Oracle® E-Business Suite

Concepts

Release 12.2

Part No. E22949-24

October 2025

ORACLE®

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

Send Us Your Comments

Preface

1   Architecture

2 File System Structure

3 Database Features

4 Patching and Management Tools

5 Technical Configuration Tools

6 Cloning

7 Security

8 Authentication and Integration

9 High Availability

10   Load Balancing

11   Globalization Support

12   Multiple Organization Architecture

Glossary

Index

Send Us Your Comments

Oracle E-Business Suite Concepts, Release 12.2

Part No. E22949-24

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 Oracle E-Business Suite Release Online Documentation CD available on My Oracle Support 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 inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 12.2 of the *Oracle E-Business Suite Concepts.*

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.

- Computer desktop application usage and terminology.

If you have never used Oracle E-Business Suite, we suggest you attend one or more of the Oracle E-Business Suite training classes available through Oracle University.

See Related Information Sources on page xii for more Oracle E-Business Suite product information.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup? ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Structure

1 Architecture
2 File System Structure
3 Database Features
4 Patching and Management Tools

Related Information Sources

This book is included in the Oracle E-Business Suite Documentation Library. If this guide refers you to other Oracle E-Business Suite documentation, use only the latest Release 12.2 versions of those guides.

Online Documentation

All Oracle E-Business Suite documentation is available online (HTML or PDF).

- **Online Help** - Online help patches (HTML) are available on My Oracle Support.

- **Oracle E-Business Suite Documentation Library** - This library, which is included in the Oracle E-Business Suite software distribution, provides PDF documentation as of the time of each release.

- **Oracle E-Business Suite Documentation Web Library** - This library, available on the Oracle Help Center (https://docs.oracle.com/cd/E26401_01/index.htm), provides the latest updates to Oracle E-Business Suite Release 12.2 documentation. Most documents are available in PDF and HTML formats.

- **Release Notes** - For information about changes in this release, including new features, known issues, and other details, see the release notes for the relevant product, available on My Oracle Support.

- **Oracle Electronic Technical Reference Manual -** The Oracle Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for each Oracle E-Business Suite product. This information helps you convert data from your existing applications and integrate Oracle E-Business Suite data with non-Oracle applications, and write custom reports for Oracle E-Business Suite products. The Oracle eTRM is available as an application in Oracle E-Business Suite.

Related Guides

You should have the following related books on hand. Depending on the requirements of your particular installation, you may also need additional manuals or guides.

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of

your Oracle E-Business Suite data.

Oracle Application Framework Component Reference Guide

This reference guide describes the components available for use in creating applications based on Oracle Application Framework. Designed to be used in conjunction with *Oracle Application Framework Developer's Guide*, it is available as a PDF book from My Oracle Support Knowledge Document 1315505.1, *OA Framework Component Reference Guide: Index*, and as HTML documentation shipped with Oracle Application Framework.

Oracle Application Framework Developer's Guide

This guide describes how to set up a suitable development environment and then build, test and deploy Oracle Application Framework applications. It includes the coding standards followed by Oracle E-Business Suite Development, instructions on how to comply with the latest Look and Feel (LAF) UI guidelines for developers, and information on extending Oracle E-Business Suite products. This guide is available as a PDF book from My Oracle Support Knowledge Document 1315485.1, *Oracle Application Framework Developer's Guide: Index*, and as HTML documentation shipped with Oracle Application Framework.

Oracle Application Framework Personalization Guide

This guide covers how to personalize Oracle Application Framework applications during development and implementation, and also while the applications are in use.

Oracle Application Management Pack for Oracle E-Business Suite Guide

This book is intended for database administrators and system administrators who are responsible for performing the tasks associated with maintaining an Oracle E-Business Suite system using the Oracle Application Management Pack for Oracle E-Business Suite.

Oracle Application Management Pack for Oracle E-Business Suite Metric Reference Manual

This book lists the target metrics for Oracle E-Business Suite that Oracle Enterprise Manager monitors.

Oracle Diagnostics Framework User's Guide

This manual contains information on implementing and administering diagnostics tests for Oracle E-Business Suite using the Oracle Diagnostics Framework.

Oracle E-Business Suite Cloud Manager Guide

This guide describes how to manage Oracle E-Business Suite environments on Oracle Cloud Infrastructure (OCI) using the Oracle E-Business Suite Cloud Manager tool.

Oracle E-Business Suite CRM System Administrator's Guide

This manual describes how to implement the CRM Technology Foundation (JTT) and use its System Administrator Console.

Oracle E-Business Suite Desktop Integration Framework Developer's Guide

Oracle E-Business Suite Desktop Integration Framework is a development tool that lets you define custom integrators for use with Oracle Web Applications Desktop Integrator. This guide describes how to define and manage integrators and all associated supporting objects, as well as how to download and upload integrator definitions.

Oracle E-Business Suite Developer's Guide

This guide contains the coding standards followed by Oracle E-Business Suite Development. It describes the Oracle Application Object Library components needed to implement the Oracle E-Business Suite user interface described in the *Oracle E-Business Suite User Interface Standards for Forms-Based Products*. It provides information to help you build your custom Oracle Forms Developer forms so that they integrate with Oracle E-Business Suite. In addition, this guide has information for customizations in features such as concurrent programs, flexfields, messages, and logging.

Oracle E-Business Suite Electronic Technical Reference Manual User's Guide

This guide describes how to set up and navigate Oracle E-Business Suite Electronic Technical Reference Manual (eTRM) user interface in Oracle E-Business Suite. It also explains how to browse and search the Oracle eTRM repository to locate desired FND and database metadata and objects, and how to view object details, reports, and diagrams.

Oracle E-Business Suite Flexfields Guide

This guide provides flexfields planning, setup, and reference information for the Oracle E-Business Suite implementation team, as well as for users responsible for the ongoing maintenance of Oracle E-Business Suite product data. This guide also provides information on creating custom reports on flexfields data.

Oracle E-Business Suite Installation Guide: Using Rapid Install

This book describes how to run Rapid Install to perform a fresh installation of Oracle E-Business Suite Release 12.2 or to replace selected technology stack executables in an existing instance.

Oracle E-Business Suite Maintenance Guide

This book explains how to patch an Oracle E-Business Suite system, describing the adop patching utility and providing guidelines and tips for performing typical patching operations. It also describes maintenance strategies and tools designed to help keep a system running smoothly.

Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2

This guide describes how to set up an Oracle E-Business Suite instance to support connections from Oracle E-Business Suite mobile apps. It also describes common administrative tasks for configuring Oracle E-Business Suite mobile apps and setup tasks for enabling push notifications for supported mobile apps. Logging and troubleshooting information is also included in this book.

Oracle E-Business Suite Mobile Apps Developer's Guide, Release 12.1 and 12.2

This guide describes how to develop enterprise-distributed mobile apps by using mobile application archive (MAA) files and how to implement corporate branding. It also explains required tasks on implementing push notifications for supported mobile apps. In addition, it includes how to implement Oracle E-Business Suite REST services to develop custom mobile apps by using the Login component from Oracle E-Business Suite Mobile Foundation or using any mobile app development framework if desired.

Oracle E-Business Suite Security Guide

This guide contains information on a comprehensive range of security-related topics, including access control, user management, function security, data security, secure configuration, and auditing. It also describes how Oracle E-Business Suite can be integrated into a single sign-on environment.

Oracle E-Business Suite Setup Guide

This guide contains information on system configuration tasks that are carried out either after installation or whenever there is a significant change to the system. The activities described include defining concurrent programs and managers, enabling Oracle Applications Manager features, and setting up printers and online help.

Oracle E-Business Suite User's Guide

This guide contains the user interface (UI) standards followed by Oracle E-Business Suite Development. It describes the UI for Oracle E-Business Suite products based on Oracle Forms, and how to apply this UI to the design of such applications.

Oracle E-Business Suite User Interface Standards for Forms-Based Products

This guide contains the user interface (UI) standards followed by Oracle E-Business Suite Development. It describes the UI for Oracle E-Business Suite products based on Oracle Forms, and how to apply this UI to the design of such applications.

Oracle E-Business Suite Integrated SOA Gateway User's Guide

This guide describes the high level service enablement process, explaining how users can browse and view the integration interface definitions and services residing in Oracle Integration Repository.

Oracle E-Business Suite Integrated SOA Gateway Implementation Guide

This guide explains how integration administrators can manage and administer the web service activities for integration interfaces including native packaged integration interfaces, composite services (BPEL type), and custom integration interfaces. It also describes how to set up and implement Service Invocation Framework to invoke SOAP and REST services from Oracle E-Business Suite, and how to manage web service security, configure logs, and monitor both inbound service invocations using Service Monitor and outbound service invocations through Service Invocation Framework using Service Invocation Monitor. In addition, this guide includes how to configure Large Language Model (LLM) service parameters and how to use LLM Service Monitor to track the LLM service invocation details.

Oracle E-Business Suite Integrated SOA Gateway Developer's Guide

This guide describes how integration developers can perform end-to-end service integration activities. These include orchestrating discrete web services into meaningful end-to-end business processes using business process execution language (BPEL), and deploying BPEL processes at runtime.

This guide also explains how to invoke web services using the Service Invocation Framework. This includes defining web service invocation metadata, invoking web services, and testing web service invocation.

Oracle e-Commerce Gateway User's Guide

This guide describes the functionality of Oracle e-Commerce Gateway and the necessary setup steps in order for Oracle E-Business Suite to conduct business with trading partners through Electronic Data Interchange (EDI). It also describes how to run extract programs for outbound transactions, import programs for inbound transactions, and the relevant reports.

Oracle e-Commerce Gateway Implementation Guide

This guide describes implementation details, highlighting additional setup steps needed for trading partners, code conversion, and Oracle E-Business Suite. It also provides architecture guidelines for transaction interface files, troubleshooting information, and a description of how to customize EDI transactions.

Oracle Cloud Using the Oracle E-Business Suite Adapter with Oracle Integration 3

This guide describes how to set up and use Oracle E-Business Suite Adapter connections in Oracle Integration to access supported Oracle E-Business Suite interfaces and REST services as inbound or outbound integrations from Oracle E-Business Suite.

Note that this book is the latest generation of Oracle Integration, Oracle Integration 3. Its prior generation, Oracle Integration Generation 2, is called Oracle Cloud Using the Oracle E-Business Suite Adapter with Oracle Integration. Both books are part of the integration documentation in Oracle Cloud Platform as a Service (PaaS) and are available in the Oracle Cloud Library on the Oracle Help Center.

Oracle Fusion Middleware Oracle E-Business Suite Adapter User's Guide

This book covers the use of Oracle E-Business Suite Adapter (formerly known as Adapter for Oracle Applications in Oracle Fusion Middleware 11g releases) in developing integrations between Oracle E-Business Suite and trading partners.

This book is available in the Oracle Fusion Middleware 14c Documentation Library, Oracle Fusion Middleware 12c Documentation Library, and Oracle Fusion Middleware 11g Documentation Library.

Oracle iSetup Developer's Guide

This manual describes how to build, test, and deploy Oracle iSetup Framework interfaces.

Oracle iSetup User's Guide

This guide describes how to use Oracle iSetup to migrate data between different instances of the Oracle E-Business Suite and generate reports. It also includes

information on configuration, instance mapping, and seeded templates used for data migration.

Oracle Report Manager User's Guide

Oracle Report Manager is an online report distribution system that provides a secure and centralized location to produce and manage point-in-time reports. Oracle Report Manager users can be either report producers or report consumers. Use this guide for information on setting up and using Oracle Report Manager.

Oracle Web Applications Desktop Integrator Implementation and Administration Guide

Oracle Web Applications Desktop Integrator brings Oracle E-Business Suite functionality to a spreadsheet, where familiar data entry and modeling techniques can be used to complete Oracle E-Business Suite tasks. You can create formatted spreadsheets on your desktop that allow you to download, view, edit, and create Oracle E-Business Suite data, which you can then upload. This guide describes how to implement Oracle Web Applications Desktop Integrator and how to define mappings, layouts, style sheets, and other setup options.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any product that includes workflow-enabled processes. It also describes how to manage workflow processes and business events using Oracle Applications Manager, how to monitor the progress of runtime workflow processes, and how to administer notifications sent to workflow users.

Oracle Workflow Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle E-Business Suite-embedded workflow processes. It also describes how to configure message metadata for Oracle Mobile Approvals for Oracle E-Business Suite and how to define and customize business events and event subscriptions.

Oracle Workflow User's Guide

This guide describes how users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle Workflow Client Installation Guide

This guide describes how to install the Oracle Workflow Builder and Oracle XML Gateway Message Designer client components for Oracle E-Business Suite.

Oracle XML Gateway User's Guide

This guide describes Oracle XML Gateway functionality and each component of the Oracle XML Gateway architecture, including Message Designer, Oracle XML Gateway

Setup, Execution Engine, Message Queues, and Oracle Transport Agent. It also explains how to use Collaboration History that records all business transactions and messages exchanged with trading partners.

The integrations with Oracle Workflow Business Event System, and the Business-to-Business transactions are also addressed in this guide.

Oracle XML Publisher Report Designer's Guide

Oracle XML Publisher is a template-based reporting solution that merges XML data with templates in RTF or PDF format to produce a variety of outputs to meet a variety of business needs. Using Microsoft Word or Adobe Acrobat as the design tool, you can create pixel-perfect reports from the Oracle E-Business Suite. Use this guide to design your report layouts.

This guide is available through the Oracle E-Business Suite online help. For more information, see: *Notes for Using Oracle Business Intelligence Publisher 10g in Oracle E-Business Suite Release 12.2*, My Oracle Support Knowledge Document 1640073.1.

Oracle XML Publisher Administration and Developer's Guide

Oracle XML Publisher is a template-based reporting solution that merges XML data with templates in RTF or PDF format to produce a variety of outputs to meet a variety of business needs. Outputs include: PDF, HTML, Excel, RTF, and eText (for EDI and EFT transactions). Oracle XML Publisher can be used to generate reports based on existing Oracle E-Business Suite report data, or you can use Oracle XML Publisher's data extraction engine to build your own queries. Oracle XML Publisher also provides a robust set of APIs to manage delivery of your reports via e-mail, fax, secure FTP, printer, WebDav, and more. This guide describes how to set up and administer Oracle XML Publisher as well as how to use the Application Programming Interface to build custom solutions.

This guide is available through the Oracle E-Business Suite online help. For more information, see: *Notes for Using Oracle Business Intelligence Publisher 10g in Oracle E-Business Suite Release 12.2*, My Oracle Support Knowledge Document 1640073.1.

Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the Oracle E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

Do Not Use Database Tools to Modify Oracle E-Business Suite Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle E-Business Suite 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 E-Business Suite data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle E-Business Suite tables are interrelated, any change you make using an Oracle E-Business Suite form can update many tables at once. But when you modify Oracle E-Business Suite data using anything other than Oracle E-Business Suite, 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 E-Business Suite.

