later in this chapter, under *Management Tasks*.

AutoConfig Directories

Several directories are created by AutoConfig, as shown in the following table.

Table 5-1 AutoConfig Directories

Directory Name	Directory Contents
<COMMON_TOP>/admin/install/<CONTEXT_NAME>	Install scripts
<COMMON_TOP>/admin/scripts/<CONTEXT_NAME>	Control scripts
<COMMON_TOP>/admin/log/<CONTEXT_NAME>	Log files

AutoConfig Operation

As AutoConfig is used for a wide range of system configuration activities, from installation to maintenance, the following discussion of its operations is divided into several sections.

Context Value Management

Context Value Management (CVM) is an AutoConfig component that is used to manage the values of variables in the context file, and automate required updates to it. CVM supports updates to both the application tier and database tier context files.

CVM actions include:

- Adding new variables to a context file.
- Updating values of variables in an existing context file.
- Applying new versions of context file templates.
- Executing scripts or configuration tools that must complete before the AutoConfig engine starts, for example when generating the tnsnames.ora file.

CVM is activated when the Applications context file is updated, but before the AutoConfig engine itself starts. This enables CVM to execute scripts or other tools to manipulate any required file on the file system, and allow the appropriate settings to be propagated as needed to both the file system and database. For example, it is possible to update values in the context file which will then be propagated to the file system.

Note: AutoConfig does not undertake all aspects of configuration management, such as operations at operating system level that may have implications outside the context of Oracle E-Business Suite.

Like the core AutoConfig components, CVM utilizes configuration files on both the application and database tiers, as shown in the following tables.

Table 5-2 Application Tier CVM Files

File Location	Description
<APPL_TOP>/bin/adcvms.sh	Main CVM script
<APPL_TOP>/admin/template/adcvmat.xml	Stores CVM-related data for the application tier

Table 5-3 Database Tier CVM Files

File Location	Description
<RDBMS_ORACLE_HOME>/appsutil/bin/adcvms.sh	Main CVM script
<RDBMS_ORACLE_HOME>/appsutil/template/adcvmdb.xml	Stores CVM-related data for the database tier

AutoConfig Files

As well as the context files and configuration scripts that have already been described, AutoConfig uses several other types of file in its configuration management activities. These may themselves be divided into different categories.

Template Files

AutoConfig template files are used as the starting point for creating site-specific configuration files. AutoConfig evaluates the context variables in a template file, determines the actual values required, and creates a configuration file with these values substituted. This process, described in more detail later in this chapter, is called *instantiation*. There is one template file for each configuration file. Template files are located in the various <PROD>_TOP/admin/template directories on the application tier, and in the <RDBMS_ORACLE_HOME>/appsutil/template directory on the database tier.

Template files used by AutoConfig can be divided into the following categories:

Templates for APPL_TOP Configuration Files - These are either files requiring configuration-specific information in the APPL_TOP, or files used to load configuration profiles into the Applications database.

Templates for Management Scripts - To run all the standard processes required by Applications, Rapid Install creates scripts to start and stop each of these required processes. These scripts need configuration information in order to:

- Create the correct environments for each process
- Start the processes with the correct parameters
- Point the processes at the correct database instance (if applicable)

Driver Files

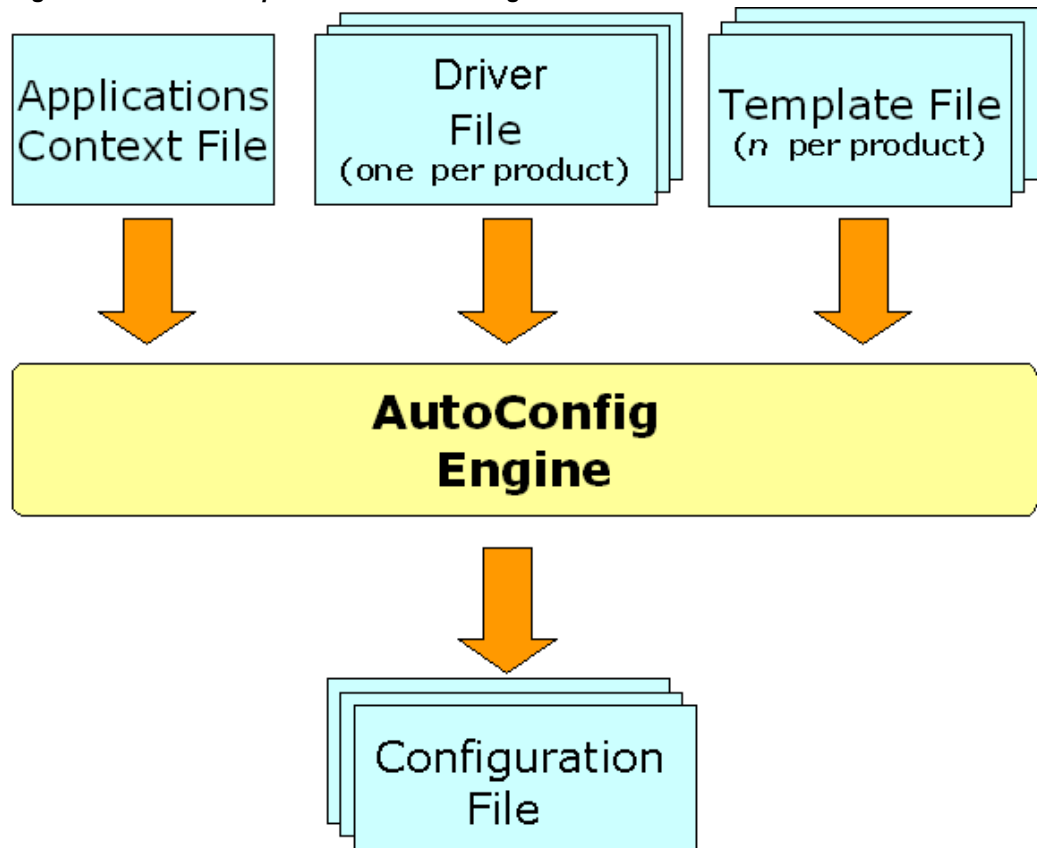
AutoConfig driver files are used to list the corresponding template files and locations, and specify the commands to be executed. For example, the commands might update profile options.

Driver files are located in each <PROD>_TOP/admin/driver directory on the application tier, and in the <RDBMS_ORACLE_HOME>/appsutil/template directory on the database tier.

Configuration Files

AutoConfig configuration files, such as *httpd.conf*, are created as a result of AutoConfig instantiating the corresponding template files. Configuration files contain values corresponding to the settings specified for a particular site. After AutoConfig has been run, numerous configuration files will have been created in various directories.

Figure 5-1 Relationship Between AutoConfig Files



Instantiation

As mentioned earlier, instantiation is the process whereby AutoConfig creates a configuration file with contents tailored for a specific environment. AutoConfig can be used to instantiate files or scripts, and then execute them for installation and configuration.

Examples of instantiation include:

- Instantiation of a configuration file to be used at runtime
- Instantiation of an SQL script to set profile options
- Instantiation of a shell script or Windows command file to run an SQL script in SQL*Plus
- Instantiation of scripts to start up and shut down application tier services

The *adautocfg.sh* script updates configuration files and profile options in the following way:

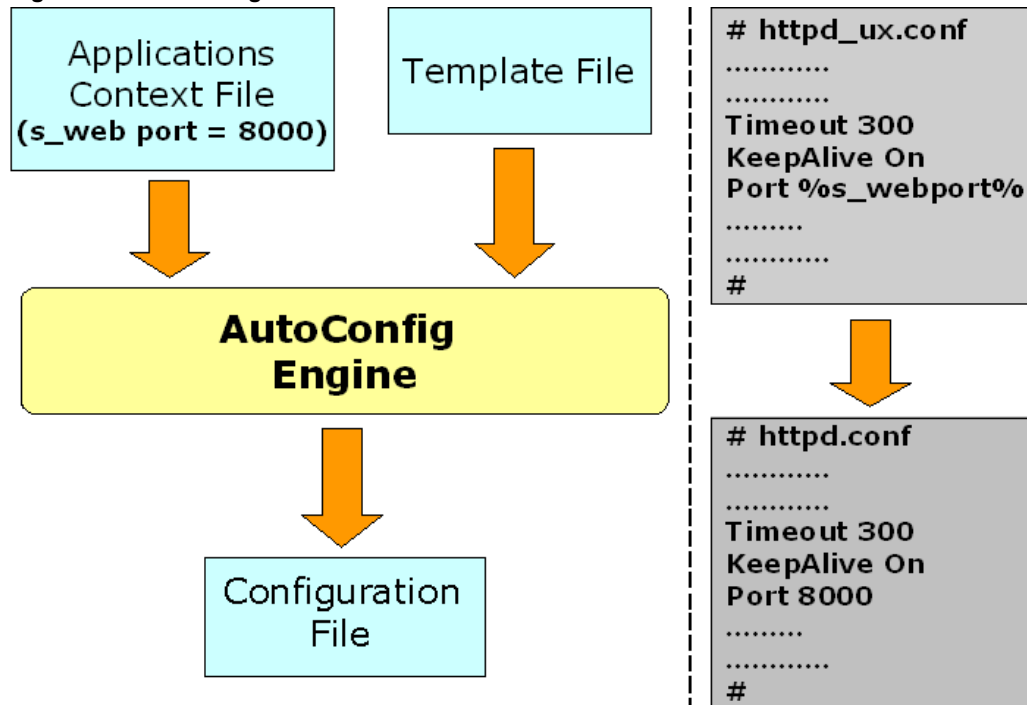
1. Instantiates template files with instance-specific values derived from the relevant context file
2. Copies in any customizations
3. Overwrites existing configuration files with newly instantiated ones
4. Runs SQL scripts to update database profile options

Role of the template and driver files

AutoConfig uses the various *template* files to determine the basic settings needed. There is one template file for each configuration file. Different versions of the template files exist for UNIX and Windows. Examples include `httpd_ux.conf` (UNIX) and `httpd_nt.conf` (Windows), and `adfrmctl_ux.sh` (UNIX) and `adfrmctl_nt.cmd` (Windows).

The *driver* files list the names and locations of the files that need to have context variables replaced. They also define the phases into which instantiation is divided, and specify the commands that are to be executed for specific products. When AutoConfig runs, it cycles through the various `<PROD>_TOP/admin/driver` directories looking for driver files such as `adtmpl.drv`, `fndtmpl.drv`, and `icxtmpl.drv`.

Figure 5-2 AutoConfig Instantiation Mechanism



Execution of Scripts

As well as its instantiation activities, AutoConfig carries out numerous other essential configuration management tasks, by executing scripts such as the following.

Table 5-4 Examples of AutoConfig Setup Scripts

Script	Action
<code>adgendbc.sh</code>	Generates the dbc file
<code>adgenjky.sh</code>	Generates JInitiator security information
<code>adcpnode.sh</code>	Registers nodes in the database
<code>ssodatan.sh</code>	Associates Portal with Oracle Single Sign-On

These and other scripts are executed as applicable, depending on the requirements of the specific Applications system.

Note: For further details of AutoConfig operation, see OracleMetaLink Note 165195.1, *Using AutoConfig to Manage System Configurations with Oracle Applications 11i*.

Phases of Operation

As AutoConfig parses the driver files, it carries out a series of actions, grouped into several distinct phases:

- **INSTE8** - Instantiates AutoConfig template files to the AutoConfig configuration files specified in the relevant template driver files.
- **INSTE8_SETUP** - Executes setup scripts that carry out activities not involving connection to the database.
- **INSTE8_PRF** - Executes setup scripts that update profile options.
- **INSTE8_APPLY** - Executes setup scripts that carry out activities involving updates to the database.
- **BINCPY** - Copies the file mentioned from the source file to the configuration file, creating parent directories for the latter if necessary. AutoConfig will report an error if the source file cannot be found.
- **BINCPY_IGERR** - Copies the file mentioned from the source file to the configuration file, creating parent directories for the latter if necessary. AutoConfig will *not* report an error if the source file cannot be found.

AutoConfig carries out these actions in the following order:

1. **All INSTE8 and BINCPY actions** - Carries out all file instantiations called for during INSTE8, INSTE8_SETUP, INSTE8_PRF and INSTE8_APPLY, and all copying from source files to target configuration files.
2. **INSTE8_SETUP actions** - For the files that were instantiated in Step 1, AutoConfig runs all SETUP scripts.
3. **INSTE8_PRF actions** - For the files that were instantiated in Step 1, AutoConfig runs all PRF scripts.
4. **INSTE8_APPLY actions** - For the files that were instantiated in Step 1, AutoConfig runs all APPLY scripts.

At the end of this process, the required configuration files and profile options have been created for the E-Business Suite installation.

Management Tasks

There are several areas in which an administrator can use AutoConfig to update, control, and monitor an Applications system. While many tasks will normally be undertaken from Oracle Applications Manager, you may on occasion need to run a script (command file on Windows) from the command line.

Managing the Context

Oracle Applications Manager (see Chapter 7) enables you to edit the Applications context as required. From the Administration tab, choose AutoConfig and click on *Edit*

Parameters for the relevant context file. After making a change to the context, you must run AutoConfig to update the relevant configuration files. Before doing so, you should examine the proposed changes by running the *adchkcfg.sh* configuration check script (described below under *Checking the System*).

Warning: Do not edit E-Business Suite configuration files manually. Any changes will be lost when AutoConfig is next run.

An AutoConfig-managed environment such as 11.5.10 will also be updated by applying an Applications patch that adds or modifies variables in the Applications context file, AutoConfig template files, or AutoConfig driver files.

It may on occasion be necessary to undo configuration changes. You can restore the previous configuration by running the *restore.sh* utility, which enables you to roll back the changes made by an AutoConfig run. This is achieved by utilizing the backup copies of the configuration files that are created when AutoConfig is run.

Note: The backup files are located in <APPL_TOP>/admin/<CONTEXT_NAME>/out/MMDDhhmm on the application tier, and <RDBMS_ORACLE_HOME>/appsutil/out/MMDDhhmm on the database tier, where the directory name indicates the month, day, hour and minute of the AutoConfig run.

You can restore the configuration that existed immediately before the current one by navigating to the appropriate backup directory and running the *restore.sh* script. To restore an earlier configuration, you must use the *Context File History* feature of Oracle Applications Manager.

Controlling the System

AutoConfig utilizes a number of application tier control scripts, located in <COMMON_TOP>/admin/scripts/<CONTEXT_NAME>.

Table 5-5 Application Tier Control Scripts

Script Name	Function
adstrtal.sh	Starts all application tier server processes
adstpall.sh	Stops all application tier server processes
adautocfg.sh	Runs AutoConfig

The corresponding directory on the database tier is <RDBMS_ORACLE_HOME>/appsutil/scripts/<CONTEXT_NAME>, where control scripts allow the database and database listener processes to be started and stopped, and AutoConfig to be run.

Checking the System

Several other scripts exist to facilitate system management in an AutoConfig-managed environment such as 11.5.10. These allow you to identify the effects of proposed changes, and note which products are maintained by AutoConfig.

Examining changes

adchkcfg.sh is located in <AD_TOP>/bin on the application tier, and in <RDBMS_ORACLE_HOME>/appsutil/bin on the database tier.

This utility generates a report that highlights differences between existing configuration files and the new ones that AutoConfig will generate. The report is called *cfgcheck.html*.

Running *adchkcfg.sh* is useful both in carrying out a test run before a planned environment change is made, and when investigating problems.

Listing enabled products

adcfginfo.sh is located in <AD_TOP>/bin on the application tier, and in <RDBMS_ORACLE_HOME>/appsutil/bin on the database tier. This utility reports if an Applications system is AutoConfig-enabled (which will always be the case for 11.5.10). In addition, it can optionally list the installed products that are maintained by AutoConfig.

Note: For further details on managing system configuration parameters, see *Oracle Applications Maintenance Procedures*.

Introduction

Cloning is the act of creating an identical copy of an existing Oracle E-Business Suite system. The system to be cloned is referred to as the *source system*, and the newly created system is referred to as the *target system*.

Cloning has various uses, such as:

- Creating a copy of a production system for patch testing
- Creating a staging area to reduce the downtime required for patching
- Refreshing a test system from a production system
- Moving an existing system to a different machine or platform

Simply copying the existing components to a new location will not provide a working Applications installation. For example, there are numerous configuration files in the file system that must be modified, depending on the physical configuration of the target environment. In addition, the Applications installation process utilizes the *Oracle Universal Installer*, which maintains key information about the installation. Copying the installation to a new location would invalidate this information, preventing the application of patches to components maintained by the Installer.

Cloning Tools

Cloning an Oracle E-Business Suite Release 11*i* system involves running the *Rapid Clone* tool. This can be used with any Oracle E-Business Suite Release 11*i* releases that are AutoConfig-enabled by default, such as 11.5.10, and earlier releases that have been migrated to use AutoConfig (see Chapter 5).

Oracle Applications Manager can also be used to manage cloning, offering either basic or advanced options (see Chapter 7).

Cloning Across Platforms

When cloning from one machine to another, the simplest case is where the two machines are running the same version of the same operating system.

A slightly more complex case occurs where the two operating systems are *binary compatible*, and the *source system* is running an earlier version of the same operating system that is being used on the *target system*. While Rapid Clone can often be used successfully in such cases, you should generally aim to clone between machines that are

running identical versions of an operating system. This minimizes the risk of problems arising because of differences between the versions.

Note: It is not supported to clone from a later version of an operating system to an earlier one.

Cloning can also be used to migrate an Applications system from any platform to any UNIX platform, such as Linux, using the *Platform Migration Utility*.

Note: For further details of migrating, see OracleMetaLink Note 238276.1, *Migrating to Linux with Oracle Applications Release 11i*.

Cloning Options

The following table summarizes the cloning options currently available.

Note: In this context, *node* refers to a logical collection of E-Business Suite processes, and not necessarily a physical machine.

Table 6-1 Cloning Options

	Single node to Single node	Recloning (of database only) ¹	Cloning an existing Clone ²	Multi-node to Multi-node	Single node to Multi-node ³	Multi-node to Single node ⁴
Available with Rapid Clone?	Yes	Yes	Yes	Yes	Yes	Yes

Notes on Table

1. Recloning of the database only can be useful if the source system has changed and the target system needs to be updated with these changes. However, if any Applications patches have been applied to the source system, the APPL_TOP must also be cloned in order to keep the file system and database synchronized.
2. A cloned system created with Rapid Clone can be used as the source system for another round of cloning.
3. If moving to a multi-node system, it is preferable to implement a shared APPL_TOP rather than clone from a single node to multiple nodes. See Chapter 9 for details.
4. This is often referred to as merging APPL_TOPs.

Note: For further details of cloning options, see *Oracle Applications Maintenance Procedures*.

Rapid Clone Operation

Rapid Clone does not modify the source system. The *adpreclone.pl* script prepares the source system to be cloned by collecting information about the database, and creating generic templates from existing files that contain source-specific hard-coded values. The template files are located in <RDBMS_ORACLE_HOME>/appsutil/template on the database tier.

After running *adpreclone.pl*, you copy the relevant files and directories from the source system files to the target system, and then run the *adcfgclone.pl* configuration script. The

values for various parameters are required to create the context file that will be used to configure the target system. A few of these values are calculated from the current target system, and `adcfgclone.pl` will prompt for the others.

For example, you will be prompted to specify a *port pool*, to use a particular range of predefined server ports. There are 100 port pools, so if, for example, you select pool port 3, the default database port number (1521) is replaced by 1524.

Note: If you are cloning to the same machine, you must specify a different port pool from the source system.

If desired, it is possible to set a specific port to a value other than the one assigned from the port pool. This requires editing the context file on the target system after `adcfgclone.pl` completes, then running AutoConfig to update the system with the new value.

Note: For further details of the steps involved in using Rapid Clone, see OracleMetaLink Note 230672.1, *Cloning Oracle Applications Release 11i with Rapid Clone*.

Rapid Clone and Oracle Universal Installer

With Oracle Universal Installer 2.2 (supplied with Oracle Applications Release 11.5.10), the Oracle Universal Installer's *global inventory* is simply a list of pointers to each *local inventory* location. There is one local inventory per ORACLE_HOME, located in `<ORACLE_HOME>/inventory`, which contains all the patch information for the ORACLE_HOME in question.

Rapid Clone first ensures that the source system local inventory is in XML format, converting it from the older binary format if necessary. The local inventory (inside the ORACLE_HOME to be cloned) is then copied to the target system and reconfigured with the new values for the target system. Rapid Clone subsequently attaches the reconfigured local inventory to the target system global inventory. If the target system does not have a global inventory, a new global inventory is created when Rapid Clone goes to attach the local inventory.

New Cloning Features in Release 11.5.10

Oracle E-Business Suite Release 11.5.10 includes a number of new features designed to make cloning more straightforward, and give greater flexibility in response to issues such as:

- Whether cloning is being used to add a node to an existing installation, or to create an entirely new installation. In the former case, there will be fewer ancillary changes.
- Types of table modification that need to take place. For example, when using Rapid Clone to add a node, a new row is inserted into FND_NODES, whereas when creating a new installation, FND_NODES is purged and a completely new set of rows inserted.
- Whether services should be set to start automatically after cloning is complete.
- Whether any data alteration is needed after cloning.