When you use Oracle E-Business Suite to modify your data, Oracle E-Business Suite automatically checks that your changes are valid. Oracle E-Business Suite 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.

1

Architecture

Introduction

This chapter describes the Oracle E-Business Suite architecture and the key features the architecture supports, plus some related points. Topics include:

- Overview

- The Client (Desktop) Tier

- The Application Tier

- The Database Tier

- The Oracle E-Business Suite Technology Layer

- Oracle Configuration Manager

- Oracle E-Business Suite Patch Nomenclature

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

A server (or services) is a process or group of processes that runs on a single machine and provides a particular functionality. For example, *Web services* process HTTP requests, and *Forms services* process requests for activities related to Oracle Forms. The *Concurrent Processing server* supports data-intensive programs that run in the background.

> **Important:** The term *server*, in the sense of a single process, is less appropriate in the Release 12 architecture. Where applicable, replacement terms such as *services* are used.

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 Oracle E-Business Suite components, and is sometimes known as the middle tier; and the *client (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. Note, however, that a node is also a software concept, referring to a logical grouping of servers.

Centralizing the Oracle E-Business Suite software on the application tier eliminates the need to install and maintain application software on each client PC, and also enables Oracle E-Business Suite to scale well with an increasing load. Extending this concept further, one of the key benefits of using the *Shared Application Tier File System* model (originally *Shared APPL_TOP*) is the need to maintain only a single copy of the relevant Oracle E-Business Suite code, instead of a copy for every application tier machine.

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

Oracle E-Business Suite Three-Tier Architecture
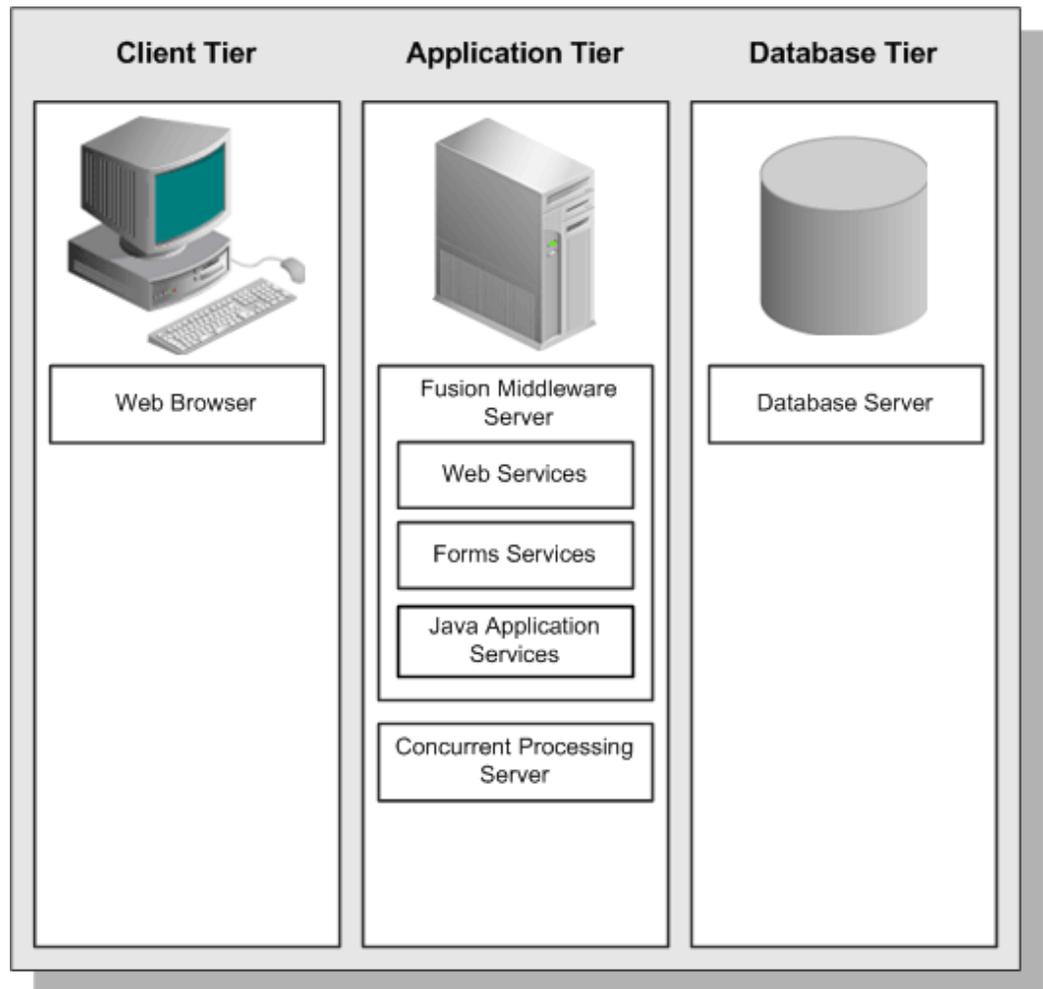
The Client Tier

The client interface is provided via either HTML for the large and growing number of HTML-based OA Framework applications, and via Java applets for the smaller number of older Oracle Forms-based applications and a few product-specific features.

> **Note:** The client tier is sometimes referred to as the *desktop tier*.

Client Tier and Application Tier Components
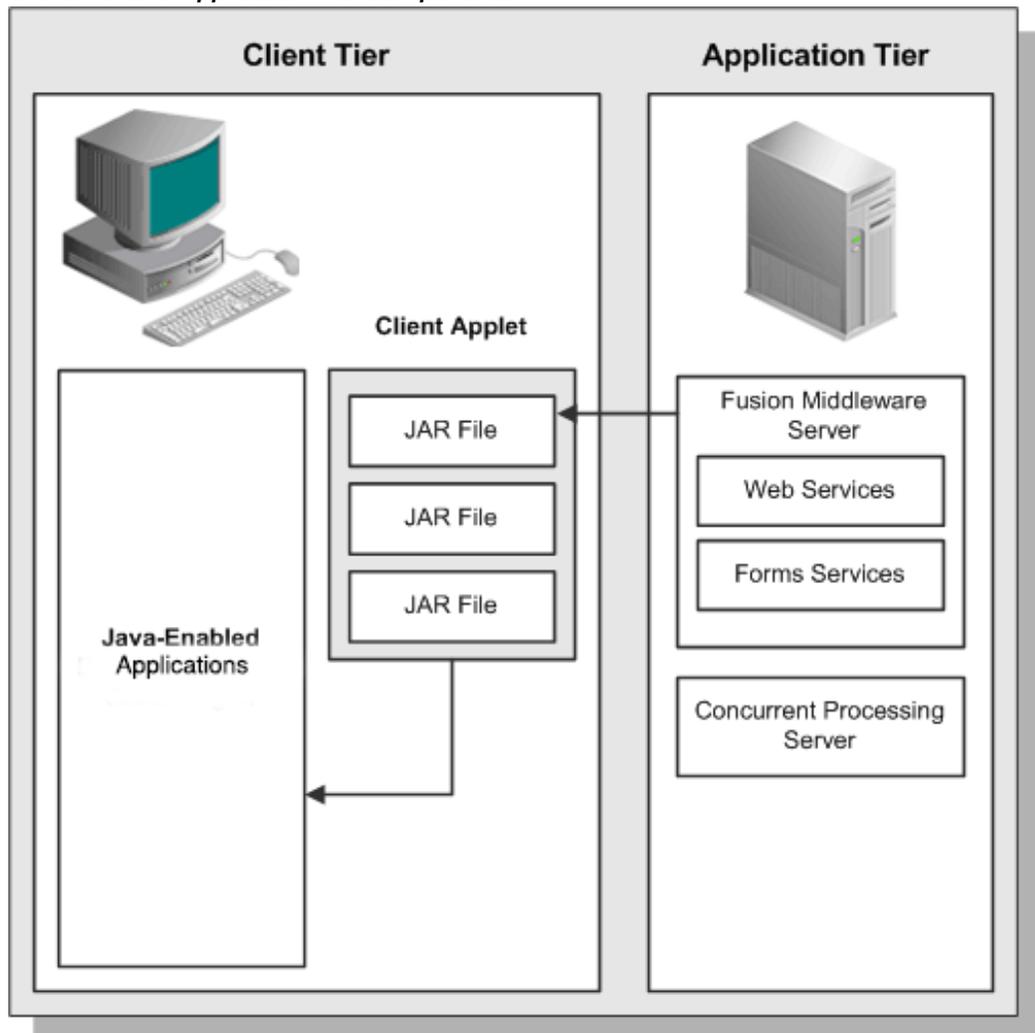
You log in via the Oracle E-Business Suite Home Page on a desktop client web browser. The Home Page provides a single point of access to HTML-based applications, Forms-based applications, and Business Intelligence applications.

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

Oracle E-Business Suite Home Page
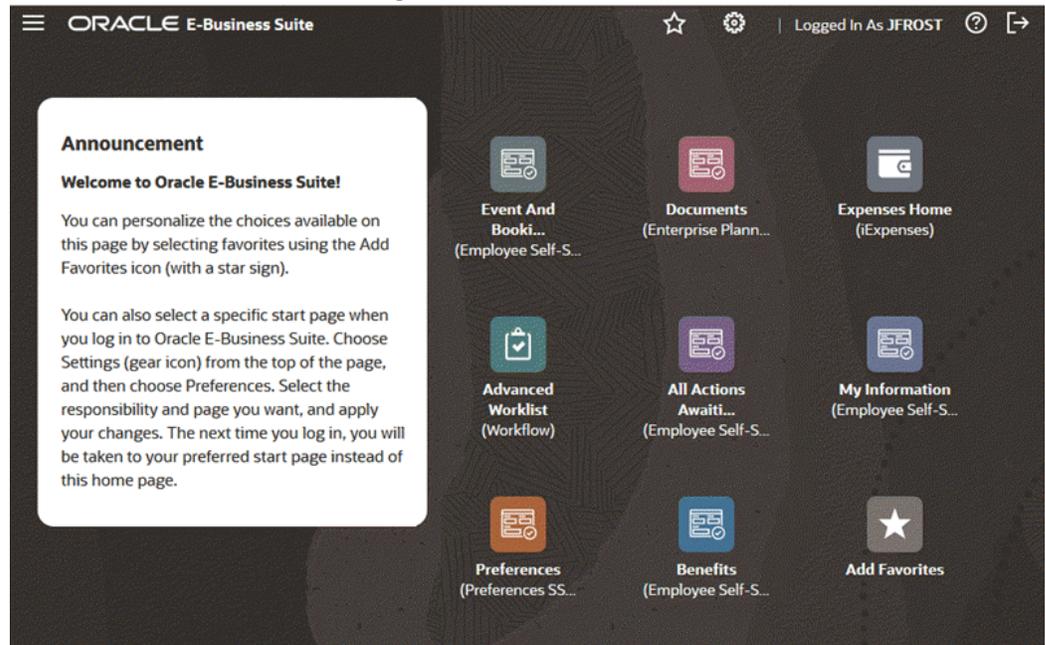
Forms Client Applet

The *Forms client applet* is a general-purpose presentation applet that supports all Oracle E-Business Suite 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 E-Business Suite forms.

Desktop Java Runtime Environment

The Oracle Forms client applet must run within a Java Virtual Machine (JVM) on the desktop client. It is launched on the user's desktop using a Java Deployment technology, either Java Web Start or Java Plug-in. These are included as part of the Java Runtime Environment (JRE) software. Java Deployment technologies require the JRE software to be installed on the client, so that Java-based applications can run on it. After you download and install the JRE software, you will be able to run Oracle Forms-based applications as shown in the example on the following screenshot.

Forms-Based Oracle E-Business Suite Interface
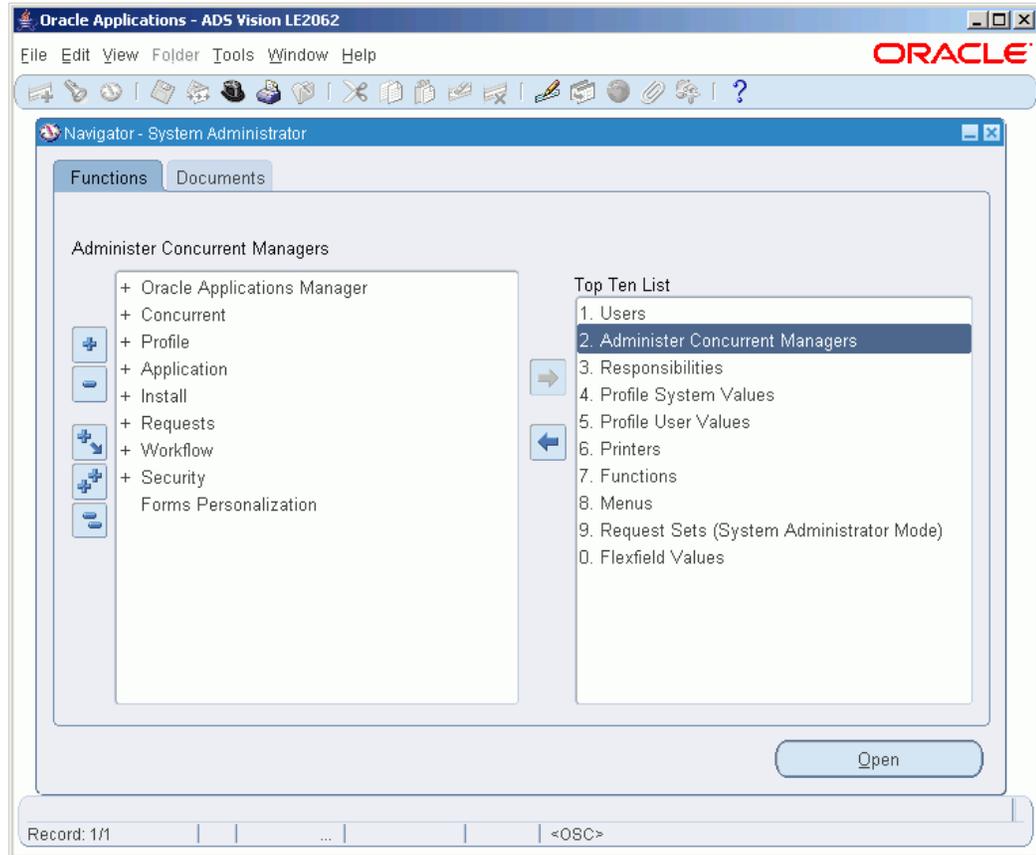
The Forms client applet and commonly used JAR files are downloaded from the Web server at the beginning of the client's first session. Less commonly used JAR files are downloaded as needed. All downloaded JAR files are cached locally on the client, ready for future sessions. This eliminates the network traffic that would be involved in downloading them whenever they were required.