Summary

In essence, Rapid Clone does the following:

- Registers the ORACLE_HOME with Oracle Universal Installer inventory, so the ORACLE_HOME can be patched by the Installer.
- Updates the files in the ORACLE_HOME that have explicitly defined path entries.
- Updates the symbolic links inside the ORACLE_HOME.
- Notifies the user if there are symbolic links outside the ORACLE_HOME that need manual validation or other attention.

In addition, there are a number of associated actions relating to the database.

Oracle Applications Manager

Introduction

Oracle Applications Manager (OAM) is a powerful, easy to use tool that enables you to manage and monitor an Oracle Applications system from an HTML-based central control console. Among other E-Business Suite system management tasks, Oracle Applications Manager can help you to:

- Configure and administer your system
- Diagnose and correct problems
- Manage patches
- Monitor and tune performance
- Monitor system security

Oracle Applications Manager is built directly into the E-Business Suite system, and complements the features of the *Oracle Enterprise Manager* tool.

Oracle Applications Manager Features

The *Applications Dashboard* provides an overview of the key features of an E-Business Suite installation, including summaries of current status, performance, critical activities, diagnostics, business flows, and security. You can then view additional information on any of these areas.

A more detailed view of the Applications system is offered by the *Site Map*, which provides easy access to the numerous features and options that exist in the areas of administration, monitoring, maintenance, and diagnostics and repair.

Administration

Oracle Applications Manager provides a comprehensive system configuration editor, for use in conjunction with the AutoConfig configuration management tool (see Chapter 5). Previous configuration settings can easily be compared with the current settings, allowing changed settings to be identified and rolled back as necessary. The configuration editor also helps you add custom variables to the context files, to allow AutoConfig and AutoPatch to preserve your system configuration customizations.

Oracle Applications Manager also provides advanced *Configuration Wizards* that guide you through formerly complex tasks such as configuring application tier load balancing, or converting to use *Secure Sockets Layer* (SSL) accelerators (see Chapter 10).

The wizards prompt you for all of the required information, and validate your settings before the changes are made to your system.

Additionally, Oracle Applications Manager helps you track configuration changes, such as recently altered site-level profile option settings. It can also help you detect potential configuration problems, such as database initialization parameters that do not meet Oracle requirements or recommendations. Oracle Applications Manager can be used to configure and control critical application tier services such as concurrent processing, Oracle Forms Listeners, Reports servers, and Web servers. For example, you can monitor and administer concurrent requests, and easily analyze the concurrent request workload to make informed configuration changes for concurrent processing.

Oracle Applications Manager can also be used to control Oracle Workflow system services, such as background engines, notification mailers, agent listeners, and queue propagation. You can monitor and analyze Oracle Workflow system activity, suspend and resume processes, retry activities that end in error, and purge obsolete Workflow data.

The *License Manager* component of Oracle Applications Manager enables you to manage all aspects of licensing your products and related features. For example, you can obtain reports of currently licensed products, country-specific functionalities, and languages, as well as an overall licensing summary for your system. You can also license additional products, country-specific functionalities, and languages.

Monitoring

Oracle Applications Manager provides extensive monitoring features for Oracle E-Business Suite. Charts and graphs provide high-level summaries, and you can drill down for more detailed information as required.

OAM can alert you when certain errors occur within the E-Business Suite system, and will provide detailed context information to help in diagnosing those errors. Information from system components, including performance details, availability, configuration, diagnostic data, and security concerns can all be presented in the context of affected business flows, so that you can better understand any impact on your business.

Monitoring support is provided for application tier services such as concurrent managers, Oracle Forms listeners, Reports servers, and Web servers. For currently running processes such as forms or concurrent requests, you can examine both application tier and database session details, right down to the currently executing SQL.

Additional monitoring features include the ability to observe CPU and memory utilization, inspect runtime diagnostics, and view log files for Forms processes. You can look into JServ processes to examine memory utilization, database connection usage, and Oracle Applications Framework activity. OAM can also be configured to automatically identify and alert you to any runaway Forms processes or runaway concurrent requests.

Maintenance

You can easily determine which patches have been applied to a system, including the individual patches included in mini-packs, maintenance packs, and merged patches. Oracle Applications Manager also enables you to examine the patched files on a system, and identify all the patches that altered a given file. For each patch applied, you can see the actions taken by each patch driver.

The *Patch Wizard* tool built into Oracle Applications Manager can recommend patches for your Oracle Applications system. The wizard takes patch data downloaded from Oracle, analyzes that data against the Applications system, and recommends patches based on criteria you specify. The wizard can also analyze individual patches, identify any prerequisites missing on the system, and show the impact that the patches would have on the system in terms of affected applications, files, and other areas. Additionally, the wizard can download multiple patches from Oracle and merge them into a single patch.

Cloning of an Applications system is greatly simplified by using Oracle Applications Manager. You can manage the cloning process directly from the console, initially choosing either *Simple Clone* (which makes an exact copy), or *Advanced Clone* (which allows modification of cloning parameters). Oracle Applications Manager then automates much of the work, guiding you through any remaining decisions and manual steps. Cloning scenarios currently include the ability to clone single node to single node, single node to multi-node, and multi-node to multi-node.

Administrators can use the *Manage Downtime* feature to schedule downtimes. When a downtime is scheduled, a message on the Home Page notifies users of the period when the system will not be available. The user has to log in, or return to the Home Page to see this message. If the downtime is in progress, an appropriate warning page will replace the login page.

During the scheduled downtime period, administrators with suitable database privileges are able to log into OAM to monitor the progress of AutoPatch and other Applications DBA utilities; access to other parts of the E-Business Suite is not allowed.

Diagnostics and Repair

OAM provides troubleshooting wizards to automate complex recovery steps. Tools such as Concurrent Manager Recovery help you diagnose problems and take corrective action from the same screen. The *Service Infrastructure Troubleshooting Wizard* ensures that the Generic Service Management (GSM) infrastructure is running normally, and if not, will guide the administrator through the steps needed to bring GSM up.

Additionally, the Applications Dashboard provides a gateway into Oracle Diagnostics for Oracle Applications, summarizing test results and providing access to detailed test reports. Output from all diagnostic interfaces can be captured and packaged for shipment to Oracle Support simply and easily. Other features include provision for customers to download the latest diagnostics pack supplied by Oracle Support, and create their own diagnostic tests as required.

Additional Features

Extensions

You can extend Oracle Applications Manager using your own custom SQL scripts. Not only can SQL scripts be organized and accessed from within the OAM console, but drilldowns can be enabled from script output to standard Oracle Applications Manager interfaces. For example, if you obtain concurrent request IDs from an SQL script, you can then drill down into the standard Oracle Applications Manager interface to obtain details of the concurrent requests.

Support Cart

The OAM *Support Cart* automates the collection of detailed node-specific information that can be sent to Oracle Support to assist with problem diagnosis. The information collected includes details of products installed, patches applied, database version, current database parameters, and system topology. Other pages from OAM can be added to the Support Cart if desired.

Generic Service Management

Traditionally, application tier processes such as Forms listeners, HTTP servers, and concurrent managers had to be started and monitored individually by system administrators, a time-consuming and potentially error-prone exercise. With the transition to web-based applications, the number of application tier processes required for Oracle's Applications products has increased significantly, with many Applications products making use of multiple application tier *services* to support one or more processes.

Service processes are similar to concurrent manager and transaction manager processes, and must be kept running on an application tier for the proper functioning of their associated products. Management of the services is complicated by the fact that they may be distributed across multiple host machines.

The *Generic Service Management* (GSM) feature simplifies management of these generic service processes, by providing a fault-tolerant framework with a central management console built into Oracle Applications Manager. With Generic Service Management, the Internal Concurrent Manager (ICM) manages the various service processes across multiple hosts. On each host, a *Service Manager* acts on behalf of the ICM, allowing the ICM to monitor and control service processes on that host.

System administrators can configure, monitor, and control services through Oracle Applications Manager, which communicates with the ICM. Generic Service Management provides a fault-tolerant system: if a service process exits unexpectedly, the ICM will automatically attempt to restart the process. If a host fails, the ICM may start the relevant service processes on a secondary host. The ICM itself is monitored and kept alive by Internal Monitor processes located on various hosts.

This new application tier service management infrastructure has several benefits:

- The service processes no longer need to be manually and individually started and monitored by Applications system administrators
- Administrators can configure and control the services through Oracle Applications Manager
- As with concurrent manager processes, system administrators can use work shifts to determine the number of processes that will be active for a service on a given node for a given time period
- Services can take advantage of the process distribution and fault tolerance capabilities that have been developed for concurrent processing

Generic Service Management is available out of the box with Oracle E-Business Suite Release 11.5.10.2, and can also be used with other AutoConfig-enabled releases.

Authentication and Integration

Introduction

The subject of authentication is a broad one, which covers a variety of technologies and components. This chapter provides a survey of the key architectural concepts and decisions involved in setting up the required level of authentication for an organization.

Authentication of Oracle E-Business Suite users can be configured to be straightforward and out of the box, using the traditional FND_USER mechanism, or it can involve various additional features and levels of sophistication, such as single sign-on and use of optional products such as Oracle Portal and Oracle Discoverer. The system administrator can choose the optimal solution for an installation, taking into account factors such as simplicity of setup and maintenance, the possible need for a single point of access to enterprise-wide applications, and the ability to integrate with third-party user directories, as well as the overall security requirements of the organization.

Advanced features that are discussed briefly include the tasks involved in keeping user profile information automatically synchronized across an enterprise, and the steps needed to link an account in Oracle Internet Directory to multiple application accounts in Oracle E-Business Suite Release 11*i*.

Note that the solutions described here do not address the issue of *authorization*. After a user has been authenticated, Oracle E-Business Suite Release 11*i* retrieves the authorization information associated with the application account the user is logged into. Authorization information for application accounts is managed through Applications responsibilities. Oracle E-Business Suite Release 11*i* applies authorization checks as and when required during the user's session.

Oracle9*i* Application Server

The Oracle9*i* Application Server (9*i*AS) provides a sophisticated, J2EE-certified application server. It was the first application server to integrate all the technology required to develop and deploy e-business portals, transactional applications, and Web services into a single product.

Oracle9*i* Application Server offers a productive development environment for developers to create Internet Applications, including J2EE Applications, Web Services, Enterprise Portals, Wireless and Business Intelligence Applications. Its open and integration-ready architecture and standards compliance ensures that Web applications can integrate with an organization's existing IT environment, including legacy systems, applications of suppliers and partners, and Oracle and non-Oracle databases.

Oracle E-Business Suite utilizes 9*i*AS to provide various components of the Applications infrastructure. These are installed as standard, with no specific user action needed.

Note: For further details about the use of Oracle9i Application Server with Oracle E-Business Suite Release 11i, see OracleMetaLink Note 186981.1, *Oracle Application Server with Oracle E-Business Suite Release 11i Frequently Asked Questions*.

Oracle Application Server 10g

Oracle E-Business Suite Release 11i is certified for integration with the following Oracle Application Server 10g components running on a *standalone* server: Oracle Single Sign-On 10.1.2, Oracle Internet Directory 10.1.2, Oracle Portal 10.1.2, and Oracle Discoverer 10.1.2.

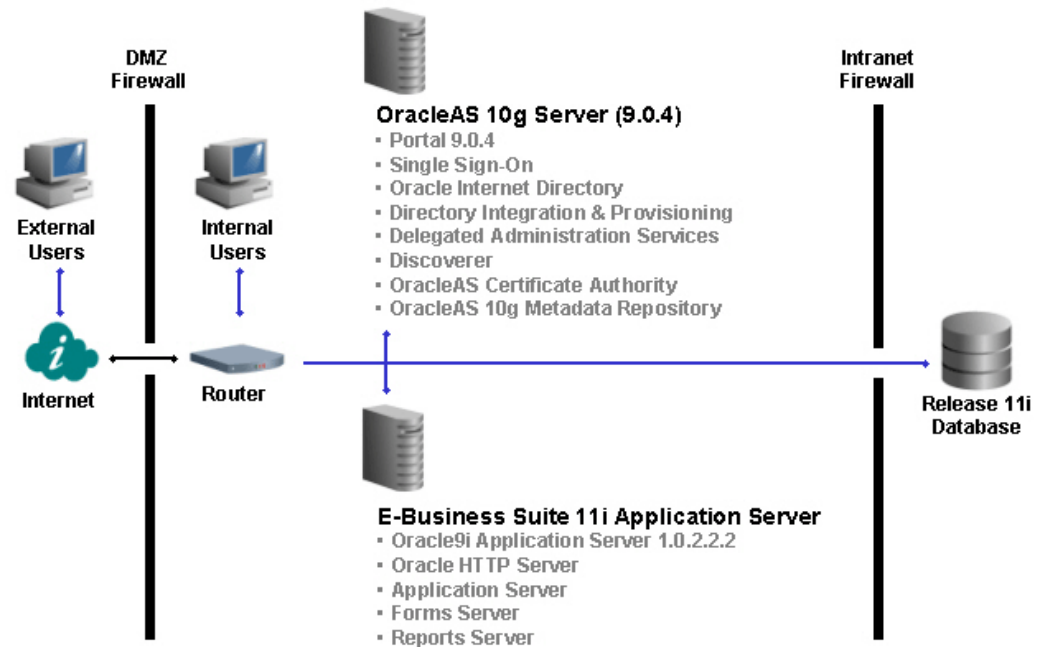
Note: This integration does not upgrade an existing Release 11i technology stack to Oracle Application Server 10g.

Benefits of integrating Oracle Application Server 10g with Oracle E-Business Suite Release 11i include:

- Performance, scalability and high-availability via distributed architectures.
- The ability to connect a single Enterprise Portal to web providers running on multiple Release 11i instances.
- Uniform Single Sign-On support for all Release 11i Applications products.
- Bidirectional Oracle Internet Directory-to-FND_USER synchronization.
- Link-on-the-fly support for environments where the Single Sign-On userids in Oracle Internet Directory differ from the Release 11i userids.
- One-to-many support for environments where a Single Sign-On userid may be associated with one or more Release 11i userids.
- Bookmarkable URLs, where Applications Navigator Portlet will produce links that authenticate users and create Applications sessions as needed.
- Integration with third-party single sign-on services (e.g. Netegrity, Tivoli, Entrust) and *Lightweight Directory Access Protocol (LDAP) V.3* directories such as SunONE/iPlanet and Microsoft Active Directory.

The architecture shown in the following diagram is an example of a relatively simple deployment. Note that the Portal, Discoverer, and OracleAS Certificate Authority implementations are optional.

Figure 8-1 Example Application Server Architecture



Support for more advanced deployment topologies is also available, including multi-node load balancing configurations, Real Application Clusters (RAC), and other distributed architectures.

Note: For details of installing Oracle Application Server 10g in an Oracle E-Business Suite Release 11i environment, see OracleMetaLink Note 233436.1.

Oracle Portal

Oracle Application Server Portal (Oracle AS Portal) is a complete, browser-based environment for the development, deployment, administration, and configuration of enterprise class portals. Oracle AS Portal incorporates a complete portal building framework, with self-service publishing features to facilitate creation and management of the information accessed within your portal. A wide variety of portal interfaces and configurations are possible, from a simple departmental-level publishing portal to an Internet-accessible portal that serves both customers and employees. Tight integration with other components of the Oracle Application Server and with the Oracle database ensures that the solution can scale to an enterprise class audience.

Oracle Discoverer

Discoverer, Oracle's powerful ad hoc query, reporting, analysis, and web publishing tool, can be utilized to allow business users at all levels of the organization gain immediate access to information from Oracle E-Business Suite Release 11i. Users can employ Discoverer to analyze data from selected business areas including Financials, Operations, Human Resources, Purchasing, Process Manufacturing, and Activity Based Management.

Note: For further details of using Oracle Discoverer with Oracle E-Business Suite Release 11*i*, see OracleMetaLink Note 313418.1, *Using Discoverer 10.1.2 with Oracle Applications 11i*.

Enterprise-Wide Single Sign-On

Single sign-on functionality enables users to access Oracle E-Business Suite and other applications through a single user ID, without having to log in to each application separately. Oracle E-Business Suite supports the use of single sign-on functionality via *Oracle Single Sign-On*, *Oracle Internet Directory (OID)*, and *Oracle Portal*.

Implementing an enterprise-wide single sign-on solution involves significant changes to the mechanism by which Oracle E-Business Suite Release 11*i* users are authenticated. Instead of authentication being performed natively, via the FND_USER table, this functionality is delegated to Oracle Single Sign-On, which can either:

- Perform user validation itself, against information stored in Oracle Internet Directory.
- Delegate validation to a third-party single sign-on server.

With either of these solutions, Oracle E-Business Suite Release 11*i* accepts identities vouched for by the single sign-on mechanism. Oracle Internet Directory complements this by acting as an integration point that enables Oracle E-Business Suite Release 11*i* to participate in enterprise level user management.

Note: Note that where a third-party single sign-on server is in use, Oracle Single Sign-On and Oracle Internet Directory are still required, to provide a bridge between E-Business Suite Release 11*i* and the third-party single sign-on solution.

Each E-Business Suite instance must still maintain a record of registered users, in the form of the traditional application accounts. However, the level of abstraction needed for an enterprise level user requires a mechanism that can uniquely identify a user across the enterprise. This is accomplished via a *globally unique identifier (GUID)*. Oracle Internet Directory and Oracle E-Business Suite store GUID information for each enterprise level user. The GUID can be considered as an identity badge that is recognized by both Oracle Internet Directory and Oracle E-Business Suite.

Another requirement in such an environment is for user enrollment to be done only once, at well defined places, with the user subsequently being known to the rest of the enterprise. Two additional features enable this:

- Support for automatic propagation of application information across an enterprise, via a *synchronization* process between Oracle Internet Directory and a third-party LDAP server.
- Support for automatic propagation of user information across an enterprise, via a *provisioning* process between Oracle Internet Directory and Oracle E-Business Suite Release 11*i*.

User information in external, third-party user directories can be synchronized with Oracle Internet Directory using the LDAP protocol. With Oracle Internet Directory, customers can manage and publish user information in a central location that various application systems, including the Oracle E-Business Suite, can reference.

Much of the complexity involved with integrating Oracle E-Business Suite into a single sign-on environment arises because of the need to consolidate fragmented or

duplicated user data in the single sign-on environment, as a legacy of integrating previously-isolated systems.

The solution described in this chapter provides mechanisms to link the existing data together using the GUID. In addition, bulk migration tools can be used to move a large number of users between Oracle Internet Directory and E-Business Suite during the transition to an integrated single sign-on environment.

Note that full synchronization of user credentials between Oracle Internet Directory and E-Business Suite Release 11i requires deployment of Oracle Application Server 10g.

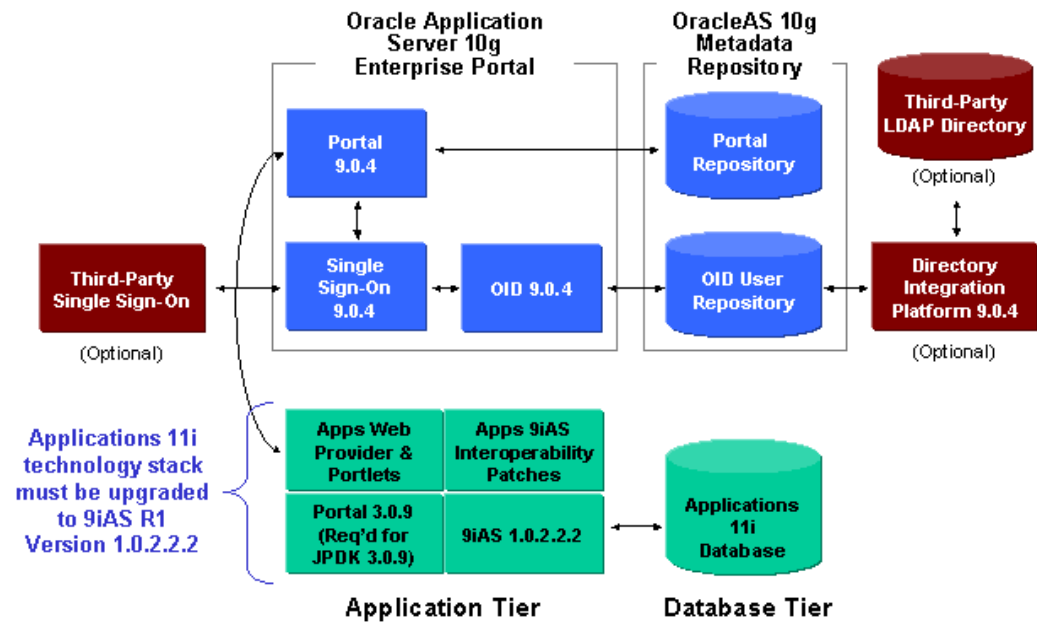
Note: For more information on implementing single sign-on with Oracle Single Sign-On and Oracle Internet Directory, see the *Oracle Applications System Administrator's Guide* and Oracle MetaLink Note 233436.1, *Installing Oracle Application Server 10g with Oracle E-Business Suite Release 11i*.

Application Server Integration Options

Application Server 10g can act as an *integration hub* that enables the Oracle E-Business Suite to work in conjunction with other enterprise software, including software from third-party vendors.

In the logical architecture shown below, the Applications web provider and portlets are installed on an existing Applications Release 11i (11.5.8 or higher) installation that has been upgraded to 9iAS 1.0.2.2.2 and Portal 3.0.9.

Figure 8-2 Application Server Integration Architecture



Key points of this configuration include:

- Users log into a standalone instance of Oracle Portal 9.0.4 running on an enterprise portal node that has been built with Oracle Application Server 10g (note that Portal implementation is optional.)

- The standalone Oracle Application Server 10g node includes the Oracle HTTP Server and is responsible for serving web requests, routing, and overall Portal processing.
- The Oracle Single Sign-On node authenticates users against Oracle Internet Directory, or optionally may be configured to delegate user authentication to third-party single sign-on services such as Netegrity SiteMinder.
- The Oracle Internet Directory server node manages user credentials, synchronizing them bidirectionally with the Applications 11i native user directory (FND_USER) via the Oracle Directory Integration Platform. Oracle Internet Directory may optionally be configured to integrate with third-party LDAP user directories via the Oracle Directory Integration Platform. Alternatively, Oracle Internet Directory may optionally be configured to delegate user authentication to a third-party LDAP directory via the OID External Authentication Plug-In, which includes support for SunONE and Microsoft Active Directory.
- The integrated E-Business Suite application tier server node with 9iAS 1.0.2.2.2 is responsible for handling the SQL request, and generating the portlet HTML that is consolidated by the Portal application server into a home page for the user.

Basic Single Sign-On Deployment Scenario

This section outlines a simple deployment scenario where an existing Oracle E-Business Suite instance is integrated with a new Oracle Single Sign-On and Oracle Internet Directory infrastructure. A subsequent discussion considers additional factors, such as the existence of a third-party single sign-on solution, or the presence of multiple user repositories.

Note: This section provides a high-level overview of the common tasks that will apply to all installations. The exact steps needed for the requirements of a particular site will be more detailed.

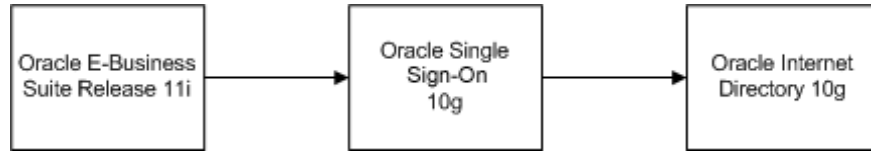
The starting point of this scenario is an existing Oracle E-Business Suite Release 11i installation, plus a new Oracle Application Server 10g installation (including Oracle Single Sign-On and Oracle Internet Directory) on a different machine.

Oracle Internet Directory has no currently existing users apart from pre-seeded users, and Oracle Portal is not implemented. The requirement is to integrate Oracle E-Business Suite Release 11i with Oracle Single Sign-On and Oracle Internet Directory.

Key Goals

- Oracle E-Business Suite Release 11i will delegate user sign-on and authentication to Oracle Single Sign-On
- Oracle Single Sign-On will authenticate user credentials against user entries in Oracle Internet Directory
- Oracle Internet Directory will store every user's single sign-on account id and password

Figure 8-3 Deploying E-Business Suite with Oracle Single Sign-On and Oracle Internet Directory



User Management Options

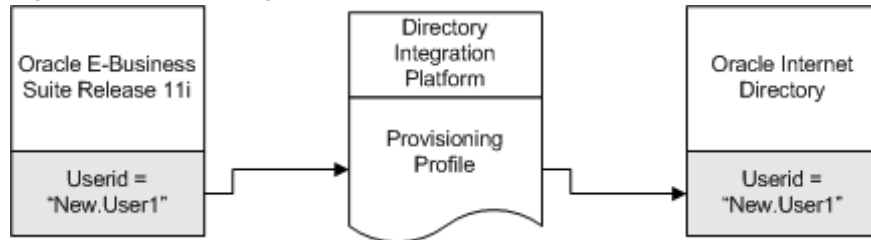
Existing Oracle E-Business Suite Release 11*i* application accounts are migrated to single sign-on accounts in Oracle Internet Directory using the *Bulk Migration Tool*. After the migration, a system administrator has a number of user management options, related to the location(s) where user information is created, and where it is provisioned (sent) to.

Option 1

All user information is created in Oracle E-Business Suite Release 11*i*, then provisioned into Oracle Internet Directory.

- Oracle E-Business Suite Release 11*i* is configured as a *provisioning integrated application* with Oracle Internet Directory
- System administrators configure the provisioning integration via *provisioning profiles*

Figure 8-4 Provisioning User Information from E-Business Suite to Oracle Internet Directory



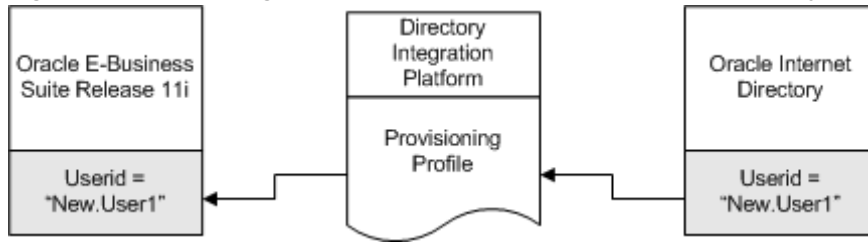
The creation of a new application account in Oracle E-Business Suite Release 11*i* will automatically trigger the creation of a new single sign-on account in Oracle Internet Directory. Some of the user attributes from the application account may be provisioned in the single sign-on account in Oracle Internet Directory during account creation.

Option 2

All user information is created in Oracle Internet Directory, then provisioned into Oracle E-Business Suite Release 11*i*:

- Oracle E-Business Suite Release 11*i* is configured as a provisioning integrated application with Oracle Internet Directory
- System administrators configure the provisioning integration via *provisioning profiles*

Figure 8-5 Provisioning User Information from Oracle Internet Directory to E-Business Suite



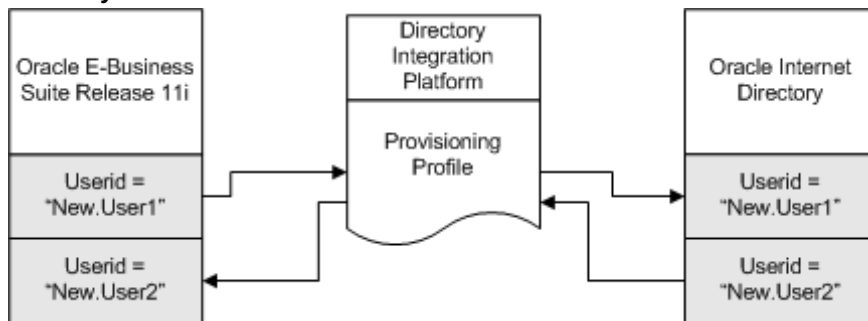
The creation of a new single sign-on account in Oracle Internet Directory will automatically trigger the creation of a new application account in Oracle E-Business Suite Release 11*i*. Some of the user attributes from the single sign-on account may be provisioned in the application account in Oracle Internet Directory during account creation.

Option 3

All user information is created in either Oracle Internet Directory or Oracle E-Business Suite Release 11*i*, then provisioned into the other system:

- Oracle E-Business Suite Release 11*i* is configured as a provisioning integrated application with Oracle Internet Directory
- System administrators configure the provisioning integration via provisioning profiles

Figure 8-6 Provisioning User Information Between E-Business Suite and Oracle Internet Directory



The creation of a new application account in Release 11*i* will automatically trigger the creation of a new single sign-on account in Oracle Internet Directory, and the creation of a new single sign-on account in Oracle Internet Directory will automatically trigger the creation of a new application account in Release 11*i*.

During account creation, some of the user attributes from the application account may be provisioned in the single sign-on account in Oracle Internet Directory during account creation, and some of the user attributes from the single sign-on account may be provisioned in the application account in Oracle Internet Directory.

Synchronizing User Attributes

For all three of the above options, a set of user attributes can, on being updated from either system, optionally be synchronized between Oracle E-Business Suite Release 11*i*

and Oracle Internet Directory. This is accomplished by configuring the provisioning profile.

Signing On

Attempting to gain access to an Oracle E-Business Suite Release 11*i* environment, a user who has not yet been authenticated with Oracle Single Sign-On is directed to a Single Sign-On login page, which can be customized to suit an individual site.

After authentication via Oracle Single Sign-On (or if authentication has previously been carried out) the user is redirected to the requested page or the user's home page in the Oracle E-Business Suite Release 11*i*.

Signing Out

When a user logs out of an Oracle E-Business Suite instance, the user is also logged out of Oracle Single Sign-On, as well as any *partner applications* applications that have been integrated with Oracle Single Sign-On. The user will see a logout page that lists all the applications the user has been successfully logged out of.

Session Timeout

It is important to understand the timeout behavior of the different sessions in a single sign-on environment, to ensure the appropriate level of security is maintained.

- If a user's application session has timed out, but not his single sign-on session, he will be directed to Oracle Single Sign-On, and then back to Oracle E-Business Suite, without being prompted to re-authenticate.
- If a user's application session and single sign-on session have both timed out, he will be directed to the single sign-on login page to re-authenticate, and then redirected back to Oracle E-Business Suite.

Until a user's application session times out (or he explicitly logs out), he can continue to access the partner application even if his Oracle Single Sign-On security cookie has expired. Since the application session timeout value takes precedence over the Single Sign-On timeout setting, Oracle recommends setting the application session timeout value to be equal to or less than that of Oracle Single Sign-On.

Advanced Single Sign-On Deployment Scenarios

This section outlines four more deployment scenarios. The guidelines given should be regarded as providing a high-level strategy rather than definitive instructions, as all real world deployments will be unique, and require detailed planning. The outline solutions build upon the basic scenario discussed above.

Scenario 1

Requirement - Need to enable Oracle Single Sign-On with Oracle E-Business Suite Release 11i

Starting Environment

- Multiple new Oracle E-Business Suite Release 11*i* environments have been installed
- Other than the default administrative accounts, no user accounts have been registered yet
- Oracle Portal is not implemented
- No Single Sign-on infrastructure in place

Solution

- Oracle Application Server 10g with Oracle Single Sign-On and Oracle Internet Directory are needed for the integration required
- Oracle E-Business Suite Release 11i will delegate user sign-on and authentication to Oracle Single Sign-On
- Oracle Single Sign-On authenticates user credentials against user entries in Oracle Internet Directory
- Oracle Internet Directory contains every user's single sign-on account id and password

Either Oracle Internet Directory or one Oracle E-Business Suite Release 11i instance can be designated as the source of user enrollment, with the following implications:

- If Oracle Internet Directory is the source, details of user accounts can be propagated to each Oracle E-Business Suite instance via the provisioning process.
- If an Oracle E-Business Suite instance is the source, the provisioning process will propagate user accounts from that instance to Oracle Internet Directory, and then to the other Oracle E-Business Suite instances.

Optionally, user profile information in an Oracle E-Business Suite Release 11i instance can be kept synchronized with the information in Oracle Internet Directory.

Scenario 2

Requirement - Need to integrate new installation of Oracle E-Business Suite Release 11i with existing third-party single sign-on and user directory infrastructure

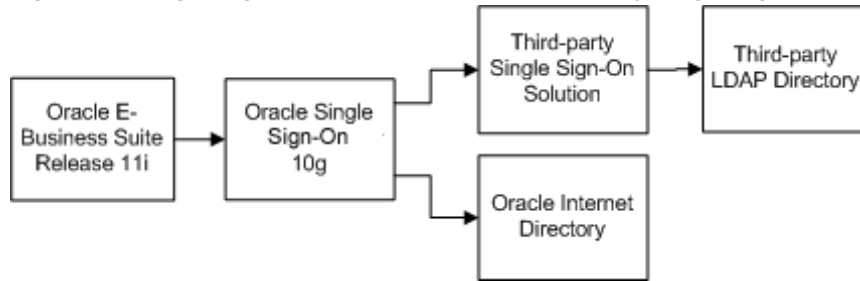
Starting Environment

- Oracle E-Business Suite Release 11i has been newly installed using the Rapid Install Wizard.
- Other than the default administrative accounts, no user accounts have been registered yet.
- Oracle Portal is not implemented.
- A third-party single sign-on solution such as Netegrity SiteMinder is in use as a corporate single sign-on solution.
- A third-party LDAP directory such as SunONE/iPlanet is in use as a corporate user directory.

Solution

- Oracle Application Server 10g (including Oracle Single Sign-On and Oracle Internet Directory) is needed for the integration.
- Oracle E-Business Suite and Oracle Single Sign-On must be set up so that Oracle E-Business Suite delegates authentication to Oracle Single Sign-On, which in turn delegates the functionality to the third-party single sign-on server in use.
- Oracle Internet Directory needs to be set up to synchronize a minimal set of information from the third-party LDAP directory for all users who will access Oracle E-Business Suite via single sign-on.
- Oracle Internet Directory also needs to be set up to provision users in Oracle Internet Directory to Oracle E-Business Suite.

Figure 8-7 Integrating E-Business Suite with Third-Party Single Sign-On and User Directory



Existing users in the third-party LDAP directory can be bulk migrated into Oracle Internet Directory, and then bulk migrated into Oracle E-Business Suite.

Optionally, user profile information in Oracle E-Business Suite can be kept synchronized with the information in the third-party LDAP directory.

Scenario 3

Requirement - Need to integrate existing Oracle E-Business Suite Release 11i with existing third-party single sign-on and user directory infrastructure

Starting Environment

- Oracle E-Business Suite Release 11i is in use, and has an up to date user repository.
- Oracle Portal is not implemented.
- A third-party corporate single sign-on solution such as Netegrity SiteMinder is in use and is to be retained.
- A third-party LDAP directory such as SunONE/iPlanet is in place as a corporate user directory and is to be retained.
- At the start of the implementation, a given user may exist in both Oracle E-Business Suite Release 11i and the third-party LDAP directory, with either the same user name in both or a different user name in each.

Solution

- Oracle Application Server 10g (including Oracle Single Sign-On and Oracle Internet Directory) is needed for the integration.
- Oracle E-Business Suite and Oracle Single Sign-On need to be set up so that Oracle E-Business Suite delegates authentication to Oracle Single Sign-On, which in turn delegates the functionality to the third-party single sign-on server.
- Oracle Internet Directory must be configured to synchronize a minimal set of information from the third-party LDAP directory for users who will access Oracle E-Business suite via single sign-on.
- Existing users in the third-party LDAP directory can be bulk migrated into Oracle Internet Directory.
- Existing accounts in both Oracle E-Business Suite and the third-party LDAP directory can be linked.
- With proper planning, new users can be synchronized from the third-party LDAP directory into Oracle Internet Directory, and then into Oracle E-Business Suite.

- Optionally, user profile information in Oracle E-Business Suite can be kept synchronized with the information in the third-party LDAP directory.

A simpler variant of this scenario arises when no third-party single sign-on/LDAP directory is involved. There is only an existing Oracle E-Business Suite Release 11i Installation plus an Oracle Single Sign-On and Oracle Internet Directory infrastructure. In such a case, all steps relating to third-party (non-Oracle) software can be ignored.

Scenario 4

Requirement - Need to enable Oracle Single Sign-On with multiple Oracle E-Business Suite Release 11i installations where no Oracle Single Sign-On infrastructure is currently in place

Starting Environment

- Multiple Oracle E-Business Suite Release 11i instances are implemented, and each has an existing user population.
- Oracle Portal is not implemented.
- No existing Oracle Single Sign-On infrastructure is in place.

Solution

- Oracle Application Server 10g (including Oracle Single Sign-On and Oracle Internet Directory) is needed for the integration.
- Each Oracle E-Business Suite instance delegates user sign-on and authentication to Oracle Single Sign-On.
- Oracle Single Sign-On authenticates user credentials against user entries in Oracle Internet Directory.
- Oracle Internet Directory contains every user's single sign-on account id and password.
- A single sign-on account needs to be created for every user in Oracle Internet Directory.
- Existing applications accounts in Oracle E-Business Suite instances need to be linked to the single sign-on account.
- Optionally, user profile information in Oracle E-Business Suite can be kept synchronized with the information in Oracle Internet Directory.

Note: For a full list of scenarios relating to integration of Oracle E-Business Suite Release 11i with Oracle Internet Directory and Single Sign-On, refer to OracleMetaLink Note 261914.1, *Integrating Oracle E-Business Suite Release 11i with Oracle Internet Directory and Oracle Single Sign-On*.

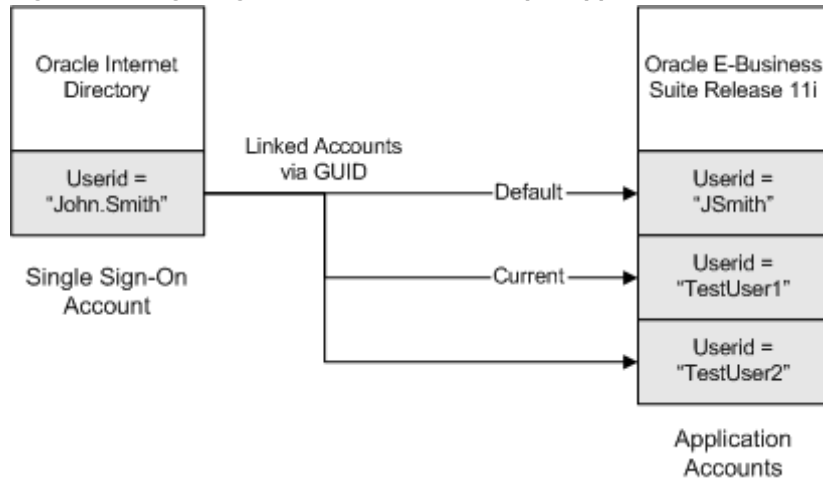
Advanced Single Sign-On Options

There are a number of advanced options that may be employed in specialized circumstances; one example is described here.

Linking Multiple Application Accounts to a Single Oracle Single Sign-On Account

Normally, a single sign-on account in Oracle Internet Directory will correspond to a single application account in Oracle E-Business Suite Release 11i. However, in special cases a user may need to have a single sign-on account in Oracle Internet Directory and multiple application accounts in Oracle E-Business Suite Release 11i.

Figure 8-8 Single Sign-On Account with Multiple Application Accounts



If required, this feature can be enabled by system administrators via the profile option 'Applications SSO Allow Multiple Accounts'.

High Availability

Introduction

High Availability covers a range of options and features that can minimize planned and unplanned downtime, or facilitate recovery after a period of downtime. They include:

- Patching Hints and Tips
- Maintenance Mode
- Shared APPL_TOP and Shared Application Tier File System
- Staged APPL_TOP
- NOLOGGING in the Applications database
- Distributed AD
- Disaster Recovery best practices

This section will provide a high-level guide to the key features that can help make an Oracle E-Business Suite highly available, with the emphasis on guidelines for making the correct decisions when planning a new installation or upgrade.

Patching Hints and Tips

Patch application is a key activity undertaken by Oracle Applications DBAs. If you need to apply a large number of patches, the required downtime can be significant. However, there are several simple ways of minimizing this downtime:

- **Schedule periodic downtime for application of the latest maintenance packs or family packs** - The more up-to-date your system, the less likely you are to experience known problems, and the easier it will be to resolve any new issues that may arise.
- **Keep AD up-to-date** - Running at the latest AD mini-pack level allows you to take full advantage of new features designed to reduce downtime and simplify maintenance.
- **Keep your test system current with your production system** - When you test the application of a patch, the test must be realistic in terms of current patch level and transaction data: you can employ the Rapid Clone tool, described in Chapter 6, to create a copy of your production system for tests.
- **Consolidate multiple patches with AD Merge Patch** - Merging multiple Applications patches into a single patch not only reduces overall downtime by eliminating duplicate tasks, but minimizes the scope for error that would arise in applying a number of separate patches.

- **Use a shared APPL_TOP and shared application tier file system** - These are recommended for environments with multiple application tier nodes.
- **Use a staged APPL_TOP** - This will reduce the downtime to the period required for the database update.
- **Use the Distributed AD feature** - This helps make full use of available hardware resources.
- **Perform maintenance during normal operation where possible** - For example, you can gather schema statistics or patch online help while the system is in use.

Where applicable, these strategies are described further below.

Maintenance Mode

Maintenance Mode is a new mode of operation introduced with Release 11.5.10, in which the Oracle Applications system is made accessible only for patching activities. This provides optimal performance for AutoPatch sessions, and minimizes downtime needed.

Note: Maintenance Mode is only needed for AutoPatch sessions. Other AD utilities do not require Maintenance mode to be enabled.

Administrators can schedule system downtime using Oracle Applications Manager, and send alert messages to users about the impending downtime. When Maintenance Mode is entered, users attempting to log on to Oracle Applications are redirected to a system downtime URL.

There are several practical points relating to the use of Maintenance Mode:

- You can toggle Maintenance Mode between Enabled and Disabled using the new *Change Maintenance Mode* menu in AD Administration, or the equivalent function in Oracle Applications Manager.
- Although you can run AutoPatch with Maintenance Mode disabled, there will be a significant degradation in performance.
- There is a separate logon page for *Restricted Mode* access while the system is in Maintenance Mode. Restricted Mode allows administrators access to specific privileged functionality, for example to view the timing report that shows the progress of a patching session.