In Release 12, the cache directory path is of the form:

```
<HOMEDRIVE>\Documents and Settings\<Windows User
Name>\Application Data\Sun\Java\Deployment\cache
```

For example:

```
C:\Documents and Settings\jsmith\Application
Data\Sun\Java\Deployment\cache
```

Selecting "Show console" on the "Advanced" tab of the JRE Plug-in control panel will allow you to observe downloading of JAR files, to confirm they are being downloaded when they should be. The Java console is shown in the following screenshot.

Java Console

```
Java Console                                    _  □  ✕

Java Plug-in 1.6.0_26
Using JRE version 1.6.0_26-b03 Java HotSpot(TM) Client VM
User home directory = C:\Documents and Settings\jfarring

---------------------------------------------------
c:   clear console window
f:   finalize objects on finalization queue
g:   garbage collect
h:   display this help message
l:   dump classloader list
m:   print memory usage
o:   trigger logging
q:   hide console
r:   reload policy configuration
s:   dump system and deployment properties
t:   dump thread list
v:   dump thread stack
x:   clear classloader cache
0-5: set trace level to <n>
---------------------------------------------------

proxyHost=null
proxyPort=0
connectMode=HTTP, native.
Forms Applet version is : 10.1.2.3

              Clear        Copy        Close
```

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

Java Web Start

Java Web Start launches Oracle E-Business Suite Java-based functionality as Java Web Start applications instead of as Java applets. Java Web Start is a Java Deployment technology that is installed as part of the Java Runtime Environment (JRE) software. It provides a browser-independent architecture for deploying Java technology-based applications to the client desktop, and is the recommended configuration for Oracle E-Business Suite.

> **Additional Information:** For details about how you can migrate an existing Oracle E-Business Suite Release 12.2 or 12.1 environment to use Java Web Start, refer to My Oracle Support Knowledge Document 2188898.1, *Using Java Web Start with Oracle E-Business Suite*.

Java Plug-in

Running Oracle E-Business Suite's Java-based content using the Java Plug-in requires a browser that supports Netscape Plug-in Application Programming Interface (NPAPI) plug-ins. Most browsers have now phased out NPAPI plug-in support, preventing he Java Plug-in from working. Oracle E-Business Suite supports launching Java applets (such as the Forms applet) using the Java Plug-in for browsers that support it.

> **Additional Information:** For further details, refer to My Oracle Support knowledge documents 393931.1, *Deploying JRE (Native Plug-in) for Windows Clients in Oracle E-Business Suite Release 12*, and 2510500.1, *FAQ: Essentials of Java Usage in Oracle E-Business Suite*.

The Application Tier

The *application tier* has a dual role: hosting the various servers and service groups that process the business logic, and managing communication between the desktop tier and the database tier. This tier is sometimes still referred to as the *middle tier*.

Several service groups or servers comprise the basic Oracle E-Business Suite application tier:

*   HTTP services

*   Java services

*   Forms services

*   Concurrent Processing server

As well as these, some less visible application tier services provide further infrastructure support.

In Release 12.2, Web and Forms services are provided by *Oracle Application Server* and Oracle Fusion Middleware. They are no longer servers in the sense of being a single process, as was the case in previous releases.

> **Tip:** It is advisable 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 and services to help 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 services

- Forms services

- Concurrent Processing server

Load balancing is discussed in more detail in a later chapter.

Application Tier ORACLE_HOMEs in Release 12.2

Oracle E-Business Suite Release 12.2 uses two application tier ORACLE_HOMEs.

- An OracleAS 10.1.2 ORACLE_HOME that was used in previous 12.x releases.

- An Oracle Fusion Middleware (FMW) ORACLE_HOME that supports Oracle WebLogic Server (WLS) and supersedes the Java (OracleAS 10.1.3) ORACLE_HOME that was used in previous releases.

The use of these two ORACLE_HOMEs enables Oracle E-Business Suite to take advantage of the latest Oracle technologies.

Application Tier Structure
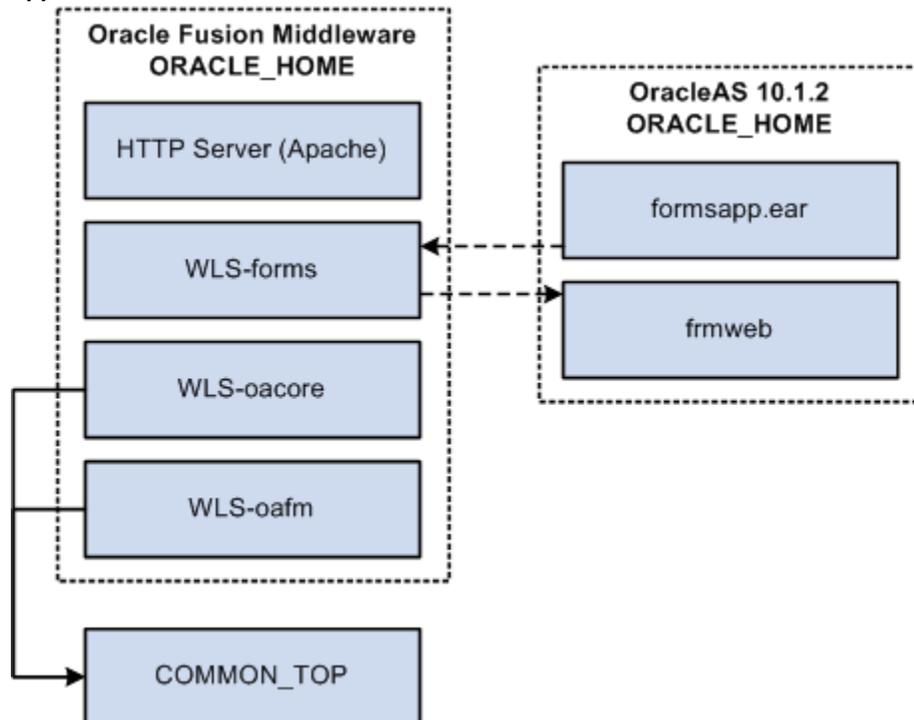
Notable features of this architecture include:

- The Oracle E-Business Suite modules (packaged in the file formsapp.ear) are deployed out of the OracleAS 10.1.2 ORACLE_HOME, and the `frmweb` executable is also invoked out of this ORACLE_HOME.

- All major services are started out of the Fusion Middleware ORACLE_HOME.

Key changes from earlier releases include:

- The Oracle Application Server 10.1.2 ORACLE_HOME (sometimes referred to as the Tools, C, or Developer ORACLE_HOME) replaces the 8.0.6 ORACLE_HOME provided by Oracle9i Application Server 1.0.2.2.2 in Release 11i.

- The FMW ORACLE_HOME (sometimes referred to as the Web or Java ORACLE_HOME) replaces the OracleAS 10.1.3.-based ORACLE_HOME used in Oracle E-Business Suite 12.x releases prior to 12.2.

Web Services

The Web services component of Oracle Application Server processes requests received over the network from the desktop clients, and includes the following major components:

- Web Listener (Oracle HTTP Server powered by Apache)

- Java Servlet Engine (Oracle WebLogic Server, WLS)

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

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.

> **Note:** Oracle WebLogic Server integration is new in Release 12.2.

HTML-Based Applications and the Oracle Application Framework

The Oracle HTML-based applications (originally known as Self-Service applications) add a browser-based, walk-up-and-use functionality to Oracle E-Business Suite. They include numerous products such as Self-Service Expenses, Self-Service Human Resources, Internet Procurement, Internet Receivables, Self-Service Time, Web Suppliers, iStore, iPayment, iSupport, and iMarketing, and have the following characteristics:

- Do not use Oracle Forms for the interface

- Use HTML documents, JavaScript, Java Server Pages, JavaBeans, and Servlets

- Dynamically generate HTML pages by executing Java code

- Use a metadata dictionary for flexible layout

- Operate via a direct connection to the Web server

The Oracle HTML-based applications can be either inquiry or transactional. Inquiry modules only read the Oracle E-Business Suite database. In contrast, transactional modules both read and update the database.

The *Oracle Application 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 Application 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.

- *Oracle WebLogic Server* supplies the Oracle Application Framework with underlying security and applications Java services. It provides the Oracle Application 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.

HTML-Based Applications Architecture
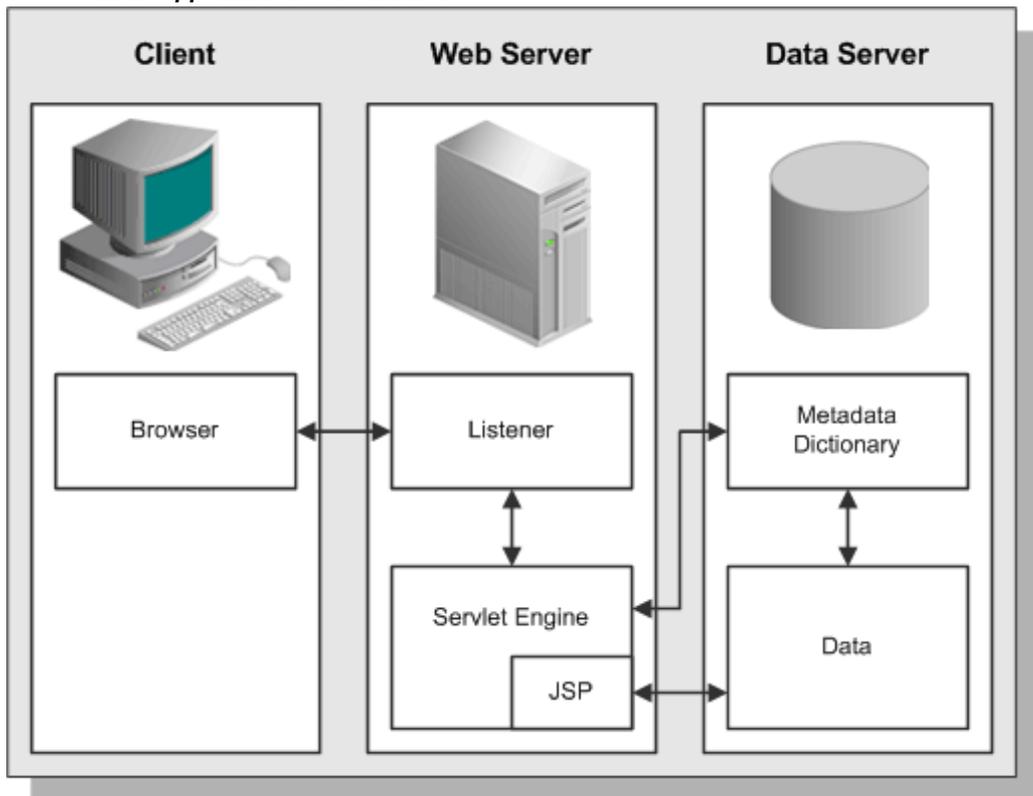
Java Servlet Access with HTML-Based Applications

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 (Oracle WebLogic Server), where it runs a JSP.

4. The JSP obtains the content from the Oracle E-Business Suite 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.

Oracle Application Framework Architecture
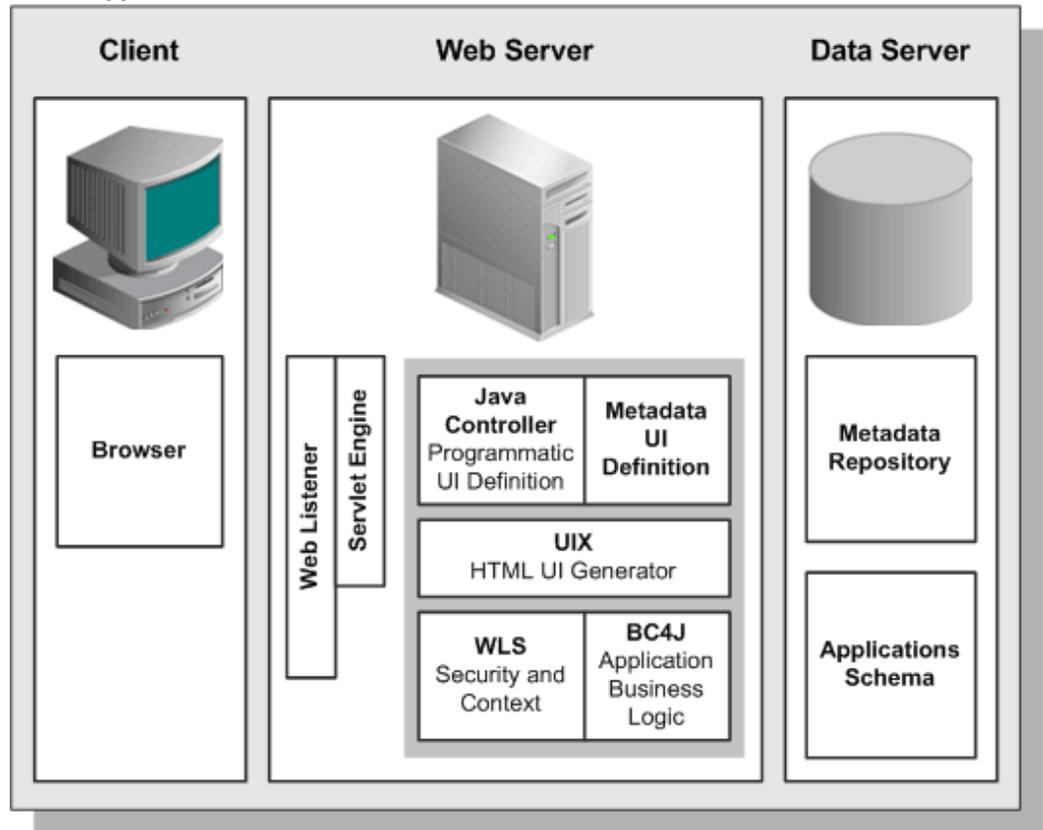
Oracle Application Framework Processing Details

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

1. Oracle WebLogic Server 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 Services

By default, Forms services in Oracle E-Business Suite Release 12.2 are provided by the *Forms listener servlet*, which, as described further below, facilitates the use of firewalls, load balancing, proxies, and other networking options.

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

Forms Listener Servlet Architecture

The Forms listener servlet is a Java servlet that delivers the ability to run Oracle Forms applications over HTTP or HTTPS connections. It hosts the Oracle E-Business Suite forms and associated runtime engine, mediating 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 listener servlet 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 listener servlet can communicate with the desktop client using either a standard HTTP network connection or secure HTTPS network connection. In contrast, Forms services (formerly known as Forms server) communicates with the desktop client using the TCP/IP network protocol, on top of which it layers its own protocol.

The Forms listener servlet communicates with the Oracle database server using the *Oracle Net* networking infrastructure.

The Forms listener servlet manages the creation of a Forms runtime process for each client, as well as network communications between the client and its associated Forms runtime process. The client sends HTTP requests and receives HTTP responses from the Web services, which acts as the network endpoint for the client.

Forms Socket Mode Architecture

In the traditional Forms server *socket mode* 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 a 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.