Shared APPL_TOP and Shared Application Tier File System

A traditional multi-node installation of Oracle Applications Release 11*i* requires each application tier node to maintain its own file system, which is made up of:

- The *APPL_TOP file system*, consisting of the APPL_TOP and COMMON_TOP directories
- The *application tier technology stack file system*, consisting of the 8.0.6 and iAS Oracle homes

Installation and migration options were subsequently introduced to enable a single APPL_TOP to be shared between all the application tier nodes of a multi-node system. This is referred to as a *Shared APPL_TOP File System*, usually abbreviated to *Shared APPL_TOP*.

A further new capability was the option to *merge* the APPL_TOPs of multiple nodes, each with its own set of application tier services, to give a single APPL_TOP that could then be shared between them all. A less common scenario would be where you wanted to merge the APPL_TOPs as part of a strategy to reduce the number of nodes, i.e. move from a multi-node installation to a single node installation.

Release 11.5.10.2 extends these concepts to enable sharing of the application tier technology stack file system as well, the result being known as a *Shared Application Tier File System*.

This section describes the benefits of using a shared APPL_TOP and shared application tier file system in an Oracle Applications Release 11*i* environment. Current restrictions are also noted where applicable.

Shared APPL_TOP

A traditional multi-node application tier installation requires the Applications file system to be installed on each application tier node. In contrast, a shared APPL_TOP means that there is a single APPL_TOP capable of being accessed simultaneously by two or more machines. In a shared APPL_TOP environment, each node has its own APPS<CONTEXT_NAME>.env file and Applications context file.

A shared APPL_TOP environment requires the relevant shared disk volume to be attached to a mount point of the same name on all participating machines, so the path to the APPL_TOP will have the same value on all machines. In cases where the COMMON_TOP file system is outside the APPL_TOP, it must also be mounted on all relevant machines.

After the required setup steps have been carried out, the participating machines can be used to provide standard application tier services such as Forms, Web, and Concurrent Processing, with any changes made to the shared APPL_TOP file system being seen immediately by all nodes.

Note: Installing two separate APPL_TOPs on to a shared disk volume does not result in a shared APPL_TOP.

A single APPL_TOP can be used to support multiple node types in an application tier. For sites that have an APPL_TOP already installed and performing all the required functions, that particular APPL_TOP can be shared immediately: it is not necessary to start from scratch and create a single APPL_TOP that can provide services for the various application tier node types (such as Forms, Web, Concurrent Processing, or Admin).

However, such a single APPL_TOP must contain all the files for each service it is intended to provide: for example, if you install a Forms-only node with the Rapid Install Wizard and later decide to share it, you cannot expect it to be able to run other application tier services in the absence of the requisite files.

Shared APPL_TOP Benefits

Benefits of using a shared APPL_TOP include:

- The ease with which machines can be added to an existing system facilitates the deployment of a large number of application tier machines, to help maximize system availability and cater for additional usage requirements.
- Overall disk space requirements are greatly reduced, as there is only a single copy of the relevant Applications code.

- Since there is only one physical APPL_TOP file system, administrative tasks such as patch application need only be carried out once, on any node, and take effect immediately on all nodes.
- Complements the Distributed AD feature (described later) in improving the efficiency of patch application and reducing downtime.

Shared APPL_TOP Restrictions

There are currently a number of limitations on shared APPL_TOP usage:

- When a patch is applied to an ORACLE_HOME, it must also be applied to the corresponding ORACLE_HOMEs on all other nodes. This does not apply if you follow the recommended strategy of using a shared application tier file system as well as a shared APPL_TOP.
- An APPL_TOP can only be shared across machines running either identical or binary compatible operating systems, an example of the latter being Solaris 6 and Solaris 8.
- Shared APPL_TOP functionality is not currently available on Windows.

Shared Application Tier File System

In a shared application tier file system, all application tier files are installed on a single shared disk resource that is mounted on each application tier node. Any application tier node can be configured to perform any of the standard application tier services, such as serving forms or web pages, and all changes made to the shared file system are immediately visible on all the application tier nodes.

Deploying a shared application tier file system can be considered an extension of the more traditional shared APPL_TOP architecture. The benefits and restrictions of a shared application tier file system are similar to those for a shared APPL_TOP. For example, you can reduce maintenance and other management overheads by migrating any existing Oracle Applications Release 11i installation to utilize a shared application tier file system, with the restriction that the machines involved must be binary compatible.

Shared Disk Resources

A shared application tier file system can reside on any type of shared disk resource, such as a remote NFS-mounted disk or part of a RAID array. However, you should ensure that performance of the chosen disk resource is adequate to meet peak demand. For example, NFS-mounted disks may give inadequate read or write performance when there is a large amount of network traffic, and RAID arrays must be implemented carefully to strike the appropriate balance between high availability, performance and cost.

Creating a Shared Application Tier File System

The Release 11.5.10.2 Rapid Install will by default configure a multi-node application tier environment to use a shared application tier file system. When running Rapid Install, you will need to specify appropriate mount points to ensure that the paths to APPL_TOP, COMMON_TOP, 8.0.6 ORACLE_HOME, and iAS ORACLE_HOME are the same on each node.

Alternatively, you can migrate any existing Oracle Applications Release 11i system to use a shared application tier file system by following Oracle MetaLink Note 233428.1, *Sharing the Application Tier File System in Oracle E-Business Suite 11i*.

High Availability Features of Shared APPL_TOP and Shared Application Tier File System

Utilizing a shared APPL_TOP and shared application tier file system improves high availability in the following ways:

- It is straightforward to add nodes to an existing installation, to provide greater resilience to node failure or to cater for additional users. This is particularly cost-effective with inexpensive Linux nodes.
- A patch only needs to be applied to one application tier node for its effects to be visible on all other nodes that share the file system. Such a single installation also helps to minimize the duration of planned maintenance downtimes, and reduces the scope for errors during installation.

Staged APPL_TOP

A *staged APPL_TOP* is an APPL_TOP that is copied from a production system for application of one or more patches, then copied back to the production system. This helps to improve availability by minimizing the production downtime needed for patch application.

The same concept can be extended to stage an entire Applications system. This involves the creation of an exact copy of a production system, including all APPL_TOPs and the production database. Patches can subsequently be applied to the staged system while the production system remains operational. Only when all patches have been applied to the test system need production downtime begin. The staged APPL_TOP is used both to apply the updates to the production database, and to synchronize with the production APPL_TOP.

Prerequisites and Restrictions

- Before a production Applications system can be copied, an up-to-date snapshot of the system must exist, to allow patch prerequisites to be checked; AutoPatch will return an error if a current snapshot cannot be found. For each APPL_TOP, run AD Administration, select the *Maintain Applications Files* menu, then the *Maintain snapshot information* menu, and choose the *Update current view snapshot* task.
- An exact copy (clone) of the production database and of each APPL_TOP of your production Applications system must be created. Production and staged systems must have the same APPL_TOP names, to ensure the patching history of the staged APPL_TOP will be valid in the production system.
- You must use different Applications system names for the staged and production systems. In addition, the database of the staged APPL_TOP should have a different SID, to prevent accidental connection to the production system.
- The staged APPL_TOP approach cannot be used to apply AD mini-packs or patches associated with a maintenance pack.

Applying Patches to the Staged System

The staged system is patched the same way as any Oracle Applications system, using AutoPatch to apply the relevant patch drivers. While carrying out this step, no patches may be applied to the production system. If they are, the staged system must be created again.

Note: See *Oracle Applications Maintenance Procedures* for more information on applying patches.

Updating the Production System

Once the staged system has had the required patches applied, the production system can be updated. After all services on the production system have been disabled, you can run AutoPatch, specifying as applicable the database driver or unified driver (see Chapter 1) of the patch you wish to apply. If you applied multiple patches to the staged system, you will need to run the database driver or unified driver for each patch you applied to the stage, in the same order. To reduce downtime, you should consider merging patches prior to staging.

The production APPL_TOP now needs to be synchronized with the staged APPL_TOP. You can complete this while the production database is being updated. Various software utilities can be used to accomplish this task, and some vendors offer hardware solutions as well.

If your topology includes multiple APPL_TOPs, each APPL_TOP must be copied to the production system. If you share a single APPL_TOP, you only need to synchronize one system. The COMMON_TOP directory, which on some systems may reside outside the APPL_TOP, also needs to be updated for each APPL_TOP in the Applications system.

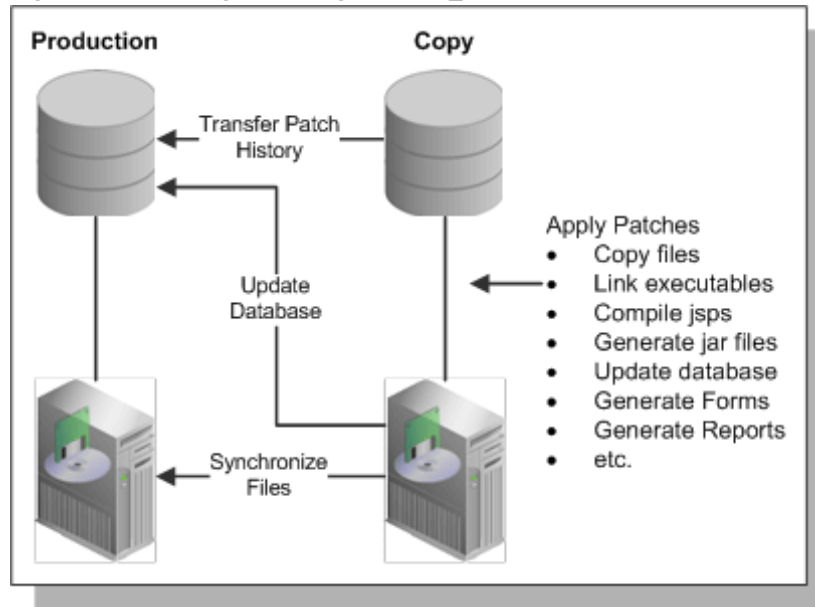
Warning: Certain configuration files, log directories and environment scripts are specific to an APPL_TOP. These files and directories must be *excluded* when copying.

Final Patching Steps

The copy and generate portions of the patch history for patches applied using a staged Applications system are now stored in the staged database, and the database portion of the patch history for these patches is stored in both the staged database and in the production database.

To ensure that the patch history of the production system is complete, you must load the copy and generate portions of all patches applied using a staged Applications system connected to your production database. The *adphmigr.pl* utility can be used to export the patch history for the copy and generate portions of patches applied using a staged Applications system from the staged database. AutoPatch can then be used to import the extracted patch history data into the production database.

Figure 9-1 Patching with Staged APPL_TOP



Combining Staged APPL_TOP and Shared APPL_TOP

The staged APPL_TOP and shared APPL_TOP strategies provide different and complementary benefits. You can, if desired, create a staged APPL_TOP of a shared APPL_TOP system, giving the benefits of shared APPL_TOP (and Distributed AD) on your staged system.

Distributed AD

Many deployments utilize large database servers and multiple, smaller application (middle) tier systems. With the increasing deployment of low cost Linux-based systems, this configuration is becoming more common.

AD has always utilized a job system, where multiple workers are assigned jobs. Information for the job system is stored in the database, and workers receive their assignments based on the contents of the relevant tables. The *Distributed AD* feature offers improved scalability, performance, and resource utilization, by allowing workers of the same AD session to be started on multiple application tier nodes, utilizing available resources to complete their assigned jobs more efficiently.

Requirements for Distributed AD

Because the AD workers create and update file system objects as well as database objects, a shared APPL_TOP (described earlier) must be employed to ensure the files are created in a single, centralized location.

Using Distributed AD

On one of your shared APPL_TOP nodes, you start your AutoPatch or AD Administration session, specifying the number of local workers and the total number of workers.

While using AutoPatch or AD Administration, you can start a normal AD Controller session from any of the nodes in the shared APPL_TOP environment to perform any standard AD Controller operations, using both local and non-local workers. This is

possible because the job system can be invoked multiple times during AutoPatch and AD Administration runs. Each time an individual invocation of the job system completes, distributed AD Controller sessions will wait until either the job system is invoked again (at which point it will once again start the local workers) or until the AD utility session ends (at which point distributed AD Controller will exit).

Note: See *Oracle Applications Maintenance Utilities* for further details of Distributed AD and AD Controller.

AD Controller Log Files

The log file created by AD Controller is created wherever the AD Controller session is started. This is to prevent file locking issues on certain platforms. It is therefore recommended that the AD Controller log file should include the node name from which the AD Controller session is invoked.

Nologging Operations

The *nologging* Oracle database feature is used to enhance performance in certain areas of Oracle E-Business Suite Release 11*i*. For example, it is used during patch installation, and when building summary data for Business Intelligence.

Use of nologging in an operation means that the database redo logs will contain incomplete information about the changes made, with any data blocks that have been updated during the nologging operation being marked as *invalid*. As a result, a database restoration to a point in time (whether from a hot backup or a cold backup) may require additional steps in order to bring the affected data blocks up-to-date, and make the restored database usable. These additional steps may involve taking new backups of the associated datafiles, or by dropping and rebuilding the affected objects. The same applies to activation of a standby database.

Note: Oracle9*i* Release 2 allows logging to be forced to take place, ensuring all data changes are written to the database redo logs in a way that can be recreated in a restored backup, or propagated to a standby database. See *Oracle Data Guard Concepts and Administration Release 2 (9.2)* for details of the *forcelogging* option.

Nologging Principles

At certain times, Oracle E-Business Suite uses the database nologging feature to perform resource-intensive work more efficiently. When an operation uses nologging, blocks of data are written directly to their data file, rather than going through the buffer cache in the System Global Area (SGA).

Instance recovery uses the online redo logs to reconstruct the SGA after a crash, rolling forward through any committed changes in order to ensure the data blocks are valid. Use of nologging does not affect instance recovery.

Database recovery requires rolling forward through the redo logs to recreate the requisite changes, and hence restore the database to the desired point in time. Since nologging operations write directly to the data files, bypassing the redo logs, the redo logs will not contain enough data to roll forward to perform media recovery. Instead, they will only contain enough information to mark the new blocks as invalid. Rolling forward through a nologging operation would therefore result in invalid blocks in the restored database. The same problems will potentially occur upon activating a standby database.

To make the backup or activated standby database usable after a nologging operation is carried out, a mechanism other than database recovery must be used to get or create current copies of the affected blocks.

There are two options, either of which may be appropriate depending on the specific circumstances:

- Create a new copy of the data files, either by backing up the tablespace again, or by refreshing the specific data files in the standby database.
- Drop and recreate the object with the invalidated blocks, using the program that maintains the object.

Nologging Usage

Nologging is used in the following situations in the Oracle E-Business Suite:

- Building new objects during patch application, where use of nologging makes the initial build faster, and the downtime required for patching shorter.
- Changing the physical structure of existing objects during patch application (such as partitioning a table), where use of nologging reduces the time needed for the operation itself, and consequently the overall downtime.
- Certain specialized tasks where logging is not required, such as manipulating data for data warehousing applications, or maintaining summary data for business intelligence queries.
- Certain concurrent manager jobs. In most such cases, the object affected by nologging will be dropped at the end of the job, and the invalidated blocks cleaned up. If a recovery is needed while concurrent jobs are in progress, re-running the affected jobs will clean up any invalidated blocks that may exist.

Actions Needed

To monitor nologging activity in your environment, you should periodically query your production database to identify any datafiles that have experienced nologging operations. You should also run the query before and after applying an Applications patch, to determine whether any nologging activity was carried out.

A suitable query can be run via monitoring software such as Oracle Enterprise Manager. Alternatively, you can construct a query based on the *unrecoverable_change#* and *unrecoverable_time* columns of the data dictionary view *v\$datafile*. These are updated every time an unrecoverable or nologging operation marks blocks as invalid in the datafile.

The results of a query can be saved as a snapshot and compared to the last snapshot. You can then identify each occasion when nologging operations have been carried out in the database, and hence when you need to refresh backup datafiles with new copies that will be usable in the event of restoration being needed.

Note: See *Oracle Data Guard Concepts and Administration Release 2 (9.2)* for further details of nologging operations and associated tasks.

Disaster Recovery

A significant problem that strikes an Oracle E-Business Suite installation could put the viability of the organization at risk. Such a problem could be:

- An external disaster, such as a fire at a company's data center, resulting in a loss of service that severely hampers the organization's ability to do business.
- An internal disaster, such as a serious error by a privileged user, resulting in major loss or corruption of data.

This section gives an overview of the area of disaster recovery, which can be considered as the final component of a high availability strategy. Disaster recovery involves taking steps to protect the database and its environment to ensure that they can still operate in the face of major problems. Oracle provides features such as *Oracle Data Guard* to help set up and maintain a secondary copy of a database, typically referred to as a *standby database*. Such a standby database is brought into use after a *failover* from the primary database, when the primary becomes temporarily unavailable following a significant hardware or software problem of some kind, and service continuity cannot be maintained by a simple database restart.

Standby Environments and Release 11i

The secondary environment should be physically separate from the primary environment, to protect against disasters that affect the entire primary site. This necessitates having a reliable network connection between the two data centers, with sufficient bandwidth (capacity) for peak redo log traffic.

Other key requirements for setting up a standby environment include:

- Use of AutoConfig (see Chapter 5), which will always be the case in Release 11.5.10.
- Duplication of the primary environment's number and type of machines, including operating system versions. (The standby machines do not necessarily need to have the same capacity.)
- Identical top-level directory structures on the secondary server to those of the corresponding primary server, with ownership and permissions set appropriately.
- Identical disk directory structure on the secondary server and primary server, with the components in the same locations on the two servers. This ensures that the secondary environment allows patching to be done after it is put into use.
- The same operating system user and group IDs for the database and Applications accounts on the primary and standby servers.
- The same names for the production and standby databases on the primary and standby servers respectively.
- Oracle Net connectivity between the primary and secondary servers.
- The production database on the primary site running in ARCHIVELOG mode.

Using Standby Databases

Oracle Data Guard provides mechanisms for propagating changes from one database to another, to avoid possible loss of data if one site fails. The two main variants of a Data Guard configuration are *physical standby* and *logical standby*. Both of these use the primary database's redo logs to propagate production changes to the standby:

- *Physical standby* uses the normal database recovery mechanism to apply the primary database's redo logs to the standby database, resulting in an identical copy of the production database.

Physical standby is also valuable for protecting against the effects of *logical data corruption*, for example as a result of a serious data manipulation error by a

privileged user. Adding a time lag before the primary's redo logs are applied to the secondary provides a window in which the problem can be identified, and action taken to prevent the relevant redo logs from being applied to the secondary.

- *Logical standby* employs the *Oracle LogMiner* utility to build SQL statements that can be used to recreate changes made to the data. While the logical standby mechanism has the potential for use with a reporting database, it is not currently utilized with Oracle E-Business Suite.

Additional Decisions

In pre-11.5.10 releases that used an Oracle 8.1.7 database, there was a choice between using *mandatory archiving* and *optional archiving* to copy archived redo logs to the secondary site. Mandatory archiving meant that all archived redo logs were written to the primary and secondary sites at the same time. This had the disadvantage that an interruption of the network connection to the secondary site resulted in the primary database no longer archiving redo logs, since they had to be archived at both sites or neither. For this reason, many sites used optional archiving, which was more complex and required additional management steps.

Release 11.5.10's usage of an Oracle 9.2 database offers a more sophisticated solution, in which you can choose to maximize performance, protection, or availability:

- Maximizing *performance* is the default, where Data Guard manages the transport of archived redo logs to the standby site. It is similar to mandatory archiving in that the database manages the copying of archived redo logs to the secondary site as part of normal operation. However, it is different in that if network connectivity is disrupted, the archiving process at the primary site will continue to operate. When network connectivity is restored, the transport of archived redo logs (including those generated during the network outage) will be resumed.
- Maximizing *protection* involves writing online redo information to both the primary and standby systems, and does not mark a commit operation as successful until both the primary and at least one standby (if there is more than one) have the data to reconstruct the transaction. However, if network connectivity to the standby database fails, processing on the primary will halt.
- Maximizing *availability* makes a best effort attempt to write online redo information to a standby before returning success to a commit operation, but if network connectivity fails it will switch the instance to the next level of protection.

Finally, you must decide whether you need to copy concurrent manager log and output files to the secondary site. This will be determined by the relative importance of being able to review log files or re-run reports on the secondary site after a failover, compared to the overhead of keeping these files synchronized.

Oracle Data Guard and Nologging

Data Guard's reliance on the redo logs generated from the production database has significant implications for operations in which Oracle E-Business Suite uses the nologging feature (described previously) to perform some resource-intensive tasks with greater efficiency.

Where the nologging feature is used, for example in applying a patch, the redo logs do not contain enough data to make the corresponding changes on the standby database. You must therefore take steps to refresh the standby after a nologging operation, to ensure it will remain usable.