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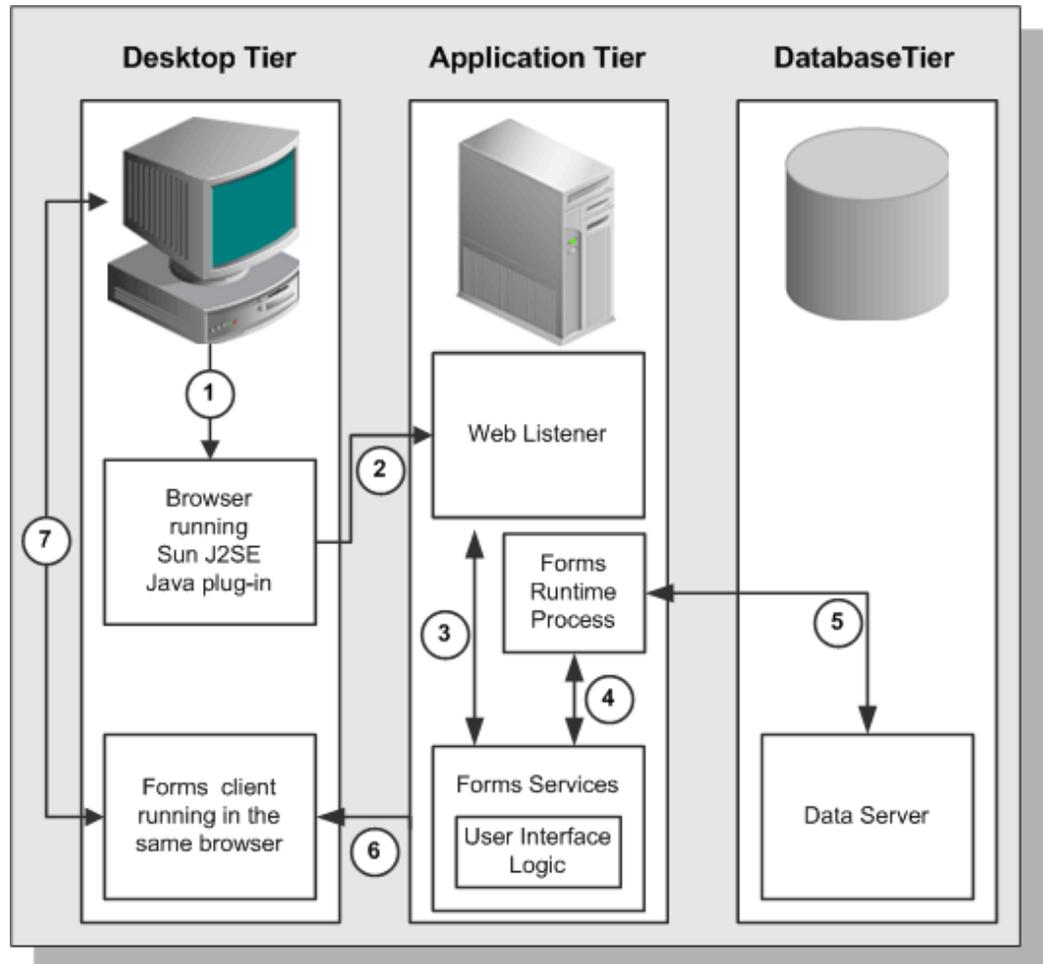
Once a connection has been made, many operations can be performed with little or no
further interaction with the Forms server. For example, when a few field values change
in response to a user action, there is no need to update the entire screen. In this scenario,
only the changed fields are updated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

Choice of Mode

As stated, by default Oracle E-Business Suite Release 12.2 utilizes Forms listener servlet
mode. However, socket mode is still supported, and may be useful in high-latency,
bandwidth-constrained WAN environments.

> **Additional Information:** For more details of utilizing Forms Socket
> Mode, see My Oracle Support Knowledge Document 384241.1, *Using
> Forms Socket Mode with Oracle E-Business Suite Release 12*.

Java APIs for Forms

Building on top of Oracle Fusion Middleware and service-oriented architecture (SOA) technology, *Oracle E-Business Suite Integrated SOA Gateway* (ISG) provides a robust communication and integration infrastructure between independently managed components and loosely coupled applications. This promotes more effective integration between heterogeneous applications, and facilitates the development and execution of complex business processes into flexible and reusable Web services. With this standardized Web service platform, Oracle E-Business Suite Integrated SOA Gateway provides a framework that efficiently supports service integration between Web applications.

> **Note:** For further details, see My Oracle Support Knowledge Document 556540.1, *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12*.

Within this integration environment, Java APIs for Forms are XML document-based integration points wrapped in Java classes, and used to execute business logic in Oracle Forms. These specialized Java classes are categorized as a subtype of Java interface, and can be service-enabled through SOA Provider.

> **Note:** For further details of Java APIs and Web services, see Oracle E-Business Suite Integrated SOA Gateway User's Guide, Development Guide, and Implementation Guide.

Concurrent Processing Server

As described previously, user interactions with Oracle E-Business Suite 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 E-Business Suite feature that allows these non–interactive and potentially long-running functions to be executed efficiently alongside interactive operations. It uses operating system facilities to enable 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 by a specialized component , the *Concurrent Processing server*.

Concurrent Processing Server Implementation

The server is located on the Oracle E-Business Suite application tier, and implemented

via the *Batch Processing Services* service group.

> **Note:** For further information about application tier services and service groups, see *Oracle E-Business Suite Installation Guide: Using Rapid Install*, Chapter 2.

Concurrent Requests and Managers

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 Location

Traditionally, the concurrent managers were installed on the same machine as the database. However, with the much faster local networks that are now common, and the typical co-location of database and application tier machines, it is generally preferable to run them on a separate machine. This greatly faciliates ease of management and maintenance, as well as providing deployment flexibility.

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 E-Business Suite, 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*

Specialist Concurrent Managers

As well as the general-purpose concurrent managers described in the previous section, there are a number of specialized concurrent managers.

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 E-Business Suite 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 without careful planning, otherwise some programs might not be able to 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 Advanced Queueing. 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

Oracle E-Business Suite Setup 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.

All reports are run through the Concurrent Processing server manager via the `rwrun` executable, which spawns an in-process server. This is in contrast to earlier releases of Oracle E-Business Suite, which used the now-obsolete Reports server.

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.

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 E-Business Suite 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:

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 *Oracle E-Business Suite Setup 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 node's Applications Listener (which is dedicated to Concurrent Processing) to spawn a process running the Service Manager executable (FNDSM). The Applications Listener must be configured to source the Oracle E-Business Suite 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.

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 services.

4. The Forms services pass the report back to the user's browser one page at a time.

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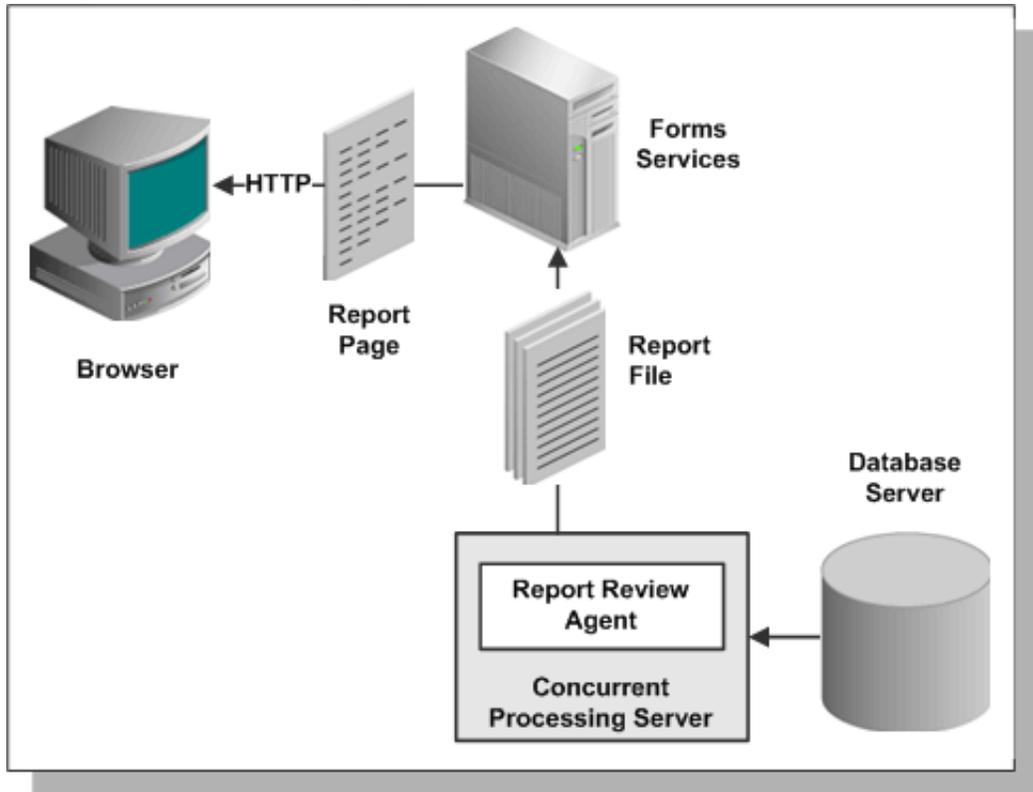
You can cater for your network capacity and report volume by using profile options to specify the maximum size of the files and pages that can be passed through the system.

Parallel Concurrent Processing

Parallel Concurrent Processing (PCP) allows concurrent processing activities to be distributed across multiple nodes in an Oracle Real Application Clusters (Oracle 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 enabled by default, so PCP is always available for use in environments where one or more concurrent processing nodes exist.

> **Note:** PCP does not require an Oracle RAC environment. Conversely, you do not have to use PCP in an Oracle RAC environment, although it typically makes sense to do so.

Managing Concurrent Processing

From the command line, two commands can be entered to control the Internal Concurrent Manager: `startmgr`, which starts the ICM; and `concsub`, 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 INST_TOP/admin/scripts/adcmctl.sh.

The various components of the concurrent processing system can be managed from various forms, such as *Concurrent Manager: Administer*, or from *Concurrent Managers* or *Concurrent Requests* under Oracle Applications Manager (OAM).

> **Additional Information:** For details of setting up and managing concurrent processing, see *Oracle E-Business Suite Setup Guide*.

Concurrent Processing and Online Patching

The introduction of online patching with Release 12.2 has involved the introduction of a new concurrent program, ADZDPATCH, to ensure that concurrent programs cannot run during an online patching cycle if they are incompatible with this new patching model.

Full, practical details of the interaction between concurrent processing and online patching are provided in the Patching Procedures chapter of *Oracle E-Business Suite Maintenance Guide*

Application Tier Administration

Any application tier node can be used to carry out the following operations:

- **Applying Oracle E-Business Suite application patches**

  In general, *Oracle E-Business Suite patches* consist of files and scripts that update the file system and database objects. In Release 12.2, a single *unified (u) driver* file combines the features of the older *copy* (c), *database* (d), and *generate* (g) driver files used in earlier releases.

  You use the *adop* utility to perform these updates. This utility may also be used to apply cumulative patches such as *mini-packs* and *maintenance packs*.

- **Maintaining Oracle E-Business Suite data**

  Some features require updates to the 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.

The Database Tier

The database tier contains the Oracle database server that stores and manages all the data maintained by Oracle E-Business Suite. This includes the various types of file in which the tables, indexes, and other database objects for your system physically reside, as well as the database executables. The database also stores the Oracle E-Business Suite online help information.

The database server communicates with the services and servers on the application tier, which mediate the communications between the database and the clients: there is no direct communication between the database and clients.

Using a Mixed Platform Architecture

The Oracle database server is sometimes available on platforms where Oracle E-Business Suite is not currently certified. In such a case, it may be possible to utilize a *mixed platform architecture*, where the database is installed on one platform and the application tier on another.

This type of deployment can enable the database to utilize the specific features offered by a particular platform. It can also allow the application tier to be managed in a more cost-effective way.

> **Additional Information:** For up-to-date details of Release 12.2 support with mixed platform architectures, see Certify on My Oracle Support.

The Oracle E-Business Suite Technology Layer

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

Products in the Oracle E-Business Suite technology layer include:

- Oracle Applications DBA (AD)

- Oracle Application Object Library (FND)

- Oracle Applications Utilities (AU)

- Oracle Common Modules (AK)

- Oracle Workflow (WF)

- Oracle Web Applications Desktop Integrator (BNE)

- Oracle Alert (ALR)

- Oracle Application Framework (FWK)

- Oracle BI Publisher (XDO)

Oracle Applications DBA (AD)

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

The AD utilities include:

- **AD Administration** - Performs general maintenance tasks for Oracle E-Business Suite.

- **AD Merge Patch** - Merges multiple patches into a single, integrated patch.

- **AutoConfig** - Manages configuration changes to components that are specific to an Oracle E-Business Suite system. Configuration changes to technology components that are not specific to Oracle E-Business Suite are managed via the native tools for those components.

- **AutoPatch** - Applies patches and adds new languages and products to an Oracle E-Business Suite system.

    > **Important:** In Oracle E-Business Suite Release 12.2, the actual patching utility you run is `adop` (AD online patching), rather than `adpatch` as in previous releases.

- **Rapid Clone** - Used to copy (clone) an Oracle E-Business Suite system.

- **Rapid Install** - Sets up a fully configured Oracle E-Business Suite system, including

the latest certified technology stack and all patches, mini-packs, and other updates.

MetaData Services (MDS)

MetaData Services is an active data dictionary that enables you to define Oracle E-Business Suite components for HTML-based applications, and generate many of the Oracle E-Business Suite runtime characteristics.

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.

Oracle Applications Utilities (AU)

The Applications Utilities (AU) component is used to maintain the Oracle E-Business Suite system.

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

Oracle Application Object Library (FND)

The Oracle Application Object Library is a key component of the Oracle E-Business Suite 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 E-Business Suite 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 Oracle E-Business Suite products.

Standard User Interface

Oracle Application Object Library supports the integration of Oracle E-Business Suite 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.

Online Help

The presentation of Online Help is also standardized across all Oracle E-Business Suite products.

Developer Features

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

- The same coding and Graphical User Interface (GUI) standards used by Oracle E-Business Suite 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 E-Business Suite 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 E-Business Suite.

Features for System Administrators

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

- Registering new Oracle E-Business Suite 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 E-Business Suite 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 E-Business Suite via user sign-ons and responsibilities. Each user must have a valid user name and password to gain access to Oracle E-Business Suite.

A *responsibility* is a level of authority in Oracle E-Business Suite that lets users access only those functions and data appropriate to their roles in the organization. For example, responsibilities may be used to allow access to a specific product, ledger, operating unit, or to a restricted list of windows, functions, reports, or groups of products.