Introduction

Oracle E-Business Suite provides numerous options for building and tailoring an installation to meet specific business and technical requirements. At a simple level, this includes the capability to utilize varying numbers of machines to distribute the various Applications technology layers according to desired expenditure and required performance. There are also more specific factors, such as physical site organization and expected growth rate. Other aspects are the possible need to provide extra capability to cope with fluctuations in demand, and the possible need for resilience in the event of problems affecting some of the hardware components. A final (and important) consideration in planning a system is the need for the appropriate level of security to be put in place.

Many of these subjects are described in other chapters of this book. This section concentrates on how to make the decisions needed to balance the load on various components or layers, which is of particular importance for two reasons:

- Load balancing can involve the entire infrastructure of an E-Business Suite installation, with a change in one area potentially having significant effects elsewhere.
- Making informed decisions about load balancing can often enable a higher level of performance to be obtained without expenditure on additional hardware.

Load balancing areas include:

- Domain Name Server
- HTTP Server
- JServ Layer
- Forms Server
- Concurrent Processing Layer
- Database Layer

The emphasis here is on describing load balancing strategies and their key features, to allow an informed decision to be made regarding the applicability and usefulness of a particular area of load balancing in achieving the desired technical and business requirements. Both advantages and limitations of the different areas will be mentioned.

Categories of Load Balancer

With the exception of the Forms Metrics server, described in Chapter 1, load balancers are not installed as part of Oracle E-Business Suite. However, Oracle E-Business Suite can be configured for use with them as required.

The main categories of load balancer are:

Session Persistent Load Balancers - After a client's HTTP connection is established with a particular server, subsequent HTTP requests from that client are directed to the same server, for the duration of the session. This persistency is also referred to as *stickiness*.

Non-Session Persistent Load Balancers - These load balancers use a round-robin strategy for balancing incoming HTTP requests, and do not maintain session persistent client connections. After a client's initial HTTP connection is directed to a given server, subsequent HTTP requests from that client will not necessarily be directed to the same server.

Secure Sockets Layer (SSL) Accelerators - Secure Sockets Layer (SSL) accelerators can be used to reduce the SSL traffic and workload of the Web servers.

Usually, an SSL accelerator is the target for HTTPS browser requests, and thus the target for all client communication. It is responsible for converting HTTPS SSL requests to non-SSL HTTP requests, directing the subsequent request to the HTTP server (running in non-SSL mode). Before sending the response back to the client browser, the SSL accelerator converts the non-SSL requests back to SSL requests, in a reverse of the initial process.

Load Balancing Options

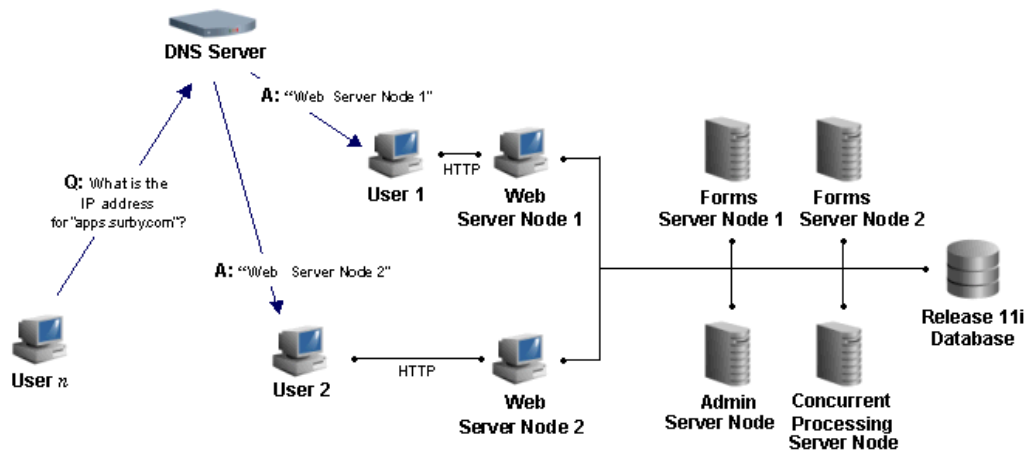
Several different types of load balancing can be employed with Oracle E-Business Suite. They range from the relatively simple to the more sophisticated, and should be chosen to integrate with the specific needs and existing infrastructure of a site.

Domain Name Server (DNS) Layer Load Balancing

This type of load balancer distributes end-user requests across multiple server nodes, based on dynamic assignments of IP addresses to a *fully qualified domain name*.

The following diagram shows an example of a configuration that uses DNS layer load balancing.

Figure 10-1 DNS Layer Load Balancing

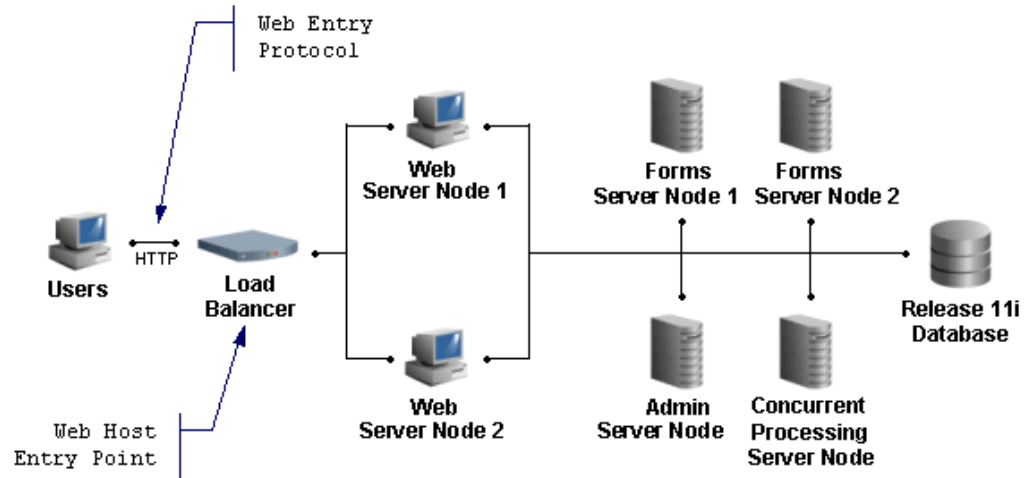


HTTP Layer Hardware Load Balancing

This type of load balancing can be used if you have a hardware load balancer that accepts HTTP communication and forwards it to a group of server nodes, sometimes referred to as a *farm*.

The following diagram shows an example of a configuration where an HTTP load balancer distributes the load across a farm consisting of two Web server nodes.

Figure 10-2 HTTP Layer Load Balancing



Hardware-based HTTP load balancers must be configured for persistent session connections for all traffic through all ports for Oracle E-Business Suite Release 11i environments. Prerequisite patches will need to be applied to all application tier server nodes.

JServ Layer Load Balancing

Oracle9i Application Server (9iAS) 1.0.2.2.2 introduced new capabilities for software-based load balancing of JServ transactions across multiple application tier server nodes, to increase the fault tolerance and scalability of large Oracle E-Business Suite installations.

In particular, 9iAS 1.0.2.2.2 uses the Oracle Process Manager module, *mod_oprocmgr*, to provide JServ death-detection and restart functionality across multiple nodes. For example, *mod_oprocmgr* may be configured to manage a JServ load balancing server pool comprising JVMs running on two different Web server nodes.

mod_oprocmgr starts the number of JVMs specified by the AutoConfig parameter *nproc*. All JVMs are started at the same time; there is no concept of starting JVMs in readiness, before they are actually required.

Also, there is also no capability for dynamically changing the number of active JVMs based on load. However, *mod_oprocmgr* monitors the JVMs via a heartbeat mechanism, killing and restarting any that become hung.

Important: If you are using an HTTP hardware-based load balancer that is capable of maintaining session-persistent client connections, you do not need to implement JServ layer load balancing.

Forms Layer Load Balancing

Two options exist for load balancing Forms-based traffic:

- JServ Load Balancing - for use with Forms Listener Servlet
- Metrics Server - for use with Forms server

Note: For additional information on load balancing configurations, see OracleMetaLink Note 217368.1, *Advanced Configurations and Topologies for Enterprise Deployments of E-Business Suite 11i*.

Setting Up Load Balancing

If you are creating a new installation of Oracle Applications, the 11.5.10 Rapid Install includes screens that allow you to set up Forms Metrics Server and JServ load balancing.

Additionally, you can use Oracle Applications Manager and AutoConfig to configure HTTP load balancing and JServ load balancing. From the Site Map, choose AutoConfig and then Configuration Wizards. This also gives you the option to enable or disable use of SSL, SSL Accelerator, and Forms Listener Servlet on your system.

Introduction

The foundation of security is *access control*, which refers to how the system is being accessed and by whom. User security consists of three principal components: *authentication*, *authorization* and an *audit trail*. Authentication validates the user's identity, authorization controls the user's access based on responsibilities assigned, and the audit trail keeps track of the user's transactions to ensure that the user's privileges are not being misused.

Authentication

Identifying and verifying who is allowed to access the system is the first line of defense. The most common approach is *password-based authentication*: if the legitimate user is the only one who knows the password, then whoever just entered the correct password is very likely to be the person authorized to use the account.

A number of practical problems can arise with passwords. These include:

- Passwords that are allowed to be too short, and thus vulnerable to being observed on entry
- Passwords that are forced to be too long, and which the user might decide to write down
- Easy-to-guess passwords, chosen as being easy to remember
- Rarely changed passwords
- Passwords that are used for multiple accounts

In a single-sign on environment (see Chapter 8), a single password allows access to more than one application, so the consequences of it being discovered or divulged are proportionately much more serious.

An attacker will generally focus on identifying the password of a powerful user such as a system administrator. Such users are generally more aware of security risks, and can be persuaded to take more care in their choice of password and to change it regularly. The Oracle E-Business Suite features various password management policies that can be enabled to secure key user accounts.

Authorization

On entering the system, the user should only be granted access to the features and specific data needed to perform his job. Routine access to highly sensitive data should

only be given to trusted users who need that level of access. The *Function Security* feature allows the System Administrator to manage the access privileges of individual users. By enforcing tighter security policies for more sensitive accounts, Function Security can mitigate the risk of unauthorized users' access to highly sensitive information.

Audit Trail

Even the most carefully planned user authentication and authorization policies cannot eliminate the risk of exploitation when the attacker is an authorized user. An *audit trail* can be used to keep track of a user's transactions to verify that the user is not misusing his access privileges. Oracle E-Business Suite can record details of every user's login, including time stamp, session ID, and information about the Function Security rules applying to that session. Information about the identity of the user is also attached to all transactions. This provides a method for detecting the party responsible for any transaction, or determining which users viewed sensitive data in a given time period.

If a valid user password has been compromised, and becomes known to an unauthorized person, it can be difficult to trace the intrusion back to the attacker. However, knowing the particular account that was used can help to identify other people who may have learned that user's password.

Network Security

An organization may or may not have physical control over the network infrastructure in use. The Internet is the best example of a network where it will not have control, and where extra steps must be taken to ensure security is not compromised.

A common concern regarding use of a public network such as the Internet is the possibility of someone eavesdropping on password transmissions by using a network sniffer. In such a case, though, the concern should be wider, and reflect the possibility of someone eavesdropping on sensitive information in general. In such cases, HTTPS (secure HTTP) connection to the E-Business Suite is recommended. All current browser-based password login screens send the password as a parameter in the HTTP form submission. Using an HTTPS connection will encrypt this information. The best practice is therefore to use HTTPS for all web-based access. On the other hand, if you have control over your network to the point where you can rule out eavesdropping, then password interception should not be an issue.

The main reason not to run HTTPS by default is performance, since it does introduce some overhead. A more strategic way to address this concern is to integrate the Oracle E-Business Suite with Oracle Application Server 10g Single Sign-On (SSO). Here, the SSO server that is responsible for user authentication is a different Web server from the one used with the E-Business Suite. Hence you can run the SSO server in HTTPS mode, while running the E-Business Suite Web server in the better-performing HTTP mode.

Oracle User Management

Introduced in Release 11.5.10, *Oracle User Management* is a secure and scalable system that enables organizations to define administrative functions and manage users based on specific requirements such as job role or geographic location.

Note: Use of the features provided by Oracle User Management is optional. They supplement, but do not replace, the standard user management and authorization mechanisms that exist in Release 11.5.10.

With Oracle User Management, instead of exclusively relying on a centralized administrator to manage all its users, an organization can create *functional administrators* and grant them sufficient privileges to manage a specific subset of the organization's users. This provides the organization with a more granular level of security, and the ability to make the most effective use of its administrative capabilities.

Oracle User Management implements several different layers of security, requiring organizations to specify:

- The set of users that will be granted access to specific areas of Oracle Applications
- The information these users will require to do their jobs
- The extent to which the users can use this information

Oracle's function and data security models constitute the base layers of this system, and contain the traditional system administrative capabilities.

Organizations can optionally add more layers to the system depending on the degree of flexibility they require. Role Based Access Control (RBAC) enables organizations to create roles based on specific job functions, and to assign these roles the appropriate permissions. With RBAC, administrative privileges and user access are determined by assigning individuals the appropriate roles.

Key features of RBAC include:

- **Delegated Administration** - Enables system administrators to delegate some of their administrative privileges to individuals that manage a subset of the organization's users.
- **Registration Processes** - Enable organizations to provide end-users with a method for requesting various levels of access to the system, based on their eligibility.
- **Self-service Requests and Approvals** - Enable end users to request initial access or additional access to the system by clicking on links embedded in a Web application.

Security Strategies

The Oracle E-Business Suite tables are no different from any other Oracle database tables, as far as a DBA is concerned, and the same security issues that apply to Oracle database installations also apply to E-Business Suite installations. While the Oracle database provides multiple mechanisms to ensure security, recovery, and high availability of databases, no amount of technology can completely protect against human problems (error or sabotage), or poor disaster recovery and corporate security policies.

Warning: A potentially disastrous security oversight is to retain the default password for a privileged account (for example, SYS/CHANGE_ON_INSTALL).

While technical measures exist to limit what a DBA can do, and selectively audit DBA activity, it is an axiom that DBAs serve in positions of trust. Organizations must therefore take appropriate steps to ensure that persons assigned to such positions are worthy of trust. For example, a company may go to great lengths to secure its trade secrets, such as locking formulae in bank vaults, obtaining patents, and performing extensive background checks on individuals privy to corporate secrets. A company's data is just as sensitive and valuable as its trade secrets; indeed, data should often be treated as the most closely-guarded secret. Therefore, the same checks should be made on DBAs as are made on staff given access to corporate secrets.

Warning: It is imperative that organizations follow appropriate procedures in hiring DBAs, since DBAs have responsibility for an organization's core, mission-critical data.

DBAs need to have extensive privileges to do their jobs; this also means that they can carry out destructive actions on a database, either by accident or intentionally. Such actions can directly and seriously impair an organization's ability to carry on with its business. For example, if your customer database is accidentally corrupted, and no backups have been made, you may lose vital customer information and not be able to fulfil orders.

Some organizations would prefer DBAs not to be able to see any of the actual data stored in the database. This is unrealistic, for both general and specific reasons:

- It is difficult (if not impossible) to prevent DBAs from seeing data in a database when the whole purpose of a database is to manage data. To put this another way, a manager who knew nothing about a group of employees could not be expected to manage them effectively, if at all.
- DBAs typically export and import tables into databases, create indexes, tune the database, and perform backups. By their nature, these operations cannot be performed (and their success confirmed) without the data being visible.
- Users making SYSDBA-type or SYSOPER-type connections to the database are not subject to the restrictions imposed by Oracle's *Virtual Private Database* (VPD) policies, or to the access control checks provided by *Oracle Label Security*.

Simply restricting the availability of the passwords for accounts with DBA privileges is an important first step in implementing security-oriented DBA access policies. This can then be extended by setting up procedures for *auditing* DBA actions, as described later in this section.

Data Encryption

For fields of especially sensitive information (such as credit card numbers), there is the possibility of *encrypting* the data stored. Currently, the Oracle E-Business Suite does not automatically encrypt data, the only exception being *iPayment*. There are several reasons for this:

- Different installations have different requirements for which data should be encrypted; for example, salary may be most confidential in a human resources database, whereas chemical formulae may be of greater sensitivity to a pharmaceutical company. Medical databases may have a number of fields that are confidential.
- While the underlying Oracle database does offer data encryption via the `DBMS_OBFUSCATION_TOOLKIT`, the keys cannot be stored in the database if a goal is to prevent the DBA from decrypting data. Also, encrypted data may be unrecoverable if the keys are lost.
- Encrypting columns has a relatively high cost in terms of resources, and encrypting numerous columns may impact system performance. It is therefore very important to be selective about the columns that are to be encrypted.
- Database-level encryption requires customized programming for each screen and report that accesses the data.

- Third party solutions that work in conjunction with the Oracle E-Business Suite can provide encryption of data where required.

Patching

Applying patches to the Oracle E-Business Suite requires the person performing the patching (normally, the DBA) to provide the passwords for both the SYSTEM and the APPLSYS accounts. Both of these accounts are highly privileged. It is possible to keep the passwords to these accounts secret until just before the patching process begins, at which point the passwords are changed to temporary values that are communicated to the DBA. When patching is complete, the passwords are set back to their previous, secret values. This procedure adds only a few minutes to the patching process, and can help improve security in cases where the DBA is, for example, located in a different country, but it must not be relied on for complete protection of data - the DBA could still carry out unauthorized actions, immediately before or after the patching operation.

Auditing DBA Activity

Rather than trying to *prevent* DBA access to data, it is more realistic to *audit* DBA access to data. Oracle 9i Release 2 added features that can be used to audit DBA actions. The audit trail, mentioned earlier, should not be used in place of procedures for hiring trustworthy DBAs, but can be a useful adjunct to them.

Oracle recommends that you strongly protect your audit trail records wherever they are stored. The most secure location for the audit trail, in terms of keeping the content from DBAs, is the operating system's own audit trail or operating system files. Oracle recommends that the database server should write audit records to the operating system, and the file to which Oracle writes audit records should have suitable operating system file protection.

Using an operating system audit trail requires a simple change of the AUDIT_TRAIL database initialization parameter from "DB" to "OS", and prevents privileged database users from reading, modifying or deleting audit records. However, this strategy is ineffective for users who have suitable operating system privileges. Also, the query advantage that SQL brings to audit analysis is lost, unless you have an operating system audit analysis tool that can read Oracle-generated operating system audit records.

Note: For further security recommendations, see OracleMetaLink Note 189367.1, *Best Practices for Securing the E-Business Suite*.

Network Topologies

Introduction

As large companies move to implement a global IT infrastructure, the choice of network topology becomes of increasing importance. This section describes the most significant strategic factors that can affect performance.

Strategies

A large, worldwide organization will typically benefit from the use of a "hub and spoke" network topology, with high-capacity links to regional hubs, and medium-capacity connections from the regional hubs to local offices. The locations of the regional hubs should be based on organizational need, carrier availability, pricing, and network latency. The routes and hops need to be as short and efficient as practicable. Network design for the Oracle E-Business Suite should be based around the needs of the majority of users; satellite users, for example, will normally be a small minority.

Note: For a discussion of the effects of different network layers on load balancing, see OracleMetaLink Note 217368.1, *Multi-Node Installations of iAS 1.0.2.2.2 for E-Business Suite 11i*.

Latency

Latency is the time for a packet to travel from its source to its destination, and is a key determinant of network efficiency. In general, Oracle E-Business Suite works very well with average latencies up to 300ms, and is usually found to give acceptable performance with latencies up to 500ms. Note that periods when forms are being loaded (for example, on startup) may be an issue in cases where latency is marginal. A consequence of this is that the newer HTML-based Applications (which do not use Forms) may give better performance than the traditional Forms-based Applications.

Satellite Links

Satellite links can be used with the Oracle E-Business Suite. They are considered to be just another network type, and may be the only choice for users in remote locations. In general, however, they should be employed only where use of terrestrial services is not feasible.

If satellite links are to be used, the network stack should be examined and tuned by a network specialist, to ensure device timeout settings, for example, are configured optimally. The goal is to achieve reliable operation, while maintaining an acceptable response time.

Wireless LANs

Wireless technology is becoming of increasing interest and use to some organizations. However, its deployment must be planned carefully. As well as the security aspects of wireless use, there are several technical considerations. For the Oracle E-Business Suite, the most important issue is the stability of the connection. It is not uncommon to experience dropouts (momentary loss of service) while using a wireless LAN. These may occur as a result of not having the latest firmware revision, or interference from devices that use a similar wavelength, such as cordless phones.

As far as supportability of wireless LANs goes, they are simply considered to be another network topology, and as such are neither supported nor unsupported. Hence it is feasible to run E-Business Suite client PCs over a wireless LAN. However, in the event of problems, it would be desirable to be able to determine whether the problem also occurs via a normal network link, i.e. whether the cause lies in the E-Business Suite or the network. Use of the Forms Listener Servlet architecture may be of benefit in a wireless LAN environment, as it is designed to attempt reconnection (via a configuration parameter) in the event of a network interruption.

Internationalization Support

Introduction

The Oracle E-Business Suite is designed for ease of deployment in a single global instance that meets the complex requirements of a worldwide enterprise. Strong internationalization support is essential in meeting this requirement. Key internationalization features provided by Release 11*i* include support for a wide variety of languages and territories, flexible date and number formats to suit local custom, multiple reporting currencies, and other country-specific functionality to provide compliance with local statutory requirements.

Applications Release 10.7 supported one language per instance. If you needed to run Oracle Applications in more than one language in the same instance, Oracle Consulting offered a customized solution to enable additional languages. Applications Release 11 supported the combination of multiple languages in the same database, provided that they used the same character set. Textual parts of Oracle Applications such as forms, reports, messages, help text, menu prompts, and lists of report names were available in all installed languages, but most data at the product level was still available only in the base, or primary, language. This meant, for example, that although you could enter payment terms only in the base language, forms could be displayed in another language. For additional multilingual support, Oracle Consulting continued to provide a customized solution.

The majority of Oracle Applications products have been restructured in Release 11*i* to provide multilingual support at the product data level, utilizing the *Unicode* character set. As Unicode supports all characters in common use in all of the world's modern languages, this removes the limitation on the number of supported languages that can be run in a single database. The additional functionality that was previously provided by Oracle Consulting is incorporated in Release 11*i* as standard.

Note: To learn about languages supported by the Oracle E-Business Suite Release 11*i*, see *Installing Oracle Applications*. To learn about character sets, see the *Oracle Globalization Support Guide*.

Languages and Character Sets on the Database Tier

By default, Rapid Install creates a production database with the US7ASCII character set, and a Vision demo database with the UTF8 character set. However, you can if desired choose any other supported character set during the installation. Rapid Install recommends a character set based on the languages you license.

Before installing Oracle Applications, you should carefully consider the future language requirements of your installation. The character set you choose during installation

determines which languages the instance can support. Review the *Oracle Globalization Support Guide* before choosing a character set. Changing character sets after installation is an involved process, and can be avoided by choosing a character set that will meet your long-term needs.

The US7ASCII character set only supports American English. All other character sets vary in the number of languages they support. For example, if you need to support the French language and also want to use the euro symbol, WE8ISO8859P15 is a superset of US7ASCII, supports both English and French, and contains the euro symbol. If you need to support English, French, Japanese, and Arabic, you must choose UTF8, Oracle's Unicode character set, as this is the only one that supports all four of these languages.

The extended multilingual support in the Release 11*i* data model may increase database storage requirements. For a new installation, consider the database space required for a single language and multiply this by the number of languages you will license. In the case of an upgrade, some of the data currently in a single language structure will be converted to a multilingual structure, which will require additional storage.

Note: For further details of installing character sets, see *Installing Oracle Applications*.

Using a multibyte character set such as the Unicode UTF8 or Japanese JA16EUC (as opposed to a single-byte character set such as WE8ISO8859P15) may also affect the overall space required for language setup and transaction data, because some characters used may require more than one byte of storage space.

Note: For further details of supported character sets, tips on choosing a database character set, and storage requirements, see the *Oracle Globalization Support Guide*.

Languages and Character Sets on the Application Tier

By default, Rapid Install creates the application tier file system for a production instance with the US7ASCII character set, and the file system for a Vision demo instance with the UTF8 character set. However, you can if desired choose any other supported character set during the installation. Rapid Install recommends the application tier character set based on the languages licensed.

To prevent data loss, character sets on all tiers must be compatible with each other. If one character set does not contain all characters in the other, replacement characters will be used and data lost as a result.

Note: As UTF8 is a superset of all other supported character sets, there are no other fully compatible character sets. If you use UTF8 on any tier, you must use UTF8 on all tiers.

Rapid Install installs American English on all servers in the application tier. Additional languages may also need be installed, so that all application tier servers have the same set of languages installed.

Character Sets on the Desktop Tier

Language support, which includes support for data input methods, character sets, and fonts, must be available on the desktop client. The character set of the browser is set by Oracle Applications for each session.

The desktop browser must support character set and language-specific capabilities. For instance, Hebrew and Arabic require bidirectional support for right-to-left display, and Arabic also requires a browser capable of special character shaping.

External Documents

External documents are those documents intended for customers and trading partners, such as bills of lading, commercial invoices, and packing slips. Oracle E-Business Suite Release 11i is capable of producing external documents in any of the active languages, simultaneously and with a single request. A company's customer in Italy, for example, could receive invoices printed in Italian, and a customer in Poland could receive invoices printed in Polish.

You can send external documents to different printers based on language, and route completion notifications to different people according to the requested language. For example, you could route all French documents to one printer, and all other documents to another printer. You could send completion notifications for Spanish documents to one user, and all notifications, including Spanish, to another.

Note: See the *System Administrator's Guide* for a list of external documents provided in Release 11i.

Globalization and Country-Specific Functionalities

Globalization is the process of designing and deploying software that meets the needs of a global enterprise. One requirement for successful globalization is to meet the statutory, legal, and cultural practices of a given locality. In Oracle E-Business Suite Release 11i, this is achieved through national and regional extensions called *country-specific functionalities*. Because country-specific functionalities are all compatible with each other, installation of all required country-specific functionalities results in a globalized implementation.

All country-specific functionalities are installed when you run the Rapid Install. You simply need to license those you wish to use. The functionality of each country-specific functionality is described in a special *User's Guide* for each country.

Dates and Numbers

You can enter and view dates in any valid format, such as 03-31-04, 31/03/04, or 2004-03-31. Any format for which SQL provides a mask is valid. The only exception is with Oracle Reports, which always uses the format DD-MON-RRRR, e.g. 31-MAR-2004.

You can also enter and view numbers with either the period (full stop) character or comma as the decimal separator. For example, you can enter 1.02 and 100,000.02, or alternatively 1,02 and 100.000,02.

Note: For further details of date and number formats, see the *Oracle Globalization Support Guide*.

Regardless of the various formats that may be used to enter dates and numbers, the actual values are stored in the database in uniform canonical formats. This allows date and number values to be entered in one format, and viewed in an alternative format by another user.

Multiple Reporting Currencies

The Multiple Reporting Currencies (MRC) feature, described in more detail in Chapter 15, allows you to report on and maintain accounting records at the transaction level in more than one functional currency. MRC is based on the Multi-Org Architecture (described in Chapter 14), and is a significant aspect of a globalization strategy.

The *primary functional currency* is the currency you use to record transactions and maintain your accounting data within the Oracle E-Business Suite. In the *primary set of books*, the functional currency is always the primary functional currency. Usually, the primary functional currency is the currency in which you perform most of your business transactions, and the one you use for legal reporting.

A *reporting set of books* is a financial reporting entity associated with a primary set of books. While the reporting set of books has the same chart of accounts and accounting calendar as the primary set of books, its use of a different functional currency (*reporting functional currency*) allows you to report in a different functional currency than that of your primary set of books.

You must define a separate set of books for each of your reporting functional currencies. For each set of books you use with MRC, you need to specify which is the primary set of books and which are the reporting set(s) of books. You then assign the reporting sets of books to the primary set of books. You must also define a primary responsibility to correspond to your primary set of books, and a reporting responsibility to correspond to each reporting set of books.

Note: For further details, see *Multiple Reporting Currencies in Oracle Applications* and *Multiple Organizations in Oracle Applications*.

NLS-Independent Application Servers

In Release 11, an application tier server was required for each language and territory configuration a user might have. For example, to process French and German forms requests, you needed to start one Forms server for French and one for German. If two users ran French, but one set the territory to France and the other to Switzerland, two Forms servers and two Reports servers would be needed to support the two users.

Release 11*i* removes this need for multiple servers. Application tier processes must be started with the character set of the server, as defined in Rapid Install. A user's NLS preferences (such as language, territory, date format, and number format) are passed with each user request to the application tier servers, where a session is started with the corresponding NLS settings.

NLS Settings

In Release 11*i*, user runtime NLS settings are stored as profile options in the database. The profile options for language and territory are configured at site level when running Rapid Install. The base language is used for the default language setting. The default user territory you choose is used for the territory profile option.

The site level profile option values provide the default NLS settings for all end users. Users inherit these values the first time they log on to Oracle Applications using the E-Business Suite home page. A user can continue to use the default values, or change any of the NLS settings to alternative values. The updated values are stored in the database at user level, and all future sessions are started with them.

Multi-Org Architecture

Introduction

The Oracle Multiple Organization Architecture (Multi-Org) provides support for multiple organizations in a single installation of Oracle E-Business Suite, with relationships you define. Multi-Org specifies how the different organizations interact, and how transactions flow between them. These organizations can be sets of books, business groups, legal entities, operating units, or inventory organizations.

Overview

When you run any Oracle Applications product in a Multi-Org environment, you first choose an organization, either implicitly by choosing a responsibility, or explicitly in a Choose Organization window. Each subsequent window and report displays information for the chosen organization only.

The following terms are fundamental to the Multi-Org architecture:

Table 14-1 Multi-Org Terminology

Set of Books	A financial reporting entity that uses a particular chart of accounts, functional currency, and accounting calendar. Oracle General Ledger secures transaction information (journal entries, balances) by set of books. When you use Oracle General Ledger, you choose a responsibility that specifies a set of books. You can then see information only for that set of books while logged in with that responsibility.
Operating Unit	An organization that uses Oracle Cash Management, Oracle Order Management and Shipping Execution, Oracle Payables, Oracle Purchasing, and Oracle Receivables. It may be a sales office, a division, or a department. An operating unit is associated with a legal entity. Information is secured by operating unit for these applications; each user sees information only for his operating unit. To run any of these applications, you choose a responsibility associated with an organization classified as an operating unit.

Note: For further details, see *Multiple Organizations in Oracle Applications*.

Multi-Org Partitioned Objects

Tables that contain Multi-Org data can be identified by the suffix "_ALL" in the table name. These tables include a column called ORG_ID, which partitions Multi-Org data by organization..

Every Multi-Org table has a corresponding view that partitions the table's data by operating unit. Multi-Org views partition data by including a DECODE on the internal variable CLIENT_INFO. This variable is set by the security system to the operating unit designated for the responsibility. It operates in a similar way to the LANGUAGE variable, which returns the language of the current session.

Note: If accessing data from a Multi-Org partitioned object when CLIENT.INFO has not been set (for example, from SQL*Plus), you must use the _ALL table, not the view.

SO_HEADERS_ALL, with its corresponding view SO_HEADERS, is an example of a Multi-Org partitioned object.

Note: For further details, see Technical Overview, *Multiple Organizations in Oracle Applications*.

Converting to Multi-Org

When you install a production E-Business Suite system, the data model is identical whether you implement Multi-Org or not. The Multi-Org views used to partition data are incorporated into the normal install, and use predefined text. When you convert to Multi-Org, tables are renamed to use the "_ALL" suffix to denote that they now contain Multi-Org partitioned data.

In Release 11i, the AD Administration utility is used for the initial conversion of seed data and transaction data to Multi-Org format. Note that the underlying data model is not changed. When you create a new operating unit, a concurrent program adds the appropriate seed data.

Note: For further details, see AD Administration, *Oracle Applications Maintenance Utilities*.

Multiple Reporting Currencies

Introduction

The Multiple Reporting Currencies (MRC) feature allows you to report and maintain accounting records at the transaction level in more than one functional currency. You do this by defining one or more *reporting sets of books*, in addition to your *primary set of books*.

In your reporting set(s) of books, you maintain records in a functional currency other than your primary functional currency.

Functional Currencies

The following terms are fundamental to using MRC:

Table 15-1 Functional Currencies and Sets of Books

Primary Functional Currency	The currency you use to record transactions and maintain your accounting data within Oracle Applications. The primary functional currency is generally the currency in which you perform most of your business transactions and the one you use for legal reporting.
Reporting Functional Currency	A currency, other than your primary functional currency, for which you need to report.
Primary Set of Books	A financial reporting entity in which you conduct business. The primary set of books uses a particular chart of accounts, accounting calendar, and functional currency (in the primary set of books, the functional currency is always the primary functional currency).
Reporting Set of Books	A financial reporting entity that is associated with a primary set of books. The reporting set of books has the same chart of accounts and accounting calendar as the primary set of books, but usually has a different functional currency. The reporting set of books allows you to report in a different functional currency than that of your primary set of books.

For each set of books you use with MRC, you need to specify which is the primary set of books and which are the reporting set(s) of books. You then assign the reporting sets of books to the primary set of books.

MRC Architecture

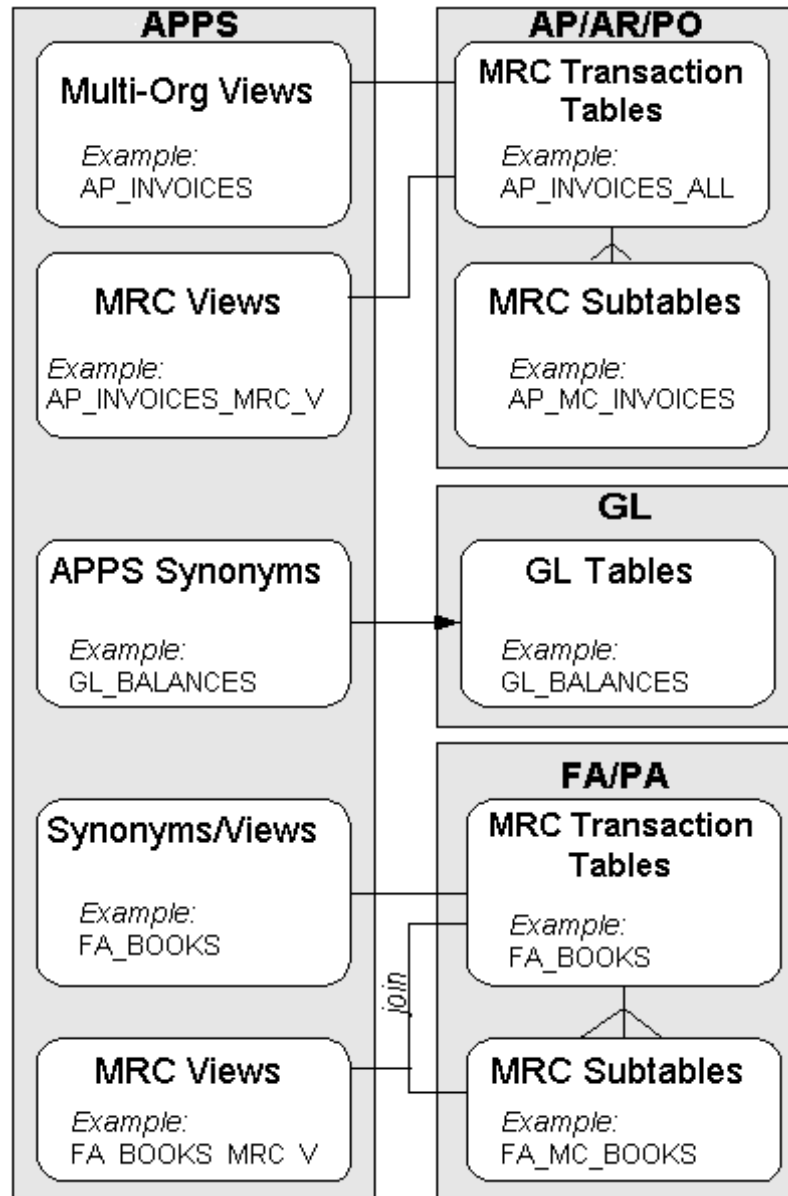
MRC is supported by the following Oracle E-Business Suite products:

- Oracle Assets
- Oracle Cash Management
- Oracle Cost Management
- Oracle General Ledger
- Oracle Global Accounting Engine
- Oracle Payables
- Oracle Receivables
- Oracle Purchasing
- Oracle Projects

Products that support MRC have an associated MRC subtable for each transaction table in the base schema that contains currency-sensitive information, such as conversion rate and transaction amount. Records in the base schema transaction table may have one or more MRC subtable records, which contain currency-sensitive information on the base transaction, in each associated reporting currency. In the APPS schema, MRC views provide reporting currency representations of each base transaction table.

Note: MRC can be used even if the Multi-Org Architecture is not implemented.

Figure 15-1 MRC Architecture

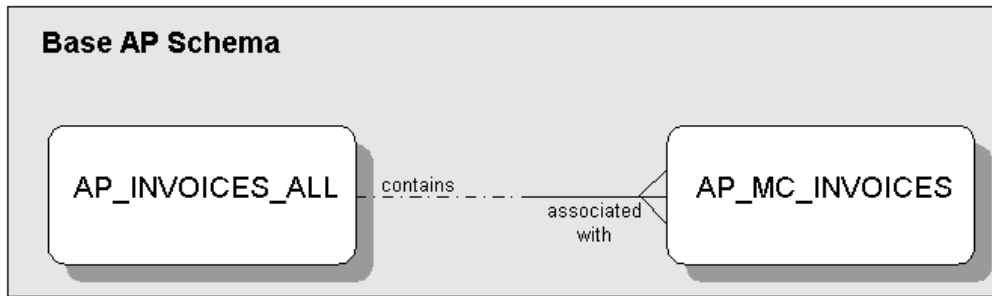


MRC Subtables

For products that support MRC, each transaction table in the base schema that contains currency-sensitive information, such as conversion rate and transaction amount, now has an associated MRC subtable.

There is an optional one-to-many relationship between records in a base transaction table and records in the corresponding MRC subtable. For each transaction record in the base transaction table, there may be one or more corresponding records in the MRC subtable. Each record is associated with a reporting currency. The primary key for an MRC subtable is the combination of the primary key of the base transaction table and the reporting set of books ID.

Figure 15-2 Transaction Table and MRC Subtable



MRC Triggers

As of Oracle Applications Release 11.5.9, all MRC triggers have been replaced by APIs. These MRC APIs are executed whenever the relevant records are inserted, updated, or deleted in the corresponding base product's transaction tables. For example, when you enter an invoice, Oracle Payables inserts records into `AP_INVOICES_ALL` and `AP_INVOICE_DISTRIBUTIONS_ALL`. When Payables inserts these records, MRC APIs are called from the corresponding table handlers to create reporting currency records in MRC subtables `AP_MC_INVOICES` and `AP_MC_INVOICE_DISTRIBUTIONS`.

Note: See *Multiple Reporting Currencies in Oracle Applications* for more information about the MRC subtables and their corresponding products.

MRC Views

MRC views provide representations of transactions in each reporting currency. They are created in the APPS schema. MRC views show information for only one set of books at a given time. That is, these views retrieve information based on the reporting set of books associated with the user's current responsibility. When you sign on to a responsibility, Oracle Applications sets the `userenv` variable based on the profile option *MRC: Reporting Set of Books*. This variable value is used by the MRC views to show only the information associated with your reporting set of books.

Glossary

Admin server

Sometimes referred to as the administration server, this is the server from which the system administrator runs programs that maintain and update an Oracle Applications database. For example, AutoUpgrade and AutoPatch are both run on this server to install, update, or upgrade the database that resides on the database server.

applet

A Java program that is downloaded to a desktop client from an HTTP server, and runs within a Java-enabled web browser. Applets are restricted in their capabilities compared to servlets.

See also: servlet

application server

Server that resides in an application (middle) tier, between the desktop clients and database tier. Desktop clients send their requests to an application server, which processes the request or send it to another server, such as the database server. The desktop clients never connect directly to the database server. The Forms server and Web server are types of application server.

See also: tier

applmgr

The account used to install and upgrade Oracle Applications. This account owns the Oracle Applications product files.

APPS schema

An ORACLE schema that has access to the complete Oracle Applications data model.

audit trail

A record of every user's login, including time stamp, session ID, and transactions carried out in the session. Can be used to identify unauthorized access and activities.

AutoConfig

A configuration management tool for an Oracle Applications environment, AutoConfig includes a number of scripts and other files that simplify the process of making updates to a system.

AutoPatch

A program that is used to apply patches and add new languages and products to an Applications system.

AutoUpgrade

The Oracle Applications upgrade program.

Automatic Undo Management (AUM)

Introduced in the Oracle9i database server, automatic undo management is based on the use of *undo tablespaces* rather than *rollback tablespaces* and *rollback segments*.

See also: rollback tablespace, rollback segment, undo tablespace

background process

A noninteractive process that runs in an operating system environment and performs a specific task.

bandwidth

The amount of data that can be sent through a network connection in a fixed period of time, usually measured in bits per second (bps). The speed and capacity of a network depend on both bandwidth and latency.

See also: latency

base language

The language used for seed data and setup data for tables that are not structured for multilingual support.

BC4J

An acronym for Oracle *Business Components for Java*, BC4J is a 100% Java-compatible, XML-based framework designed to facilitate the development and portable deployment of multi-tier database applications.

browser

See: Web browser

CBO

See: cost-based optimizer

certificate file

Contains the identity of a trusted source that the desktop client uses to guarantee the authenticity of a JAR file. Information within the certificate file allows the desktop client to decrypt the digital signature of the JAR file. The identity must be validated successfully before the desktop client downloads and executes the JAR file.

See also: digital signature, JAR file

CGI

Acronym for *Common Gateway Interface*, a specification for transferring information between a Web server and a program designed to process data that conforms to the CGI specification. Such *CGI programs* are a very common way to enable a Web server to interact dynamically with users.