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 E-Business Suite, a standard 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 (WF)

Oracle Workflow delivers a complete workflow management system that supports business process based integration. Its technology enables modeling, automation, and continuous improvement of business processes, routing information of any type according to user-defined business rules. Oracle Workflow also 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.

Workflow 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 *Workflow Engine*, embedded in the Oracle database, implements process definitions at runtime. The Workflow Engine monitors workflow states and coordinates the routing of activities for a process. Changes in workflow state, such as the completion of workflow activities, are signaled to the engine via a PL/SQL or Java API.

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.

Features and Usage

A completed application transaction or event can initiate a workflow process by raising a business event or by calling a series of Workflow Engine APIs. The Workflow Engine drives through the process, performing all automated steps and calling the Notification System to deliver notifications for steps that involve human intervention.

You can review and respond to your business process notifications from one central page, 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.

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.

Additional Capabilities

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 data content 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 Web Applications Desktop Integrator (BNE)

Oracle Web Applications Desktop Integrator brings Oracle E-Business Suite data to the desktop, where you can use familiar spreadsheet data entry and modeling techniques in Microsoft Excel to complete your Oracle E-Business Suite tasks. You can create formatted spreadsheets on your desktop that allow you to download, view, edit, and create Oracle E-Business Suite data. To save time, you can use data entry shortcuts such as copying and pasting (or dragging and dropping) ranges of cells, or use Excel formulas to calculate amounts. You can combine speed and accuracy by invoking lists of values for fields within the spreadsheet.

After editing the spreadsheet, you can validate the data before uploading it to the Oracle E-Business Suite. Validation messages are returned to the spreadsheet, allowing you to identify and correct invalid data.

The fields that appear in the spreadsheet, their positions, and their default values can all be customized through the Oracle Web Applications Desktop Integrator layout functionality. A layout lets you create a more productive work environment by removing unnecessary fields from the spreadsheet, and by organizing the spreadsheet in a way that conforms to your practices.

Oracle E-Business Suite Desktop Integration Framework is a development tool that lets you define custom integrators for use with Oracle Web Applications Desktop Integrator. An integrator is a set of metadata that encapsulates all the information needed to integrate a particular Oracle E-Business Suite task with a desktop application. With Oracle E-Business Suite Desktop Integration Framework, you can define custom integrators for tasks in a standard Oracle E-Business Suite application that are not covered by seeded integrators, or for tasks in custom applications developed for your instance.

Oracle E-Business Suite Desktop Integration Framework provides a graphical user interface in which developers can define integrators and all associated supporting objects, without needing to work directly with the underlying Oracle Web Applications Desktop Integrator tables and APIs. Through this user interface, Oracle E-Business Suite Desktop Integration Framework reduces development time, increases developer productivity, and enhances ease of maintenance for the integrators you define.

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 E-Business Suite database does not have adequate free space.

Oracle BI Publisher (XDO)

Oracle BI Publisher (formerly known as Oracle XML Publisher) is a Java-based product based on the World Wide Web Consortium (W3C) *Extensible Stylesheet Language* (XSL). Specifically, Oracle BI 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).

Oracle BI 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 Oracle BI 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 Oracle BI 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 Oracle BI Publisher are a Java-based set of publishing tools accessible via Java APIs from Oracle E-Business Suite or 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, for example as PDF.

- **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 Oracle BI 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 Oracle BI 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.

- **Oracle BI Publisher APIs** - Oracle BI Publisher provides APIs to allow other Oracle E-Business Suite products to communicate directly with the underlying processes.

Network Topologies

Oracle E-Business Suite is designed to run on a local area network (LAN), with the database and application tiers co-located. This is because Oracle E-Business Suite processes generate significant network traffic between the database tier and the application tier, making performance very sensitive to the network characteristics of the

connection, especially the latency. The best performance is therefore expected when the database and application tiers are located in the same data center.

However, you may be able to achieve adequate performance with the database tier and the application tier situated in separate data centers within the same geographical region if you are able to deploy a specialized network solution that guarantees high bandwidth and low latency between the data centers.

A deployment that separates the database tier and application tier over a wide area network (WAN) will not provide performance suitable for running EBS, and is not supported.

> **Additional Information:** For a discussion of the effects of different network layers on load balancing, see My Oracle Support Knowledge Document 380489.1, *Using Load-Balancers with Oracle E-Business Suite Release 12*.

2

File System Structure

Introduction

An Oracle E-Business Suite Release 12.2 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 Oracle E-Business Suite components are installed on desktop client machines, although JAR files and their supporting utilities are downloaded as required.

Depending on how you chose to install Oracle E-Business Suite, 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 tier machines. This chapter discusses the association between these environment settings and the corresponding files and directories.

Oracle E-Business Suite Environment

Oracle E-Business Suite makes extensive use of environment settings to locate executable programs and other files essential to Oracle E-Business Suite operation. These environment settings are defined when you install Oracle E-Business Suite. 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 on UNIX (`.cmd` on Windows). Environment files and settings are discussed in more detail later in this chapter.

Instance Home

Like other 12.x releases, Oracle E-Business Suite Release 12.2 uses the concept of a top-level directory for an Oracle E-Business Suite instance. This directory is referred to as the *Instance Home*, and denoted by the environment variable $INST_TOP.

Using an Instance Home provides the ability to share application and technology stack code among multiple instances, for example a development instance and a test instance. Most configuration files created by AutoConfig are stored under the Instance Home.

Other benefits of the Instance Home include support for *read-only file systems*. Prior to Release 12.2, all the configuration files modified by AutoConfig were located under $INST_TOP. The following parts of the file system could be made read-only: APPL_TOP, OracleAS 10.1.2 ORACLE_HOME, and OracleAS 10.1.3 ORACLE_HOME. In Release 12.2, the APPL_TOP and FMW_Home contain some configuration files modified by AutoConfig. Only the OracleAS 10.1.2 ORACLE_HOME configuration files are located under $INST_TOP, and so can be made read-only.

In a single-node application tier environment, the basic structure of the Instance Home is: `<s_base>/inst/apps/<context_name>`, where `s_base` (which does not have a corresponding environment variable) is the top level of the Oracle E-Business Suite installation, and `<context_name>` is the highest level at which the applications context exists. For example, $INST_TOP might be `/u01/R122_EBS/fs1/inst/apps/vision_testsys`, where `vision_testsys` is the context name.

In a multi-node application tier environment, the Instance Home is typically located on a local filesystem. For example, $INST_TOP might be `/inst/R122_EBS/fs1/inst/apps/vision_testsys`, where `vision_testsys` is the context name.

Instance Home Directory Structure
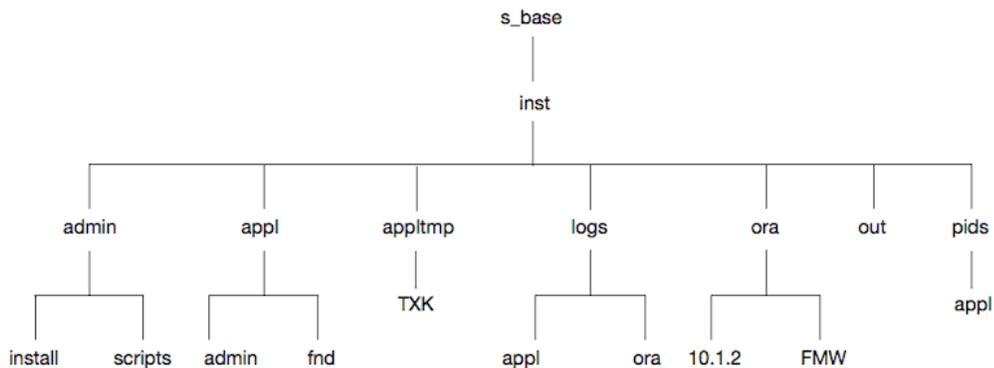
The HTTP and Oracle WebLogic Server configuration files are not stored under the $INST_TOP. For further details, refer to the Techical Configuration chapter of *Oracle E-Business Suite Setup Guide*.

> **Note:** The Oracle Fusion Middleware top-level directory and its subdirectories (described later) are completely new in Oracle E-Business Suite Release 12.2, as this technology was not used in previous releases.

The EBSapps Directory

The EBSapps directory is another key high-level directory that was introduced in Oracle E-Business Suite Release 12.2. It will be located under a path with a name such as `/u01/R122_EBS/fs1`, at the same level as the `inst` directory.

EBSapps Directory Structure

```
                              s_base
                                |
                               fs1
                                |
                             EBSapps
         _____|_____
        |                           |                           |
      10.1.2                       apps                        comn
        |              _____|_____      _____|_____
     Apache          |     |      |     |       |        |      |      |     |      |
                     ad  admin   ap   zsa      zx      admin  java  _pages util webapps
```
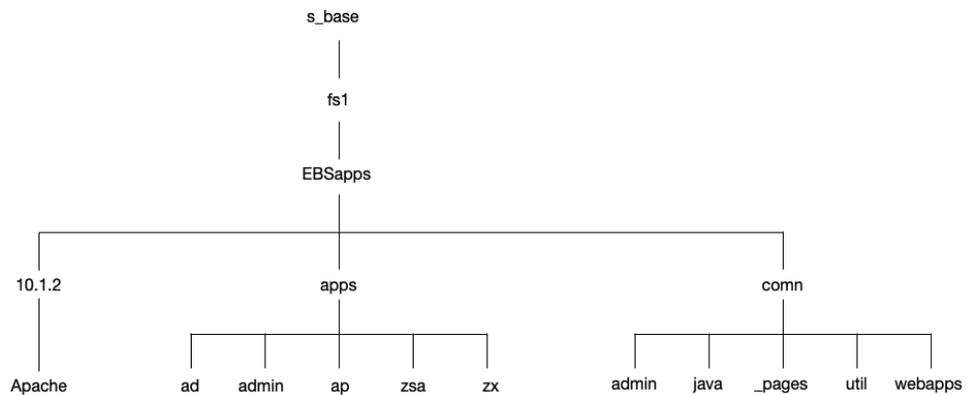
Fusion Middleware Home

The Oracle Fusion Middleware (FMW) directory is another key high-level directory. It is completely new in Oracle E-Business Suite Release 12.2, as this technology was not used in previous releases.

Typically, the FMW directory will be located in a path such as `/u01/R122_EBS/fs1`, at the same level as the `inst` and `EBSapps` directories. It includes subdirectories such as:

- **EBS_Domain**: The Oracle WebLogic Server domain used to deploy Oracle E-Business Suite. All the domain-specific configuration and log files are located here.

  Each node in a WLS domain has a *Node Manager*, which can be started and stopped using the `$ADMIN_SCRIPTS_HOME/adnodemgrctl.sh` script.

- **wlserver_10.3**: Contains all the binaries and libraries required to support Oracle Weblogic Server.

- **oracle_common**: Contains all the Java Required Files (JRFs) needed by Oracle E-Business Suite.

- **webtier**: Contains the HTTP server instance used by EBS. All the HTTP-specific configuration and log files are located here.

- **Oracle_EBS-app1**: Oracle E-Business Suite is deployed as an Oracle Home under FMW. This directory contains all the configuration files related to oacore, forms, oafm, and forms-c4ws.

Some of these, as well as some other subdirectories, are shown in the following diagram.

Fusion Middleware Directory Structure
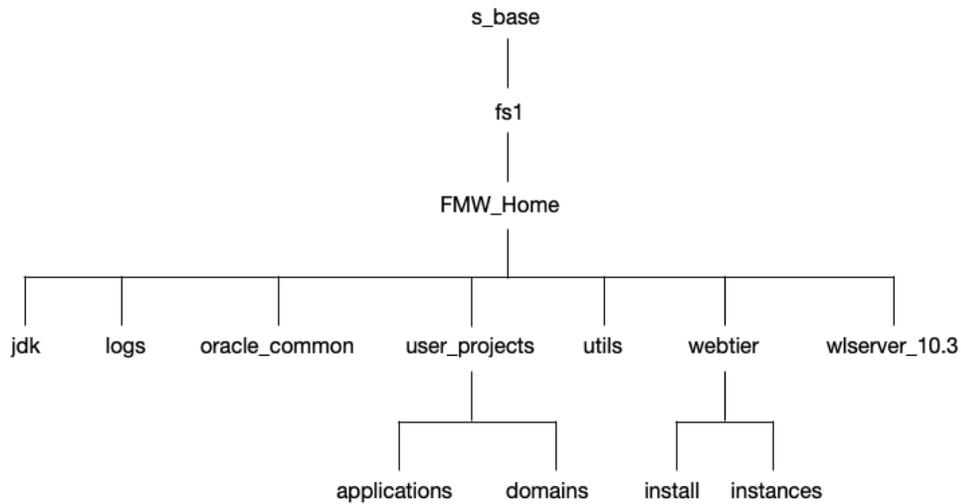
The comn Directory

The **EBSapps/comn** (COMMON_TOP) directory contains files used by many different Oracle E-Business Suite products, and which may also be used with third-party products.

COMMON_TOP Directory Structure
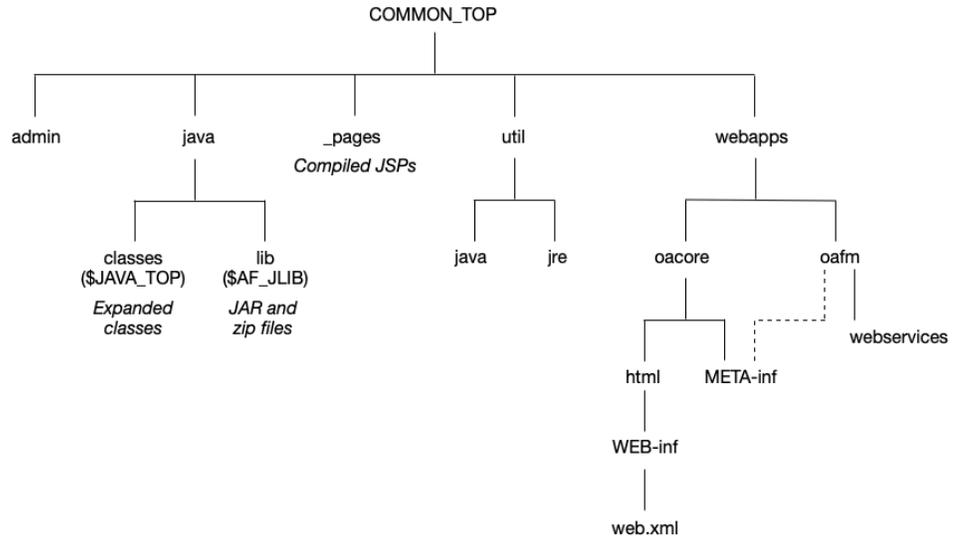
> **Note:** Only selected files and directories are shown in the diagram and discussed below.

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 E-Business Suite 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.

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.

> **Additional Information:** For further details, see *Concurrent Processing Server* in Chapter 1 of this book, and the concurrent processing chapters of *Oracle E-Business Suite Setup Guide*.