See also: Perl

character set

A set of encoded binary values that represents the letters, numerals, and punctuation marks of a language, or of a group of languages that use similar written symbols. For example, the WE8ISO8859P1 character set can be used by English and many other languages that use a Latin-based alphabet and Arabic numerals. Terminals and printers

handle text data by converting these encoded values to characters. A character set may also be called a *codeset*.

client/server architecture

A configuration in which one or several servers perform database processing or other functions for applications that are run on clients. Software must be installed on each client before the client can interact with the servers. Client/server architecture is not used by the Oracle E-Business Suite in Release 11i.

clone

A copy of an Oracle Applications system, typically used for testing purposes. A clone can be created by using the Rapid Clone tool, or from Oracle Applications Manager.

codeset

See: character set

command

An instruction or request for the system to perform a particular action. An entire command may consist of the command name, plus one or more parameters and qualifiers.

command file

A file containing a predefined sequence of commands to be executed by the operating system.

Common Gateway Interface

See: CGI

concurrency

Simultaneous access of the same data by multiple users.

concurrent manager

A process manager that coordinates the processes generated by users' requests to run various data-intensive programs. An Oracle Applications product group can have several concurrent managers.

See also: internal concurrent manager

concurrent process

A task run by a concurrent manager. A concurrent process runs simultaneously with interactive functions and other concurrent processes.

Concurrent Processing server

An Oracle Applications server that runs time-consuming, noninteractive tasks in the background.

concurrent queue

A list of concurrent requests awaiting completion. Each concurrent manager has its own queue of pending requests.

concurrent request

A request issued to the Concurrent Processing server when you submit a noninteractive task, such as running a report.

consolidated update

A collection of recommended patches and rollups for a particular maintenance release, consolidated into a single patch that is installed immediately following use of Rapid Install or application of a maintenance pack.

See also: maintenance pack, Rapid Install

cost-based optimizer

Oracle database server component that determines the optimum execution path for an SQL statement by considering statistical information for the tables and indexes that the SQL statement will access. The cost-based optimizer (CBO) also considers hints, which can be employed to suggest a particular access path. In Oracle Applications Release 11*i*, cost-based optimization supersedes the rule-based optimization used in earlier releases.

See also: hint

customization

Enhancements to an Oracle Applications system made to fit the needs of a specific user community.

data dictionary

A set of Oracle database tables and views that contains administrative information about users, data storage, and privileges. It is created and maintained automatically.

database

A collection of data, stored in tables, and objects, such as stored procedures and triggers. The term can also refer to the software used to create, store, and manage this data, such as the Oracle database server.

See also: Database server, Oracle database

database administrator (DBA)

The person who prepares the Oracle database server and Oracle tools for an installation or upgrade of Oracle Applications, and performs maintenance on them after the installation. The DBA has highly privileged access to the database via the SYSTEM and SYS accounts.

database instance

The combination of background processes and memory used by an Oracle database. Often simply called an *instance*, and used to refer to a running Oracle database system. There is always a one-to-one correspondence between an Oracle instance and a system global area (SGA).

See also: SGA

database object

A logical entity created and stored in a database. Tables, views, synonyms, indexes, sequences, stored procedures, materialized views, and triggers are all examples of database objects.

Database server

[1] The Oracle database that constitutes the database tier of an Applications system. The database server stores the data maintained by Oracle Applications, and processes SQL and Java requests from Concurrent Processing servers, Forms servers, and Web servers.

[2] The machine on which an Oracle database resides.

See also: Oracle database

database space

The amount of disk space used by a set of database objects.

dbc file

A configuration file that contains information required to connect to the database.

demonstration product group

A product group that includes predefined transaction data for Oracle Applications products. It is used primarily for system testing and user training.

See also: product group

dependent product

An Applications product that is not licensed, but whose files are shared in part by a fully installed Applications product. A dependent product is also known as a *shared product*.

desktop client

A computer that sends user requests to the Web server and handles responses such as screen updates, popup lists, graphical widgets, and cursor movements.

diagnostic patch

A special patch used to gather additional information needed by Oracle to resolve a problem.

digital signature

A means of guaranteeing the authenticity of a program or collection of data, such as a JAR file. It is typically an encrypted message that contains the identity of the code's author.

See also: certificate file, SSL

Discoverer server

An application server that hosts Oracle Discoverer. The Discoverer server complements the Reports server, by allowing Applications users to perform ad hoc queries and analyze the resulting query output.

See also: Reports server

distributed concurrent processing

See: parallel concurrent processing

distributed directory structure

Applications product files installed in more than one file system, as when there is insufficient disk space in a single file system for all Applications product files.

DMZ

The demilitarized zone (DMZ) is the area between outer and inner firewalls, used to protect servers from attempts at unauthorized access to a network or intranet.

See also: firewall

environment file

A command file that sets environment variables. Only servers running UNIX use environment files; Windows servers use the Windows registry instead.

See also: registry

environment setting

A parameter that controls the behavior of Applications and Applications programs for your installation. Environment settings are stored as environment variables on UNIX servers and as registry keys or environment variables on Windows servers.

environment variable

A variable maintained by the UNIX shell that can be referenced by any program running within the shell. Environment variables hold values used by many Oracle programs and utilities.

On Windows, a string consisting of environment information, such as a drive, path, or filename, associated with a symbolic name. You can define environment variables using the System applet in the Control Panel or from the Windows command prompt.

export utility

An Oracle database server utility used to write database data to operating system files external to the database. The corresponding *import* utility can subsequently be used to read the data back into the same database, or a different one.

See also: import utility

extension

The part of a filename (suffix) after the dot, used to indicate the type or purpose of the file. For example, the extension *.sql* denotes a SQL*Plus script.

extent

A specific number of contiguous data blocks that store a specific type of information. A *segment* is made up of a number of extents.

See also: segment

family consolidated update patch

All upgrade-related patches consolidated from all products within a product family. Family consolidated update patches are released as needed and are only available for upgrading to Release 11*i* from Release 10.7 or 11.0.

family pack

An aggregation of patches at the product family level. Product family codes end in *_PF*. Successive family packs can be identified by the alphabetical suffix; for example, *11i.HR_PF.D*, which supersedes *11i.HR_PF.C*. Family packs are cumulative, each containing the fixes and enhancements that were in its predecessors.

failover

The utilization of an alternative component in a computer system to allow processing to continue after a similar component fails.

firewall

A computer that acts as an intermediary to protect a network from unauthorized access, by examining requests and countering suspicious access attempts. A firewall is often used in conjunction with a *proxy server*.

See also: proxy server

form

A related collection of fields, regions, and graphical components that appears in a single window. Oracle Applications Forms resemble paper forms used to run a business. You enter data by typing information into the form.

Forms client

A Java applet that runs on a desktop client and provides the user interface and interaction with a Forms server.

Forms server

A type of application server that hosts the Forms server engine. It mediates between the desktop client and the database, providing input screens for the Forms-based products on the desktop client and creating or changing database records based on user actions.

functional currency

In a Multiple Reporting Currencies (MRC) environment, a currency other than your primary currency.

Generic Service Management (GSM)

A fault-tolerant framework for generic service processes. GSM utilizes a central management console built into Oracle Applications Manager.

See also: Oracle Applications Manager

Gigabyte (GB)

A unit of memory or disk space equal to 1,073,741,824 bytes. One Gigabyte is equal to 1,024 Megabytes. Often rounded to 1,000,000,000 bytes, i.e. a billion bytes.

GSM

See: Generic Service Management

GUI (Graphical User Interface)

An interface used with personal computers and workstations that allows the user to access fields and regions of the screen with a pointing device, typically a mouse.

hint

An optimization suggestion placed in a Comment of an SQL statement.

See also: cost-based optimizer

HTML

An acronym for *HyperText Markup Language*, a simple predefined markup language used to format documents for viewing with a Web browser.

See also: XML

HTTP

An acronym for *HyperText Transfer Protocol*, the network protocol used for communication between an HTTP server and a Web browser, to determine the actions Web servers and browsers take in response to various commands.

See also: HTTPS

HTTPS

A version of HTTP that provides additional features needed for secure data transmission.

See also: HTTP

HTTP listener

A program on an HTTP server that accepts and processes incoming HTTP requests from Web browsers.

HTTP server

An application server that runs an HTTP listener, and sends out Web pages in response to HTTP requests from remote browsers. *Web server* is often used as a synonym.

See also: HTTP listener, Web browser

import utility

An Oracle database server utility used to read operating system files written by the *export* utility. Can be used to restore data into a database or transfer data to another database.

See also: export utility

index

A database object associated with a table, used by the Oracle database server to locate rows of that table quickly.

initialization parameters

Parameters defined in an initialization file that configure an Oracle database. The parameter settings affect many aspects of database activity, including performance. Generally, the parameter settings should only be changed under expert supervision.

instance

See: database instance

internal concurrent manager

A special concurrent manager process that monitors, controls, and dispenses requests to all other concurrent manager processes.

See also: concurrent manager

interoperability patch

A patch needed to enable use of Oracle Applications products with a newer version of the technology stack, for example to enable integration with Oracle Single Sign-On.

intranet

A network of computers that are internal to an organization. An organization's intranet can be set up to communicate with the globally-distributed Internet, using appropriate security measures such as firewalls and demilitarized zones (DMZs).

See also: DMZ, firewall

J2EE

An acronym for *Java 2 Platform, Enterprise Edition*, J2EE is a development environment that facilitates development, deployment, and management of multi-tier enterprise level applications.

Java

A computer language used to produce programs that can be downloaded and run on a desktop client using a Web browser. It is also used to produce platform-independent programs that run on a server, either interactively or when invoked through a request from a Web browser.

See also: applet, servlet

JavaServer Page

See: JSP

JAR (Java ARchive) file

A collection of Java *classes* compressed into a file for faster download to a desktop client. There are several specialized types of JAR file.

See also: Java class

Java class

Components of a Java program that define objects and operations performed on objects. Also identifies an operating system file that contains a program or part of a program written in Java.

JDBC (Java Database Connectivity)

A Java programming interface that enables Java programs to access the Oracle database server.

JServ

A *JServ* process is a specialized servlet designed for communication with the Oracle HTTP Server.

See also: Oracle HTTP Server, servlet

JSP

Acronym for *JavaServer Page*, an extension to Java servlet technology. JSPs are compiled into servlets when first requested, and can receive input from a servlet or send output to a servlet. Their dynamic scripting capability allows page logic to be separated from page display, giving greater flexibility in processing HTTP requests.

See also: servlet

JVM (Java Virtual Machine)

A runtime environment that interprets (translates) a compiled Java program, supplied in the form of *bytecode*, to machine code. Each operating system's JVM translates bytecode to instructions that can be executed by the CPU. Java bytecode is therefore executable by any JVM running on any machine. A Java-enabled Web browser has an internal JVM that allows it to execute applets or applications written in Java.

LAN (Local Area Network)

A limited-distance, high-speed, data communications network that allows various data processing resources to be connected and shared. A LAN is a network contained within a single physical site (one or more buildings), as opposed to a Wide Area Network (WAN).

See also: WAN

latency

In networking, the time a packet of data takes to travel from its source to destination. A measure of the speed of a network.

See also: bandwidth

LDAP

An acronym for Lightweight Directory Access Protocol, LDAP allows clients to access information from a directory server in which corporate directory entries are arranged in a hierarchical structure that can be used to reflect geographical or other organizational boundaries.

See also: Oracle Internet Directory

load balancing

The mechanism for distributing tasks to the least-busy server of the servers that are handling a particular workload. For example, using load balancing, the HTTP server connects the Forms client to the Forms server that has the lightest load. This server is called the *least loaded host*.

LOCAL

A Windows-specific environment setting that identifies the network alias of an ORACLE instance running on the local machine or on a networked machine. This variable overrides any setting for ORACLE_SID and causes the Oracle Net software to manage the connection request.

See also: ORACLE_SID, TWO_TASK

locally managed tablespaces

Tablespaces that are not managed from the data dictionary. They offer a choice of extent management, allowing extent sizes either to be determined automatically by the

system, or for all extents to be made the same size. This provides greater flexibility than the traditional dictionary-managed tablespaces.

See also: Oracle Applications Tablespace Model

log in

Perform a sequence of actions that authenticates a user and establishes communication with an operating system or application such as the Oracle database server or Oracle E-Business Suite. Logging in also sets up appropriate characteristics for the session, based on the user's particular privileges.

MAC address

A hardware address that uniquely identifies a node on a network, specifically in the Media Access Control (MAC) sub-layer of the Data Link layer of the OSI Reference Model. The other sub-layer of the Data Link layer is the Logical Link Control (LLC) layer.

materialized view

Schema objects that store the results of a specific query, enabling data to be summarized, precomputed, replicated, or distributed.

Megabyte (MB)

A unit of memory or disk space equal to 1,048,576 bytes (1024 x 1024). Often rounded to one million bytes.

Multiple Organization Architecture (Multi-Org)

A single installation of any Oracle Applications product that can support any number of organizations or different sets of books. The data contained in product schemas is for all organizations, and is partitioned by the ORG_ID column in tables.

Multiple Reporting Currencies (MRC)

An Oracle Applications feature that allows you to create, maintain, and report on accounting records at the transaction level in more than one functional currency.

Maintenance Mode

A mode of operation in which the Oracle Applications system is accessible only for patching activities.

See also: Restricted Mode

maintenance pack

An aggregation of patches for all products in the E-Business Suite. For example, the Release 11.5.10 Maintenance Pack contains all the latest code for all products when 11.5.10 was created. Maintenance packs are numbered sequentially such as 11.5.8, 11.5.9, 11.5.10, and are cumulative. A new *Rapid Install* program is normally released with a maintenance pack.

See also: Rapid Install

minipack

An aggregation of patches at the product level, for example Oracle Inventory. Minipacks are cumulative, a later one including all the fixes and enhancements in its predecessors. Sometimes referred to as a *patchset*.

NLS (National Language Support)

Oracle's National Language Support (NLS) allows you to store, process, and retrieve data in the language native to your users. It ensures that database utilities and error messages, sort order, date, time, monetary, numeric, and calendar conventions automatically adapt to the native language and locale.

node

[1] A networked machine with its own *MAC address*.

[2] In the context of Oracle E-Business Suite, a logical set of processes running on one hardware machine. Sometimes used as a synonym for *server*. Multiple nodes can be created on one machine, or nodes can be allocated their own dedicated machines. There is an increasing trend towards deploying multiple nodes on multiple machines, to increase fault tolerance and lower the cost of ownership. This is particularly true for the application tier.

See also: MAC address, server

OID

See: Oracle Internet Directory

one-off patch

A patch that addresses a single fix or enhancement. Such standalone patches are released only to meet an immediate need for a fix or enhancement that cannot wait until an aggregate bundling is available.

operating system

The computer software that performs basic tasks such as scheduling CPU time, allocating memory, and supervising communications between different computers.

ORACLE_HOME

An environment setting that specifies the top-level directory for Oracle database server program files.

ORACLE_SID

An environment setting that specifies the SID of an ORACLE instance. See also *TWO_TASK* and *LOCAL*.

See also: SID

Oracle Applications Manager

A sophisticated tool that enables system administrators to monitor and manage an Oracle Applications system from an HTML-based central control console.

Oracle Business Components for Java

See: BC4J

Oracle HTTP Server

Oracle HTTP Server is the Web server used by Oracle Application Server. Built on Apache Web server technology, Oracle HTTP Server supports Java servlets, JavaServer Pages (JSPs), perl, PL/SQL, and CGI applications.

See also: JSP, servlet

Oracle Applications Tablespace Model (OATM)

A new model that uses locally managed tablespaces to provide more flexible storage options than the dictionary-managed tablespaces used in the traditional tablespace model.

See also: locally managed tablespaces

Oracle database

The database management system used by Oracle Applications Release 11*i*. An Oracle database is made up of various different types of file.

See also: instance

Oracle JInitiator

A JVM used to support Forms-based Oracle Applications functions, JInitiator is downloaded and installed in the client browser when first needed.

Oracle Internet Directory

Oracle Internet Directory (OID) is a general purpose directory service for retrieval of information about users and resources, which utilizes LDAP in conjunction with the high performance, scalability, robustness, and availability of the Oracle database.

See also: LDAP

Oracle Net

The Oracle software that enables network connectivity between a client machine and the Oracle database server. Oracle Net manages communication sessions between these machines by opening and closing sessions, and by packaging and sending SQL statements and the corresponding database responses.

Oracle Portal

Part of the Oracle Application Server product, Oracle Portal is a tool for building customized corporate pages to provide a personalized view of selected applications and data (called a *portal*). Oracle Portal provides a customizable, easy to use HTML-based interface. Support tools for performance monitoring and security management are also included.

ORACLE schema

See: schema

Oracle Single Sign-On

An Oracle product that authenticates the username and password of a user attempting to access an application, and passes the user's identity to various applications as required.

See also: single sign-on

ORACLE user ID

A user name employed to access an ORACLE instance. Used in conjunction with a *password*.

Oracle Workflow

A product that facilitates automation of business processes, internally or externally routing information of any type according to easily-changed business rules.

Oracle XML Publisher

An Oracle Applications product that uses standard technologies and tools to facilitate the rapid and easy development of reports in the end-user's preferred format.

See also: XML

parallel concurrent processing

Distribution of concurrent processes amongst multiple Concurrent Processing servers. Also called *distributed concurrent processing*.

password

An identification word, associated with a username, that must be supplied to access an ORACLE instance or an Oracle Applications system.

Perl

An acronym for *Practical Extraction and Report Language*, Perl is an interpretive programming language that is particularly useful for processing text. As such, it has become one of the most popular languages for writing CGI scripts.

See also: CGI

platform

The type of computer system (hardware and operating system), on which application programs run. The hardware components include the central processing unit (CPU), which performs logic operations and data management, and the main memory (RAM and ROM). The software components include the *operating system*, the computer's coordinating program. Sun SPARC Solaris is an example of a platform; Sun SPARC refers to the hardware, and Solaris refers to the operating system. Some Oracle Applications functionality is platform-specific, meaning its behavior may vary between different platforms.

PL/SQL

A procedural extension of the SQL programming language that provides programming constructs such as blocks, conditionals, and functions.

portal

See: Oracle Portal

portlet

A reusable component that provides access to an information source, possibly summarizing the content. Portlets are the fundamental building blocks of an Oracle Portal page.

See also: Oracle Portal

primary functional currency

The currency used to record transactions and maintain accounting data within the Oracle E-Business Suite.

See also: reporting functional currency

primary set of books

A financial reporting entity in which business is conducted.

See also: reporting set of books

Pro*C/C++

An Oracle precompiler product that allows developers to embed standard database calls to an ORACLE database in C and C++ programs.

product group

A set of Oracle Applications product schemas linked together by a single Oracle Application Object Library schema.

provisioning

The transfer of user attributes from one repository (such as Oracle E-Business Suite) to another (such as Oracle Internet Directory). Provisioning is controlled by *provisioning profiles*.

proxy server

A secure means of allowing users behind a firewall to access external Internet resources. All requests from intranet users go via the proxy server rather than directly to the destination server. The proxy server then passes the returned information on to the client. All Internet traffic for a site thereby passes through a single, secure point.

RAC

See: Real Application Clusters

Rapid Install

The installer for Oracle Applications. Normally, a new version is released to accompany a maintenance pack. Using Rapid Install provides the same Applications code as could be obtained by applying the associated maintenance pack (such as Release 11.5.10) to an earlier release level (such as Release 11.5.8). It also provides the latest certified technology stack. In contrast, applying a maintenance pack provides Applications code only.

See also: maintenance pack

Real Application Clusters

Real Application Clusters (RAC) is a database clustering technology whose shared storage capabilities allow multiple machines to work in parallel on the same data, reducing processing time significantly. RAC also offers resilience, allowing processing to continue in the event of one or more machines being unavailable because of planned or unplanned downtime.

registry

On a Windows machine, a central repository that holds configuration information. During installation, Oracle Applications writes data to the registry. You can also edit the registry directly with the Registry editor (*regedt32.exe* or *regedit.exe*), though this should only be undertaken by an expert user.

See also: registry key, registry subkey

registry key

A folder that appears in the left pane of the Registry editor window. A key can contain *registry subkeys* and *value entries*.

See also: registry, registry subkey

registry subkey

A key within a registry key; analogous to a subdirectory in a file system. Subkeys are provided to carry out product-specific functions. Oracle E-Business Suite stores information about a product group in a registry subkey.

See also: registry, registry key

remote procedure call

A remote procedure call (RPC) is a protocol that enables a client to execute a program on a server. The client sends a message to the server with appropriate arguments, and the server returns a message containing the program's results.

report

A user-organized display of Oracle E-Business Suite information. A report can be viewed online or sent to a printer. The content of a report can range from a summary to a complete listing of values.

Report Review Agent

A tool used by Oracle Applications to view concurrent processing files online.

Reports server

A server that is automatically installed on the same node as the Concurrent Processing server. Reports generated by the Reports server are monitored and administered separately from concurrent processing reports.

reporting functional currency

A currency, other than the primary functional currency, for which reports need to be generated.

See also: primary functional currency

reporting set of books

A financial reporting entity that is associated with a primary set of books. It has the same chart of accounts and accounting calendar, but usually a different functional currency.

See also: primary set of books

reserved word

A word that has a special meaning to a computer program, for example END. Custom-built programs that integrate with Oracle Applications must not use reserved words.

responsibility

A collection of functions within Oracle E-Business Suite. Each user is assigned one or more responsibilities to allow access to the appropriate functions and data.

Restricted Mode

A mode of operation that allows Applications administrators to carry out privileged tasks such as viewing the progress of a patching session.

See also: Maintenance Mode

rollback segment

Oracle database object used to undo (roll back) changes to the database in the event of media recovery or request for transaction rollback. Superseded by *undo segment* in Oracle9i.

See also: undo tablespace

rollback tablespace

A tablespace created for rollback segments. Superseded by *undo tablespace* in Oracle9i.

See also: undo tablespace

rollup patch

An aggregation of patches at either a functional level (such as flexfields), or at a specific product or family release level (such as Oracle Marketing).

RPC

See: Remote procedure call

schema

An ORACLE account or ORACLE ID.

segment

A set of *extents* that contains all the data for a specific logical storage structure in a tablespace. Examples include the data segment for a table and index segment for an index.

See also: extent

server

[1] A process or group of processes that runs on a single machine and provides a particular functionality. For example, the HTTP server responds to HTTP requests. Sometimes used synonymously with *service*.

[2] A computer dedicated to a particular role. For example, a database server.

service

[1] A synonym for *server*.

[2] On Windows, a process that provides a particular operating system or application functionality, such as the telnet remote logon service. Used by Oracle E-Business Suite to support concurrent processing, for example.

servlet

A small Java program that runs on an HTTP server, rather than being downloaded to a desktop client and run as an applet in a Web browser environment. Servlets have greater capabilities and fewer restrictions than applets.

See also: applet, JSP

setup data

Company-specific configuration data, such as locations, freight terms, and payment terms. You create this data when initially configuring an Oracle Applications product.

SGA (System Global Area)

An Oracle-reserved section of memory that provides communication between database users and the ORACLE background processes.

shared product

See: dependent product

short name

An abbreviation for an Oracle Applications product (such as *po* for Purchasing).

SID

An acronym for *System Identifier*. The SID is used to distinguish between different Oracle instances that may be running on a system.

See also: ORACLE_SID

single sign-on

The infrastructure whereby a user need only be authenticated explicitly once, with subsequent connections to other applications being authenticated transparently to the user. SSO allows a user to access multiple accounts and applications with a single username and password.

See also: Oracle Single Sign-On

sizing factor

An integer that determines the growth rate, as a percentage of their defaults, for the database objects of an Oracle Applications product.

SQL (Structured Query Language)

An internationally standardized language that is used to access data in a relational database.

SQL script

A file containing SQL statements that you can run with a tool such as SQL*Plus to query or update ORACLE data.

SSL

An acronym for Secure Sockets Layer, SSL is a standard for the secure transmission of documents over the Internet using HTTPS. SSL uses digital signatures to check that transmitted data has not been tampered with.

See also: digital signature, HTTPS

subdirectory

A directory that is contained within another directory.

Support Cart

A component of Oracle Applications Manager. Support Cart is used to gather diagnostic information for use by Oracle Support.

See also: Oracle Applications Manager

synonym

An alias for a table, view, sequence, or program unit that masks the real name and owner of the object, provides public access to the object, and simplifies the writing of SQL access statements for the object.

syntax

The spelling and grammar of a computer language, which directs how commands, qualifiers, and parameters are organized to form valid commands.

SYS username

One of two standard DBA usernames automatically created with each database (the other is SYSTEM). SYS owns the base data dictionary tables and views.

See also: SYSTEM username

SYS.DUAL table

Owned by the SYS user, the DUAL table contains exactly one row. It is used as a "dummy" table in a SQL statement to return values that are not stored in tables, including constant values, evaluations of arithmetic expressions, or system values such as the current date.

system administrator

The person who manages administrative tasks in Oracle Applications, such as registering new users and defining system printers, using the System Administrator responsibility.

System Identifier

See: SID, ORACLE_SID

SYSTEM schema

See: SYSTEM username

SYSTEM tablespace

Holds data dictionary tables owned by the SYS account. It is created when you install the database.

SYSTEM username

One of two standard usernames automatically created with each database (the other is SYS). The SYSTEM username is the preferred username to use when performing database maintenance.

See also: SYS username

table

The basic unit of storage in a relational database management system. A table represents entities and relationships, and consists of one or more units of information (rows), each of which contains the same types of values (columns).

tablespace

An Oracle database storage unit that groups related logical structures together. Traditionally, one tablespace was needed for each Oracle Applications product's database tables, and another for its indexes. As of Release 11.5.10, this model

is superseded by the Oracle Applications Tablespace Model, which uses far fewer tablespaces.

See also: Oracle Applications Tablespace Model

TCP/IP

Acronym for *Transmission Control Protocol/Internet Protocol*, an industry-standard networking protocol used for communication between computers and related devices.

temporary tablespace

A tablespace used when a SQL statement requires the creation of temporary segments (for example, creation of an index).

three-tier architecture

A computing architecture where the components are separated into three layers, or tiers: the desktop (or client) tier, the application tier, and the database tier.

TWO_TASK

Under UNIX, an environment setting that identifies the network alias of an ORACLE instance running on the local machine or on a networked machine. This variable overrides any setting for ORACLE_SID and invokes the *Oracle Net* software to manage the connection request.

See also: LOCAL, ORACLE_SID

undo tablespace

Tablespace used to store *undo records*, which maintain read-consistency in the database during normal operation, and also play a key role in database recovery.

See also: rollback tablespace

URL

An acronym for *Uniform Resource Locator*, a URL is a text representation of the location of a resource available via an intranet or the Internet.

user ID

See: username

username

A name that identifies a user requesting access to a secure environment or program, such as an Oracle database or Oracle Applications system. Every username is associated with a password. In Oracle Applications, a username is normally associated with a *responsibility*.

See also: responsibility

view

A tailored presentation of data in one or more tables. A view can be thought of as a stored query.

WAN (Wide Area Network)

A communications network that connects geographically separated areas.

See also: LAN

Web browser

A program that runs on a desktop client, sending HTTP requests to a Web server to retrieve HTML pages and Java applets.

See also: HTML, HTTP, HTTP listener, HTTP server

Web server

See: HTTP server

XML

eXtensible Markup Language, a metalanguage that allows specialized markup languages to be designed to suit different types of document. XML differs from HTML in not being based on a predefined format.

See also: HTML, Oracle XML Publisher

Symbols

8.0.6 ORACLE_HOME, 2-11
9.2.0 ORACLE_HOME, 2-2
<CONTEXT_NAME>.env, 2-5

A

AD Administration, 1-18, 1-20
 adadmin, 2-9
 Change Maintenance Mode menu, 9-2
AD Merge Patch, 1-20
adadmin, 1-18
adautoCfg.sh, 5-2
adcfgclone.pl, 6-3
adcfginfo.sh, 5-9
adchkcfg.sh, 5-9
adconfig.pl, 5-2
adconfig.sh, 5-2
adconfig.txt file, 2-14
Admin server
 description, 1-18
adovars.env file, 2-9
adpatch, 1-18
adpreclone.pl, 6-2
adstpall.sh, 5-8
adstrtal.sh, 5-8
agents, 1-23
AK
 definition of, 1-20
ALR, 1-24
AOL/J, 1-6
APPCPNAM environment setting, 2-12
APPL_TOP, 2-4
 directory and upgrades, 2-5
 distributing across disks, 2-10
 environment setting in .env, 2-11
APPLCSF environment setting, 2-12
APPLDCP environment setting, 2-12
APPLFENV environment setting, 2-11
Application Object Library directory (fnd), 2-9
application tier
 components, 1-4
 definition, 1-1
 load balancing, 1-5

Applications Context, 2-5
Applications context, 5-1
Applications context file, 5-2
Applications Dashboard
 Oracle Applications Manager, 7-1
Applications patches, 1-18
Applications Utilities
 see AU, 1-21
Applications Utilities directory, 2-9
APPLLOG environment setting, 2-12
APPLOUT environment setting, 2-12
APPLPTMP environment setting, 2-12, 2-13
APPLTMP environment setting, 2-12, 2-13
APPS schema
 and base product schemas, 3-1
 benefits of using, 3-2
 custom schema access, 3-3
 data access, 3-3
 overview, 3-1
APPS<CONTEXT_NAME>.env, 2-11
AU, 1-21
AU_TOP environment setting, 2-12
audit trail, 11-2
auditing, 11-4
AutoConfig, 1-20
 adconfig.sh, 2-9
AutoConfig configuration file, 5-4
AutoConfig driver file, 5-4
AutoConfig template file, 5-4
automatic segment space management, 3-5
Automatic Undo Management, 3-6
automatic undo management, 3-6
AutoPatch, 1-18, 1-20
 adpatch, 2-9
 and Maintenance Mode, 9-2
AutoUpgrade, 1-18, 1-20
 adaimg, 2-9

B

base language, 13-1
BC4J, 1-6
Bookmarkable URLs, 8-2
Business Event System, 1-23

C

- cfgcheck.html, 5-9
- character sets
 - difficult to change, 13-2
 - multibyte, 13-2
 - on database tier, 13-1
 - on desktop tier, 13-2
- CLIENT_INFO, 14-2
- cluster, 1-1, 4-3
- COMMON_TOP directory, 2-3
 - contents, 2-3
- concurrent manager, 1-12
 - log and output directories, 2-3
- concurrent processing
 - controlling the size of files, 1-17
 - description, 1-12
 - display of output, 1-16
- Concurrent processing server
 - description, 1-12
- concurrent program library, 1-14
- concurrent requests
 - definition, 1-12
- Configuration Wizards
 - Oracle Applications Manager, 7-1
- conflict domain
 - Concurrent Processing, 1-13
- Conflict Resolution Manager
 - Concurrent Processing, 1-13
- consolidated environment file, 2-5
- context file
 - AutoConfig, 5-1
- context name, 5-1
- Context Value Management, 5-3
- copy driver
 - patch, 1-18
- core technology directories, 2-9
- cost-based optimization, 4-1
 - use of materialized views, 4-3
- country-specific functionality, 13-3
- customizations
 - custom schema access, 3-3
 - data access, 3-3
- CVM
 - See* Context Value Management

D

- Daily Business Intelligence, 1-18
- data definition
 - in XML Publisher, 1-25
- DATA directory, 2-2
- database context file, 5-2
- database driver, 1-18
- database home, 2-2
- database recovery, 9-8
- Database Resource Manager, 4-2, 4-2
- Database tier
 - description, 1-19

- database tier, 1-1
- date formats, 13-3
- DB directory, 2-2
- DBI, 1-18
- desktop tier, 1-2
 - definition, 1-1
- devenv.env file, 2-14
- dictionary-managed tablespaces, 3-5
- Discoverer End User Layer, 1-11
- Discoverer server
 - description, 1-11
- Distributed AD, 9-4, 9-7
- DNS Layer Load Balancing, 10-2

E

- E-Business Suite Home Page, 1-3
- encrypting, 11-4
- Entrust
 - third-party single sign-on service
 - integration with Oracle Application Server, 8-2
- environment files, 2-2, 2-10
 - devenv.env, 2-14
 - fdenv.env, 2-14
- environment settings, 2-1, 2-10
- environment setup files, 2-11
- external documents, 13-3

F

- failover, 9-10
- farm, 10-3
- file types, 2-7
- FND_TOP environment setting, 2-12
- fdenv.env file, 2-14
- FNDNAM environment setting, 2-11
- FO
 - See* formatting object
- formatting object, 1-24
- Forms client applet
 - description, 1-4
- Forms Listener Servlet, 1-9
- Forms server
 - architecture, 1-8
 - communication with database server, 1-8
 - communication with desktop client, 1-7
 - description, 1-7
 - load balancing, 1-8
- Forms service, 1-1
- FORMS60_MAPPING environment setting, 2-13
- function security, 11-2
- functional administrators
 - Oracle User Management, 11-3

G

- generate driver

- patch, 1-18
- Generic Service Management, 7-4
- global inventory
 - Oracle Universal Installer, 6-3
- grants
 - between user IDs, 3-1
- GRAPHICS60_PATH environment setting, 2-13
- GWYUID environment setting, 2-11

H

- hints, 4-1
- HTTP Hardware Load Balancing, 10-3
- HTTP server, 1-1
- httpd.conf, 5-4
- HTTPS, 11-2

I

- iAS directory, 2-2
- iAS ORACLE_HOME, 2-11
- immediate program
 - Concurrent Processing, 1-12
- instance recovery, 9-8
- instantiation, 5-4
- integration hub
 - Oracle Application Server 10g role as, 8-5
- Internal Concurrent Manager
 - definition, 1-13
- Invoker Rights, 4-2, 4-2

J

- JAR files, 1-4
- Java Archive files
 - see JAR files, 1-4
- Java Server Page (JSP), 1-6
- Java servlet, 1-6
- Java servlet engine, 1-6
- JAVA_TOP environment setting, 2-4
- JInitiator
 - see Oracle JInitiator, 1-4
- JServ Layer Load Balancing, 10-3
- JServ Load Balancing, 1-10
- JSP files, 2-4

L

- language support, 13-1
 - in earlier releases, 13-1
- LDAP, 8-2
- License Manager, 7-2
- Lightweight Directory Access Protocol, 8-2
- Link-on-the-fly support, 8-2
- load balancer
 - categories, 10-2
- load balancing, 1-5
 - Forms server, 1-8

- local inventory
 - Oracle Universal Installer, 6-3
- locally managed tablespaces, 3-5
- log files, 2-3
- logical data corruption, 9-10
- logical standby, 9-10

M

- Maintenance Mode, 9-2
- maintenance packs, 1-18
- Manage Downtime, 7-3
- mandatory archiving, 9-11
- materialized views, 4-3
- merged APPL_TOP, 9-3
- metadata
 - in Oracle Discoverer, 1-11
- MetaData Services, 1-21
- Metrics server
 - Forms load balancing, 1-8
- Microsoft Active Directory
 - LDAP V.3 directory
 - integration with Oracle Application Server, 8-2
- mini-packs, 1-18
- mod_oprocs
 - and JServ functionality, 10-3
- mod_oprocmgr, 1-10
- MRC, 13-4
- MRC (Multiple Reporting Currencies)
 - overview of architecture, 15-2
 - transaction table/subtable illustration, 15-4
 - triggers, 15-4
 - views, 15-4
- Multi-Org, 14-1
- multilingual support, 13-2
- multiple ORACLE_HOMEs, 2-2
- Multiple Organization Architecture, 14-2, 14-2
 - CLIENT_INFO, 14-2
 - ORG_ID, 14-2
 - overview, 14-1
 - partitioned objects, 14-2
 - patching, 14-2
 - SO_HEADERS_ALL, 14-2
- Multiple Reporting Currencies, 13-4
- Multiple Sets of Books Architecture (MSOBA), 4-2

N

- National Language Support
 - see NLS, 13-2
- Netegrity
 - third-party single sign-on service
 - integration with Oracle Application Server, 8-2
- NLS
 - default date and number formats, 13-4
 - language profile option, 13-4

- profile options, 13-4
 - territory profile option, 13-4
- NLS and application servers
 - in earlier releases, 13-4
 - in Release 11i, 13-4
- NLS settings, 13-4
- NLS settings and the Personal Home Page, 13-4
- NLS_DATE_FORMAT environment setting, 2-12
- NLS_LANG environment setting, 2-12
- NLS_NUMERIC_CHARACTERS environment setting, 2-12
- node, 1-1
- nologging, 9-8
- notifications, 1-23
- number formats, 13-3

O

- OA_HTML environment setting, 2-4
- OAM, 7-1
- OAM Support Cart, 7-4
- OATM
 - See* Oracle Applications Tablespace Model
- OATM Migration Utility, 3-6
- OID, 8-4
- OLTP, 1-11
- one-to-many-support, 8-2
- online transaction processing
 - see* OLTP, 1-11
- operating unit
 - definition of, 14-1
- optional archiving, 9-11
- Oracle Alert, 1-24
- Oracle Application Object Library, 1-21
- Oracle Application Server, 2-2
- Oracle Applications Architecture, 1-1
- Oracle Applications Framework, 1-5
- Oracle Applications Manager, 7-1
 - and Maintenance Mode, 9-2
- Oracle Applications Tablespace Model, 3-4
- Oracle Applications technology layer, 1-20
- Oracle Common Modules, 1-20
- Oracle Data Guard, 9-10
- Oracle Discoverer 10g, 1-11
- Oracle Enterprise Manager, 7-1
- Oracle General Ledger
 - directory structure, 2-5
- Oracle HTML-Based Applications
 - Java servlet access, 1-6
- Oracle HTML-based Applications
 - files, 2-4
- Oracle HTTP server, 1-5
 - components of, 1-5
- Oracle Internet Directory, 8-4
- Oracle JInitiator
 - description, 1-4
- Oracle Label Security, 11-4
- Oracle LogMiner, 9-11

- Oracle Net, 1-8
- Oracle Portal, 8-4
- Oracle Single Sign-On, 8-4
- Oracle Universal Installer, 6-1, 6-3
- ORACLE user ID
 - overview, 3-1
- Oracle User Management, 11-2
- Oracle Workflow Builder, 1-23
- Oracle Workflow Event Manager, 1-23
- Oracle XML Publisher, 1-24
- Oracle9i
 - Database Resource Manager, 4-2
 - Invoker Rights, 4-2
 - partitioned tables, 4-2
- Oracle9i Application Server, 8-1
- ORACLE_HOME
 - 8.0.6, 2-2
 - 9.2.0, 2-2
 - database, 2-2
- ORG_ID, 14-2
- output files, 2-3
- overview pages, 1-18

P

- Parallel Concurrent Processing, 1-17
- partitioned objects
 - in Multi-Org, 14-2
- partitioned tables, 4-2
- partitions, 4-2
- partner applications, 8-9
- password-based authentication, 11-1
- patch driver files, 1-18
- Patch Wizard, 7-3
- PATH
 - environment setting, 2-12
- PDF, 1-24
- Personal Home Page and NLS settings, 13-4
- phases
 - of AutoConfig operation, 5-7
- physical standby, 9-10
- PL/SQL output files
 - location, 2-13
- PLATFORM environment setting, 2-11
- port pool
 - choice when cloning, 6-3
- Portable Document Format, 1-24
- primary functional currency
 - definition of, 15-1
- primary set of books, 15-1
 - definition of, 15-1
- product top directory
 - see* <PROD>_TOP, 2-6
- profile, 1-22
- provisioning, 8-4
- provisioning integrated application, 8-7
- provisioning profiles, 8-7, 8-7

Q

queue handler, 1-24

R

RAC

See Real Application Clusters

Rapid Clone, 1-20

Rapid Install, 1-20

Rapid Install Portal, 2-4

RBAC

See Role Based Access Control

Real Application Clusters, 1-2, 4-3

Report Review Agent, 1-16

reporting functional currency

definition of, 15-1

reporting set of books

definition of, 15-1

Reports server

architecture, 1-10

clustering and load management, 1-11

description, 1-10

language selection, 1-10

language support, 1-11

REPORTS60_TMP environment setting, 2-13, 2-13

responsibility, 1-22

restore.sh, 5-8

Restricted Mode

and Maintenance Mode, 9-2

Rich Text Format

document format, 1-25

Role Based Access Control, 11-3

RTF, 1-25

S

Secure Sockets Layer, 7-1

seed data

tablespace management and, 3-4

server, 1-1

service, 1-1

services, 7-4

servlet engine, 1-6

in Oracle HTML-based Applications, 1-5

set of books

definition of, 14-1

Shared APPL_TOP, 1-1, 9-2

Shared Application Tier File System, 9-3

SID, 5-1

single sign-on, 8-1

Site Map, 7-1

SO_HEADERS_ALL, 14-2

source system

cloning, 6-1

spawned program

Concurrent Processing, 1-14

split configuration, 1-19

SSL, 7-1

staged APPL_TOP, 9-5

Standard Manager

Concurrent Processing, 1-13

standby database, 9-10

subscriptions, 1-23

SunONE/iPlanet

LDAP V.3 directory

integration with Oracle Application Server,
8-2

synchronization, 8-4

System Identifier, 5-1

system integration hub

Oracle Workflow role as, 1-23

T

tablespaces, 3-3

target process

Concurrent Processing server, 1-12

target system

cloning, 6-1

template

in XML Publisher, 1-25

third-party utilities

location, 2-4

three-tier architecture, 1-1

tier, 1-1

Tivoli

third-party single sign-on service

integration with Oracle Application Server,
8-2

Transaction Manager

Concurrent Processing, 1-13

transactional data

tablespace management and, 3-4

U

undo tablespaces, 3-6

Unicode, 13-1

unified driver

patch, 1-18

US7ASCII

character set, 13-1

user credentials

managed by Oracle Internet Directory, 8-6

UTF8, 13-1

character set, 13-1

utl_file_dir, 2-13

utl_file_dir parameter, 2-13

V

Virtual Private Database, 11-4

W

W3C, 1-24

Web Review Agent, 1-16
Web server
 description, 1-5
work shift
 Concurrent Processing, 1-12
worker
 Concurrent Processing server
 See target process

Workflow Engine, 1-24
worklist, 1-24
World Wide Web Consortium, 1-24

X

XSL-FO, 1-24