The java directory

Rapid Install places all Oracle E-Business Suite class files in the COMMON_TOP/java/classes directory, pointed to by the $JAVA_TOP environment variable. Zip and jar files are installed in the $COMMON_TOP/java/lib directory,

pointed to by the $AF_JLIB environment variable. The top-level Java directory, $COMMON_TOP/java, is pointed to by the $JAVA_BASE environment variable.

The util directory

The *util* directory contains additional utilities that ship with Oracle E-Business Suite. These include the Java Runtime Environment (JRE) and Java Development Kit (JDK).

The appl Directory

Many Oracle E-Business Suite files are stored in the **EBSapps/appl** directory, which is generally known as the APPL_TOP.

APPL_TOP Directory Structure
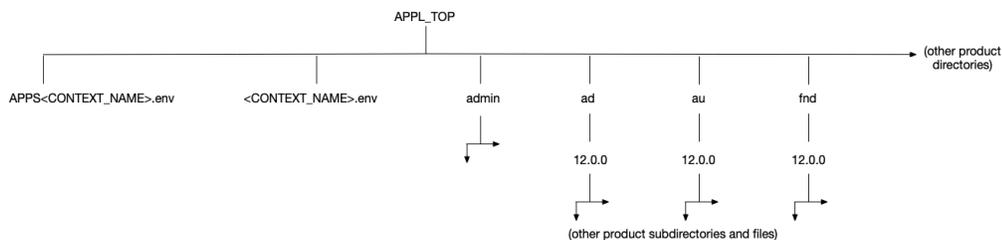
The APPL_TOP directory contains:

- The core technology files and directories.

- The product files and directories (for all products).

- The main Oracle E-Business Suite 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 *Oracle Applications context*, described further in Chapter 5. Its default value is <SID>_<hostname>.

Rapid Install creates a directory tree for every Oracle E-Business Suite product in this APPL_TOP directory, whether licensed or not. Regardless of registration status, *all* Oracle E-Business Suite products are installed in the database and the file system.

    Warning: Do not attempt to delete any files belonging to unregistered or unused products.

Each APPL_TOP directory is associated with a single Oracle E-Business Suite database.

Product Directories

Each product has its own directory under APPL_TOP. The directories are named in accordance with the product's standard abbreviation, such as *gl* for Oracle General Ledger. Within each product's directory is a subdirectory that is named after the base Oracle E-Business Suite release number, 12.0.0. Within this subdirectory, various additional subdirectories are used to store the product-specific 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/R122_EBS/EBSapps/appl, then the value contained in the AD_TOP environment variable is /d01/R122_EBS/EBSapps/appl/ad/12.0.0, and the AD_TOP environment variable points to the <APPL_TOP>/ad/12.0.0 directory.

For the same APPL_TOP, the value of AU_TOP is /d01/R122_EBS/EBSapps/appl/au/12.0.0, and the AU_TOP environment variable points to the <APPL_TOP>/au/12.0.0 directory. The same principle applies to all directories, apart for the admin directory.

Product Files

Each <PROD>_TOP directory, such as <APPL_TOP>/gl/12.0.0, 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 E-Business Suite accesses files in the forms subdirectory under the 12.0.0 directory.

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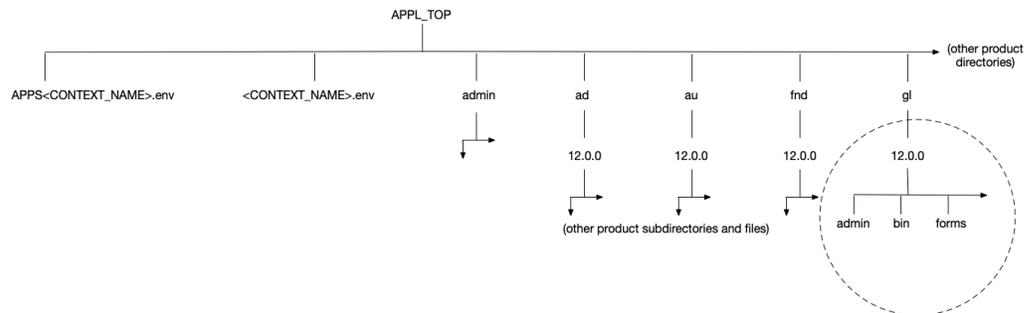
Within each <PROD>_TOP directory, the product's files are grouped into subdirectories according to file type and function. The next figure expands the inset to show the full directory structure for gl.

Detail of gl Directory Structure

```
                                      12.0.0
    ┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐
  admin   bin  forms html  java  lib   log   mds  media mesg  out  patch reports sql   bin
```
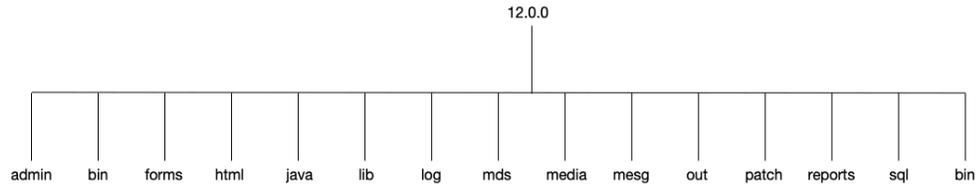
The following table summarizes product subdirectories and the types of files each one may contain.

> **Important:** Not all products use all the subdirectories listed in this table.

Applications Product Subdirectories and File Types

Subdirectory Name	Description
admin	The <PROD>_TOP/admin directory contains *product-specific* files used to upgrade each product. This is in distinction to the <APPL_TOP>/admin directory, which contains upgrade-related files for *all* products.
driver	Contains driver files (.drv files) used in upgrading.
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.

Subdirectory Name	Description
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 JavaServer Page (JSP) files, primarily for HTML-based Applications products.
include	Contains C language header (.h) files that my 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 the $AF_JLIB directory.
lib	Contains files used to relink concurrent programs with the Oracle server libraries. These files include: • 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.

Subdirectory Name	Description
log and out	Contains output files for concurrent programs: • .mgr (main log file for concurrent manager) • .req (log file for a concurrent process) Note that *log* and *out* 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).
media	Contains image files used to support display of text and graphics on the desktop client.
mesg	Contains .msb files (binary message files used at runtime), plus language-specific message files such as the US.msb file for American English and D.msb file for German. The files include the forms messages that are displayed at the bottom of the screen or in popup windows. Concurrent programs also print messages in the log and output files.
patch	Contains patch files used for updates to the data or data model.
reports	Contains Oracle Reports platform-specific rdf binary report files for each product. Reports for each language are stored in subdirectories of the *reports* directory.
resource	Contains .pll files (PL/SQL library files for Oracle Forms), which, like the *plsql* directory files, are later copied to AU_TOP.
sql	Contains .sql files (SQL*Plus scripts) for concurrent processing.

Dual and Non-Editioned File Systems

The description of the file system given so far in this chapter applies, in general terms at least, to earlier 12.x releases as well as 12.2. A key difference in Release 12.2 is the

support needed for the new *online patching* mechanism.

Described in more detail in the Patching and Management tools chapter, the online patching mechanism introduced with Oracle E-Business Suite Release 12.2 uses a *dual file system*. At a given time, one file system (designated 'run') is part of the running system, while the other (designated 'patch') is either having patches applied or standing by in readiness for patch application. The two file systems are often referred to as fs1 and fs2.

> **Note:** It is important to distinguish between the file system (fs1 or fs2)) itself and its current role, which alternates between 'patch' and 'run' with every patching cycle.

As well as fs1 and fs2, there needs to be a *non-editioned file system* (fs_ne), which is used to store files containing data that is needed across all file systems.

Dual File System and Non-Editioned FIle System Directory Structures

```
s_base (s_ebs_base/fs1)                          s_patch_base (s_ebs_base/fs2)

    ├── EBSapps                                      ├── EBSapps
    │      ├── appl                                  │      ├── appl
    │      ├── comn                                  │      ├── comn
    │      └── 10.1.2                                │      └── 10.1.2
    ├── inst                                         ├── inst
    └── FMW_Home                                     └── FMW_Home


s_ne_base (s_ebs_base/fs_ne)

    └── EBSapps
           ├── appl
           │      ├── ad
           │      │     ├── bin
           │      │     │     └── adop (script used to launch adop tool)
           │      │     └── custom
           │      │            └── adop_sync.drv (driver used to sync customizations)
           │      ├── <product name>
           │      └── <product name>
           ├── patch (PATCH_TOP where all patches are downloaded)
           └── log
                  └── adop (consolidated adop logs)
                         └── <adop session ID>
                                └── <context_name>
                                       └── <time stamp>
                                              └── <phase>
```
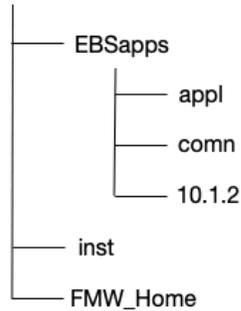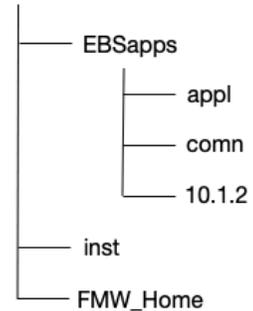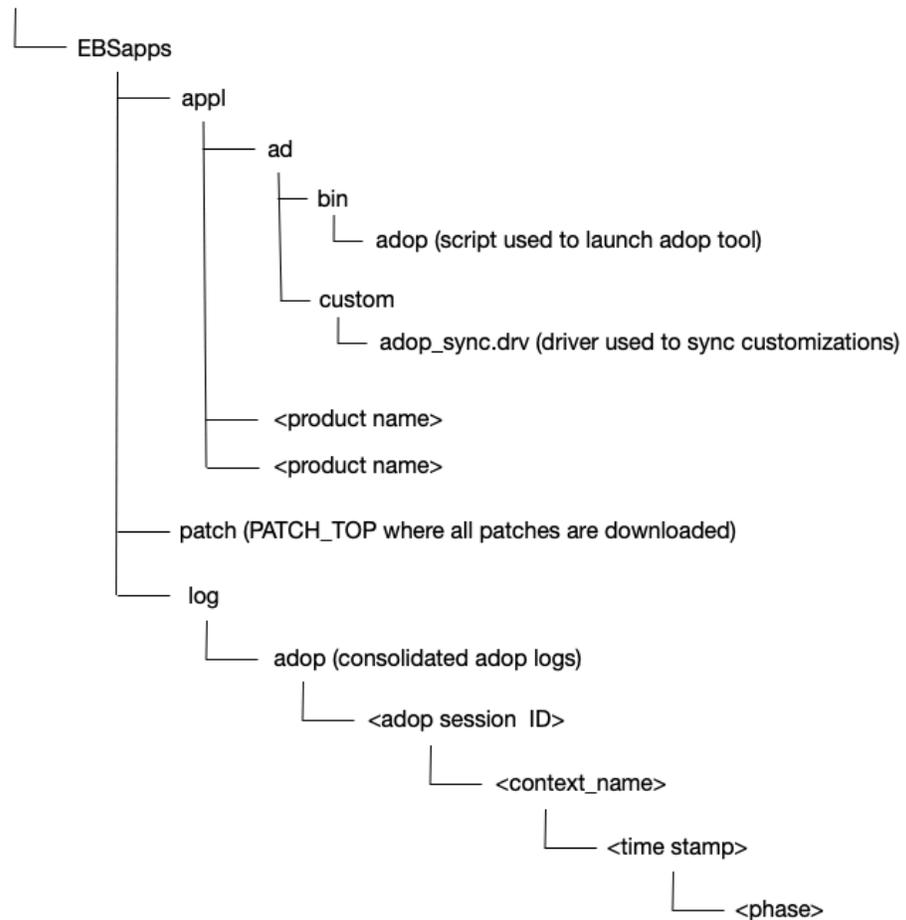
The non-editioned file system is designed to store files which will never be changed by

run and patch file system life cycles.The most notable examples of this are transactional data and certain log files.

Log Files

Oracle E-Business Suite Release 12.2 has seen significant changes to the locations under which log files are stored. In large part, this is because of the introduction of online patching and the additional file systems it employs. The adoption of Oracle WebLogic Server for some configuration management tasks has brought its own log file requirements, over and above those of the traditional AutoConfig tool.

$LOG_HOME

This environment variable and its associated location are used in Release 12.2 as they were in previous releases. Most of the Oracle E-Business Suite log files are stored here, under $LOG_HOME/appl/admin/log. Examples include service control logs, AutoConfig logs, and runtime-generated logs.

A change in Release 12.2 is that HTTP, Oracle WebLogic Server, and concurrent processing log files are not stored under $LOG_HOME as they were in previous releases:

- HTTP log files are located under the native instance home.

- Oracle WebLogic Server log files are located under the domain home.

- Concurrent processing log files are located on the non-editioned file system (fs_ne).

Patching Log Files

The adop log files for online patching are located on the non-editioned file system (`fs_ne`), under:

```
s_ne_base/EBSapps/log/adop/<adop_session_id>/<phase_timestamp>/<context_name>
```

For example, if `s_ne_base` was `/u01/R122_EBS/fs_ne`, the adop log files would be located under:

```
/u01/R122_EBS/fs_ne/EBSapps/log/adop/<adop_session_id>/<phase_timestamp>/<context_name>
```

For more information, see 'The adop Utility' in the Patching Utilities chapter of *Oracle E-Business Suite Maintenance Guide*.

Configuration Log Files

The AutoConfig log files are stored under <INST_TOP>/admin/log/<MMDDhhmm> on the application tier, and <RDBMS_ORACLE_HOME>/appsutil/log/<CONTEXT_NAME>/<MMDDhhmm> on the database tier.

For more information, see the Technical Configuration chapter of *Oracle E-Business Suite Setup Guide*.

Language Files

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

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

> **Additional Information:** For further details, see the *Oracle Globalization Support Guide*.

Core Technology Directories

The admin, ad, au, and fnd directories are the *core technology directories*. They, along with product directories such as ar and gl, are located under the $APPL_TOP directory. For example, /u01/R122_EBS/fs1/EBSapps/appl/admin.

The admin directory

This directory and its subdirectories contain files and scripts used by the AD utilities during upgrade and maintenance processes.

These files and scripts include:

- The adovars.env environment file, which defines certain file and directory locations.

- Scripts run during the upgrade.

- <SID>/log and <SID>/out directories for upgrade, log, and output files respectively.

- A <SID>/restart directory where AD programs create restart files.

The ad (Applications DBA) directory

This directory and its subdirectories contain installation and maintenance utilities, including:

- AD Administration (adadmin)

- AutoConfig (adconfig.sh)

The au (Applications Utilities) directory

This directory and its subdirectories contain product files that are consolidated in a single location for optimal processing. These files include:

- 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 when regenerating the desktop client JAR files, in the java subdirectory

- Certain reports needed by products such as Discoverer, in the reports subdirectory

> **Additional Information:** If applicable, refer to My Oracle Support Knowledge Document 2277369.1, *Oracle E-Business Suite Support Implications for Discoverer 11gR1* .

The fnd (Application Object Library) directory

This directory and its subdirectories contain the scripts and programs that are used as the foundation for all Oracle E-Business Suite products to build data dictionaries, forms and C object libraries.

Environment Settings

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

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

Oracle E-Business Suite Environment Files

Filename	Location	Path	Environment
<CONTEXT_NAME>.env or CONTEXT_NAME>.cmd	11.2.0 ORACLE_HOME	<s_base>/11.2.0	Oracle Server Enterprise Edition
<CONTEXT_NAME>.env or <CONTEXT_NAME>.cmd	OracleAS 10.1.2 ORACLE_HOME	<s_base>/inst/apps/<CONTEXT_NAME>/ora /10.1.2	Oracle tools technology stack
<CONTEXT_NAME>.env or <CONTEXT_NAME>.cmd	APPL_TOP	<s_base>/EBSapps/appl	Applications

Filename	Location	Path	Environment
APPS<CONTEXT_NAME>.env or APPS<CONTEXT_NAME>.cmd	APPL_TOP	<s_base>/EBSapps/appl	Consolidated setup file

> **Note:** <s_base> is the directory where the EBSApps, 11.2.0 (database), inst, and FMW_Home directories exist.

On UNIX, Oracle E-Business Suite includes a consolidated environment file called `APPS<CONTEXT_NAME>.env`, which sets up both the Oracle E-Business Suite and Oracle technology stack environments. When you install Oracle E-Business Suite , 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<CONTEXT_NAME>.cmd`. Running it creates a command window with the required environment settings for Oracle E-Business Suite. 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.

Key Oracle E-Business Suite Environment Variables

Parameter	Description
APPLFENV	The name of the environment file, <CONTEXT_NAME>.env. If you rename the environment file, this parameter setting must be updated.
PLATFORM	The operating system in use. The value (for example, LINUX) should match the value in the $APPL_TOP/admin/adpltfrm.txt file.
APPL_TOP	The top-level directory for this Oracle E-Business Suite installation.

Parameter	Description
NE_BASE	The top-level directory under which non-editioned code is installed (see The Non-Editioned File System in the Online Patching section of the High Availability chapter).
ADMIN_SCRIPTS_HOME	Directory under $INST_TOP that Identifies the location of scripts such as adautocfg.sh, adpreclone.sh, adstrtal.sh, and adstpall.sh.
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 E-Business Suite initial sign-on form. The default is APPLSYSPUB/PUB.
FND_TOP	The path to the Application Object Library directory. For example, /u01/R122_EBS/EBSapps/appl/fnd/12.0.0.
AU_TOP	The path to the Applications Utilities directory. For example, /u01/R122_EBS/EBSapps/appl/au/12.0.0.
<PROD>_TOP	The path to a product's top directory. There is one entry for each Oracle E-Business Suite product.
PATH	Sets the directory search path, for example for $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.

Parameter	Description
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" (required), 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, /u01/R122_EBS/inst/apps/<context_name>/logs/appl/conc.
APPLLOG	The subdirectory for concurrent manager log files. The default is *log*.
APPLOUT	The subdirectory for concurrent manager output files. The default is *out*.
APPLTMP	Identifies the directory for Oracle E-Business Suite temporary files. The default is $INST_TOP/appltmp on UNIX.
APPLPTMP	Identifies the directory for temporary PL/SQL output files.
INST_TOP	Identifies the top-level directory for this instance. For example, /u01/R122_EBS/inst/apps/<context_name>. Introduced with Release 12.0.
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", for example 29-JUL-11.

Parameter	Description
NLS_NUMERIC_CHARACTERS	The National Language Support numeric separators. The default is "." (period and comma).

Most temporary files are written to the location specified by the APPLTMP environment setting, which is set by Rapid Install.

Oracle E-Business Suite products also create 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 set to a database directory defined for PL/SQL file I/O. Refer to My Oracle Support Knowledge Document 2525754.1, *Using UTL_FILE_DIR or Database Directories for PL/SQL File I/O in Oracle E-Business Suite Releases 12.1 and 12.2*.

Rapid Install sets the value of the database directory defined for PL/SQL file I/O and assigns it to the APPLPTMP environment variable.

Some Oracle E-Business Suite 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 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 database directory location for PL/SQL file I/O 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 environment files play a key role in an Oracle E-Business Suite 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 12.2 environment, adovars.env is maintained by AutoConfig, and should not be edited manually.

The adovars.env file includes the following variables:

Variable	Description
AF_JLIB	Indicates the directory to which all Java archive files are copied. For example, /u01/R122_EBS/EBSapps/comn/java/lib.
JAVA_BASE	Indicates the top-level Java directory. For example, /u01/R122_EBS/EBSapps/comn/java.
JAVA_TOP	Indicates the directory to which all Java class files are copied. For example, /u01/R122_EBS/EBSapps/comn/java/classes.
OA_JAVA	Indicates the directory to which all Java archive files are copied. For example, /u01/R122_EBS/EBSapps/comn/java/classes.
OA_JRE_TOP	Indicates the location where the JRE is installed. For example, $FMW_HOME/jdk/jre.
OAH_TOP	Defines the location to which HTML files are copied. For example, $FMW_HOME/Oracle_EBS-app1/applications/oacore.
OAD_TOP	Defines the locations to which context-sensitive documentation files are copied. For example, /u01/R122_EBS/EBSapps/comn.
LD_LIBRARY_PATH	Path used on many UNIX platforms to list the directories that are to be 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 E-Business Suite 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 E-Business Suite 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 employed by Rapid Install.

The fndenv.env file

This file sets additional environment variables used by the Oracle 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 E-Business Suite. This script is located in FND_TOP/usrxit, and is automatically called by fndenv.env. This allows you to compile and link custom Oracle Forms user exits and concurrent programs with Oracle E-Business Suite.

3

Database Features

Introduction

This chapter describes the Oracle E-Business Suite data model, including schemas, Oracle user IDs, and related database server features.

Schemas

A given Oracle database can store the objects associated with a single installation of Oracle E-Business Suite. In general, product *code* objects are stored in the *APPS schema*, whereas product *data* objects are stored in the relevant *base product schemas*. These schemas are described further below.

The EBS_SYSTEM Schema

The AD-TXK Delta 13 release update pack modernized the Oracle E-Business Suite database architecture by introducing a new schema, *EBS_SYSTEM*, which is analogous to the Oracle Database's SYSTEM schema. Prior to the introduction of the EBS_SYSTEM schema, Oracle E-Business Suite installed application objects in the Oracle Database SYS and SYSTEM schemas. Adoption of the EBS_SYSTEM schema obviates the need for any objects owned by Oracle E-Business Suite to reside in the SYS or SYSTEM schemas.

Key characteristics of a Oracle E-Business Suite installation that has been migrated to the EBS_SYSTEM schema include:

- All Oracle E-Business Suite database objects that once resided in the SYS or SYSTEM schemas were migrated to appropriate Oracle E-Business Suite schemas. Depending upon the Oracle E-Business Suite object type and function, the object was migrated to EBS_SYSTEM, APPS, or APPS_NE.

- All Oracle E-Business Suite administration actions (such as running `adop`, `adadmin` and other utilities) will now prompt for the EBS_SYSTEM password instead of the SYSTEM password. Highly privileged operations that were previously run by the

SYS or SYSTEM accounts are now run by EBS_SYSTEM.

- Access to the Oracle Database SYS and SYSTEM accounts, and the Oracle Database server operating system, are no longer required for Oracle E-Business Suite system administrative functions.

> **Additional Information:** To learn more about migrating to the EBS_SYSTEM schema and the procedures for managing privileges in a migrated environment, refer to My Oracle Support Knowledge Document 2755875.1, *Oracle E-Business Suite Release 12.2 System Schema Migration*.

The APPS Schema

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

Utilizing a single schema that has access to all objects avoids cross-product dependencies, and creates a hub-and-spoke access model rather than the spider web model that would otherwise be needed. The APPS schema also improves the reliability of and reduces the time needed for installation, upgrading, and patching, by eliminating the need for cross-product grants and synonyms.

The following code objects are installed in the APPS schema:

- Packages

- Procedures

- Functions

- Triggers

- Views

- Materialized views

- Java classes

- Queues

Base Product Schemas

All data objects for a product are owned by a specific schema for that product, known as the base product schema.

The following objects are installed in the base product schemas:

- Tables

- Sequences

- Indexes

- Constraints

- Queues

Relationship Between APPS Schema and Base Product Schemas

The base product schemas also contain grants from various tables and sequences to the APPS schema, as well as synonyms from the APPS schema to the same objects.

APPS Schema and Base Product Schemas
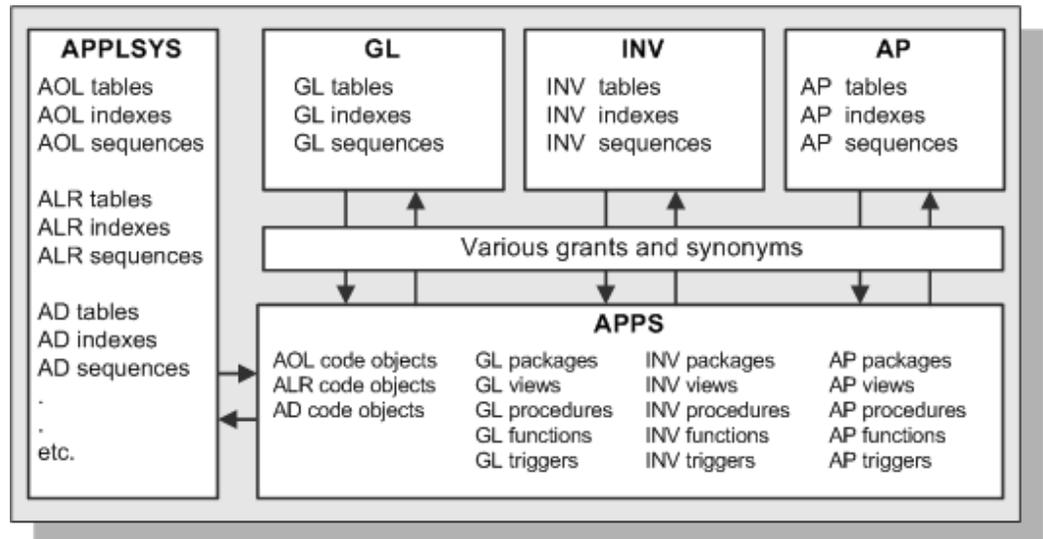
Custom Schema Access

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

> **Warning:** Since the APPS schema has all privileges to all Oracle E-Business Suite 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.

Schemas and 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 E-Business Suite through the Sign-On screen, or when using concurrent processing with Oracle Reports or SQL scripts.

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

Oracle User IDs

Each Oracle E-Business Suite product has a default Oracle user ID, with the product abbreviation used as both the schema name and password (unless you set a different password durining installation). For example, the default Oracle user ID/password combination for Oracle General Ledger is *GL/GL*.

> **Important:** For security, Oracle recommends taking advantage of the option to change the default passwords during installation. However, you should 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 (see earlier figure for details of the schemas).

Space Management

This section discusses how the Oracle database is set up to meet the space management needs of Oracle E-Business Suite. It provides information on tablespaces, firstly outlining the basic tablespaces required, then discussing the traditional tablespace structure used to support Oracle E-Business Suite products, and finally describing the tablespace model that is used as standard with Oracle E-Business Suite Release 12.2.

> **Important:** Oracle E-Business Suite Release 12.2 requires an Oracle
> database block size of 8K. No other block size may be used.

Introduction to Tablespaces

An Oracle database 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 E-Business Suite products. Alternatively, separate temporary tablespaces can, if desired, be created for individual products. Since users access Oracle E-Business Suite 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 E-Business Suite 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.

> **Additional Information:** For further information about tablespaces, see
> the *Oracle 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 E-Business Suite product's tables and indexes. The value of this factor was a percentage of the typical estimated growth rate for Oracle E-Business Suite 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.

Oracle Applications Tablespace Model

Oracle E-Business Suite Release 12.2 utilizes as standard a modern 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 Oracle E-Business Suite products in an OATM environment share a much smaller number of tablespaces, rather than having their own dedicated tablespaces.

Oracle E-Business Suite 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, which are summarized in the list below:

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

- Uses locally managed tablespaces, whose choice of extent management types (autoallocate or uniform) allows more precise control over unused space and hence can help reduce 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 thereby improving runtime performance.

- Makes best use of the restricted number of raw devices available in Oracle Real Applications Cluster (Oracle RAC) and other environments, where every tablespace requires its own raw device.

- Maximizes usefulness of wide disk stripe configurations.

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.

> **Note:** For more information on the Oracle Applications Tablespace

Model, refer to *Oracle E-Business Suite Setup Guide*.

Performance Features

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

Query Optimization

The SQL used in Release 12.2 has been extensively tuned for *cost-based optimization*. In calculating the lowest cost (most efficient) method of executing an SQL statement, the Oracle query optimizer evaluates many factors to calculate the most efficient way to execute a SQL statement. For example, the optimizer considers the available access paths, factoring in statistical information for the tables and indexes that the SQL statement will access. The optimizer also considers *hints*, which are optimization suggestions placed in a comment of the SQL statement.

As part of its operation, the optimizer creates a set of potential execution plans for the SQL statement, based on the available access paths and any hints. It 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. Finally, the optimizer compares the costs of the execution plans and chooses the one with the smallest cost, which means optimum execution characteristics.

For some operations, such as batch processing, Release 12.2 uses cost-based optimization 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 12.2 uses cost-based optimization to achieve the best response time for processing the *first rows* that are accessed by the statement.

Several other Oracle database performance features in Release 12.2 also require use of the cost-based query optimizer.

> **Note:** For further details of optimization, see: *Oracle Database Concepts* and *Oracle Database Performance Tuning Guide*.

Database Resource Manager

The *Database Resource Manager* 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.

> **Additional Information:** For further details, see *Oracle Database Concepts* and *Oracle Database Administrator's Guide*

Database Partitioning

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 containing the partition key can access them instead of the original tables or indexes and thereby reduce the I/O needed.

Partitioning can significantly enhance performance and manageability. For example, the speed of operations that involve copying or deleting data may be greatly improved by the use of partitioned tables. Operations that might have taken hours might now be completed in minutes. This can be useful in data warehouse applications.

> **Important:** Custom partitioning of standard Oracle E-Business Suite tables in Release 12.2 is fully supported for objects that are not already partitioned.

Partitioning should always be planned and tested carefully before being implemented on a critical system. After implementation, it is essential to check that the desired performance benefits have been achieved.

> **Note:** For more information about partitioning, refer to My Oracle Support Knowledge Document 554539.1,*Using Database Partitioning with Oracle E-Business Suite*.

Segment Creation on Demand

This feature allows initial segment creation for nonpartitioned tables and indexes to be delayed until data is first inserted into an object. This is beneficial because while some applications modules need to be delivered with large schemas that contain many tables and indexes, only a subset of these objects may actually be used. Use of delayed segment creation means that empty database objects do not consume any space, reducing disk usage and speeding up installation.

Nologging Operations

The *nologging* Oracle database feature is used to enhance performance in certain areas of Oracle E-Business Suite. For example, it may be 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:** Oracle Database also 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* for details of the *force logging* clause for database and tablespace commands.

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 restored 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 after the nologging operation is complete, 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 Oracle E-Business Suite 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.

Scalability Features

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.

Oracle Real Application Clusters

Oracle Real Application Clusters (Oracle RAC) harness the processing power of multiple interconnected computers. Oracle RAC software called *Oracle Clusterware* 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 Oracle RAC environments, all active instances can concurrently execute transactions against a shared database. Oracle RAC coordinates each instance's access to the shared data, to provide data consistency and data integrity. From a developer's point of view, Oracle RAC enables applications to be scaled to meet increasing data processing demands, without the need to change the application code.

All Oracle E-Business Suite modules can be successfully deployed against a Oracle RAC-enabled 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 an Oracle RAC cluster.

Oracle E-Business Suite supports use of the Oracle Database *Single Client Access Name* (SCAN) feature, which allows all clients to use a single network name (typically defined in DNS) to access an Oracle RAC database. For example, in a non-SCAN two-node cluster the application tier TNS connection descriptor would look something like this:

```
VISION =
(DESCRIPTION=
(ADDRESS_LIST=
(ADDRESS=(PROTOCOL=tcp)(HOST=CNODE1)(PORT=1521))
(ADDRESS=(PROTOCOL=tcp)(HOST=CNODE2)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=VISION)))
```

With SCAN, the equivalent descriptor would be:

```
VISION = (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=CSCAN)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=VISION)))
```

Using SCAN means that clients do not have to be modified when the cluster configuration is changed. The feature also faciliates use of load balancing with an Oracle RAC database.

Automatic Storage Management

Automatic Storage Management (ASM) provides a file system and volume manager dedicated to the storage of Oracle database files. It extends the concepts of disk striping and mirroring, to optimize performance and remove the need for manual I/O tuning.

Edition-Based Redefinition

Traditionally, applications such as Oracle E-Business Suite have had to be unavailable while their associated database objects are patched or upgraded. The *Edition-Based Redefinition* (EBR) feature of the Oracle Database allows this downtime to be avoided.

Every database has at least one *edition*, and can support multiple editions. A new edition is created as a child of an existing edition. The changes are made in the child edition, while the parent edition remains available for use. Changes to data (such as when a patch is applied) are made by writing only to new columns or new tables not seen by the old edition.

EBR allows an application's database objects to be changed without interrupting the

application's availability. For example, an Oracle E-Business Suite patch can be applied while users are logged in and active. This is accomplished by making the changes in the privacy of a new edition.

Key objects used by EBR include:

- *Editioning views* expose a different projection of a changed table to each edition, allowing each to see only its own columns.

- *Cross-edition triggers* propagate data changes made by the old edition into the columns of the new edition, which subsequently replaces the old edition.

For details of how Oracle E-Business Suite utilizes EBR to support online patching, see the Patching and Management Tools chapter of this book. For a detailed description of how to prepare for and use EBR in online patching, refer to *Oracle E-Business Suite Maintenance Guide*.

4

Patching and Management Tools

Introduction

Powerful yet simple-to-use management tools are critical to the usefulness of any applications offering, and one fundamental management need is the ability to apply patches quickly and easily. This chapter surveys the relevant tools provided with Oracle E-Business Suite.

The most important new feature in Oracle E-Business Suite Release 12.2 is the ability to patch a running system, instead of having to take the system down for a significant period of time while the patches are applied. An outline of patching principles and nomenclature is followed by a description of the online patching environment and its associated tools.

This discussion of patching is complemented by the second part of the chapter, which examines the main features of the *Oracle Applications Manager* tool.

Patching Principles and Nomenclature

Patching Principles

As described in more detail later in this chapter, all Oracle E-Business Suite Release 12.2 patches are applied online: that is, to a running system with users logged on and working in their normal way. Use of online patching is mandatory in Release 12.2.

> **Note:** For full details of the commands and options used in online patching, see the Patching section of *Oracle E-Business Suite Maintenance Guide*.

Patch Nomenclature

Oracle E-Business Suite Release 12.2 includes numerous products that are designed to serve a wide variety of needs in a consistent and complementary way. All have

abbreviated names or two, or more usually three, letters. These short names are used in special patch names (see below) and other places. In general, the short names relate to the full product name. Common examples include Applications Technology (ATG), Financials (FIN), Human Resources (HR), and Supply Chain Management (SCM).

You can search a list of product abbreviations and full names within Oracle Applications Manager, using the *License Manager* tool. From the License Manager home page, choose Reports > Licensed Products. From the List of Products, you can view details of licensed, shared, or not licensed products, or search for a particular product abbreviation or name.

> **Additional Information:** To learn more about the License Manager tool, see *Oracle E-Business Suite Maintenance Guide*.

Common Oracle E-Business Suite patching terms include the following.

> **Note:** The specific examples shown in this section are intended solely to illustrate the principles involved, and do not represent current release levels.

- A *release update pack* increments the third section of the release number, for example to 12.2.1 or 12.2.2, and consists of a cumulative aggregation of product family release update packs created across Oracle E-Business Suite since the initial (base) release.

- A *product family release update pack* is an aggregation of patches created for a specific product family after the initial release, and designated by the release number followed by the product short name and a letter. For example, in 'R12.ATG_PF.B. DELTA.3', '12' is the major release number, 'ATG_PF.B' means 'Applications Technology Product Family pack B', and 'Delta.3' means this is the third *rollup patch* on top of Applications Technology Product Family pack B.

- A *codelevel* is essentially a rollup patch with a name that looks like a version number. Higher codelevels include all fixes from lower codelevels, plus additional fixes. For example, R12.AD.B was the second release of AD in Release 12. R12.AD.B. 1 was a higher codelevel, which included all code in R12.AD.B plus more fixes; R12. AD.B.2 was a higher codelevel again, which included all code in R12.AD.B.1, plus further fixes.

Online Patching

Traditionally, Oracle E-Business Suite has needed to be shut down to apply patches and bug fixes of various kinds. Scheduling the sometimes lengthy downtimes needed can prove difficult to combine with requirements for the highest possible availability of business-critical systems. This is exacerbated by the increasing move to consolidate systems into a single instance that may support worldwide operations, and which

consequently does not have a natural downtime window.

The introduction of *online patching* in Oracle E-Business Suite Release 12.2 greatly reduces the need for such downtime. All Oracle E-Business Suite patches in Release 12.2 are applied online.

Key features of online patching include:

- Oracle E-Business Suite patching operations are carried out while the applications are in use and users are online. This is accomplished by means of a copy of the application components that are stored in the database and file system.

- Oracle E-Business Suite patching is performed using the new *adop* (AD Online Patching) utility.

- A short period of downtime is required, but this amounts to little more than a restart of the application tier services: the time the applications are unavailable is measured in minutes rather than hours, and this can be set to be at the most convenient time, for example outside normal business hours.

- Maintenance Mode is not needed or used in Release 12.2.

The patching model used before Release 12.2 was designed to minimize downtime by performing its operations in the shortest time possible, using whatever resources are needed. In contrast, the online patching model is designed to minimize downtime by allowing patching operations to be performed while users remain on the system.

Overview

In traditional patching, application of a patch is a single logical operation. In online patching, it can be thought of as a series of related steps:

1. A copy is made of the code of the running system that resides in both the file system and database.

2. Patches are applied to the copy while users continue to access the running system.

3. The copy becomes the new running system.

4. The original running system (now obsolete) is recycled for use in future patching cycles.

These steps constitute the *online patching cycle*. To implement this mechanism, various changes have been made to the Oracle E-Business Suite infrastructure.

Any method used to create a copy of the running application must take into account that an Oracle E-Business Suite application comprises both code and data, stored in the database and the file system. A key concept is the *edition* as a copy of the application code: the running application executes on the *run edition*, while any patching activity takes place in the *patch edition*. The implementation mechanism uses the Oracle Database *Edition-Based Redefinition* (EBR) feature to cater for database objects, and a dual

file system to cater for objects stored in the file system.

Two complete file systems are always present in an Oracle E-Business Suite Release 12.2 system. As noted above, the run file system is the one currently in use by the running application, while the patch file system is the either being patched or awaiting the start of the next online patching cycle. In other words, the two file systems swap roles with every online patching cycle.
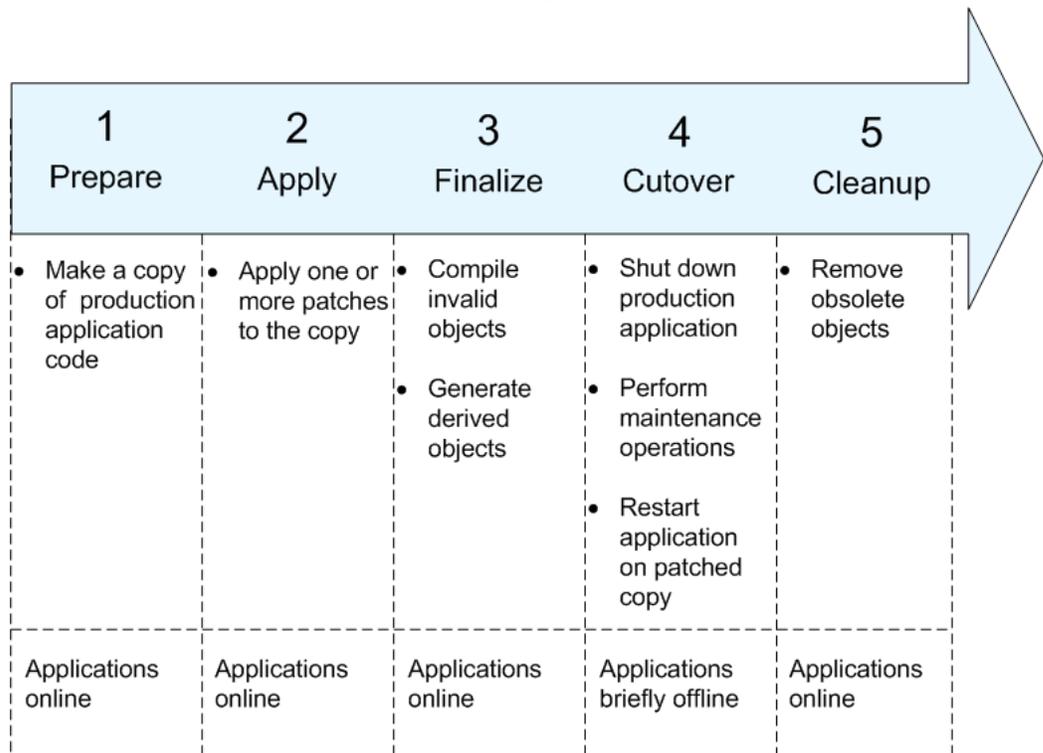
In the database, there is also a run edition and patch edition, but they do not swap roles back and forth as in the file system: the patch edition only comes into existence during a patching cycle, and becomes the new run edition at the end of the cycle. The former database run edition (the old edition) and the obsolete objects it contains are discarded at the end of an online patching cycle, and the space reclaimed during the cleanup phase.

These concepts will all be described in more detail after a discussion of the way they are used in the online patching cycle.

The Online Patching Cycle

The online patching cycle is divided into five phases:

Online Patching Phases

1 Prepare	2 Apply	3 Finalize	4 Cutover	5 Cleanup
• Make a copy of production application code	• Apply one or more patches to the copy	• Compile invalid objects • Generate derived objects	• Shut down production application • Perform maintenance operations • Restart application on patched copy	• Remove obsolete objects
Applications online	Applications online	Applications online	Applications briefly offline	Applications online

Patching Cycle Overview

The key actions in the various stages of the online patching cycle can be summarized as follows:

- **Prepare**
  - Synchronizes patch edition and run edition on the file system.
  - Creates a new patch edition in the database.

- **Apply**
  - Executes patch drivers to update patch edition.
  - Patches applied: can be one or many, including customizations.

- **Finalize**
  - Compiles invalid objects.
  - Generates derived objects.

- **Cutover**

- Configures patch edition file system to be the new run edition file system.

- Configures patch edition of database to be the new run edition.

- Restarts application tier services.

- **Cleanup**
  - Delete obsolete code and seed data to recover space.

Each of the phases will now be discussed in more detail.

Prepare

The following actions are taken in this phase.

On the *file system*:

1. The patch file system is synchronized with the run file system.

2. The patch edition files become an exact copy of the run edition files.

   > **Note:** By default, synchronization is *incremental*: that is, only files that were changed in the last patch application are copied.

In the *database*:

1. A patch edition is created in the database.

2. All code objects in the patch edition begin as pointers to code objects in the run edition. Code objects in the patch edition begin as lightweight "stub objects" that point to the actual object definitions, which are inherited from earlier editions. Stub objects consume minimal space, so the database patch edition is initially very small in size.

3. As patches are applied to the patch edition, code objects are *actualized* (have a new definition created) in that edition.

   > **Note:** Storage objects (such as transaction tables) are *not* copied.

Apply

The following actions are taken in this phase:

1. Patches are applied to the patch edition. During this process, any changed stub objects will become actual code objects in the patch edition.

2. The changes introduced by the patches are made in the isolation of the patch edition.

- Changes to *code objects* are only visible in the patch edition of the file system and database.

- Changes to *tables* are stored in new columns or rows that are only visible to the patch edition.

> **Note:** At this point, users still remain connected to the application and performing their work.

Finalize

This phase is used to perform the final operations that can be executed while the application is online:

1. Invalid objects are compiled.

2. Derived objects are generated.

3. Any actions that must be performed at cutover are pre-computed and stored for quick execution at that time.

Cutover

This phase involves:

1. Shutdown of application tier services.

2. Execution of any required cutover actions to maintain non-editioned objects and data.

3. Configuration of the patch edition of the file system as the new run edition.

4. Configuration of the patch edition of the database as the new run edition.

5. Restart of application tier services on the new run edition.

Although the cutover phase does require a short period of application tier services downtime, the online patching cycle can be paused for as long as required prior to running this phase. You could, for example, add such a pause to ensure that the downtime period will be outside business hours.

> **Note:** The database remains open throughout the entire online patching cycle, including cutover.

Cleanup

The following database actions are taken in this phase, which occurs after users have been brought back online to the newly-patched application:

1. Old code objects that are no longer visible in the run edition are dropped.

2. Old data (rows or columns) that is no longer visible in the run edition is deleted or dropped.

3. Old database editions that no longer contain actual objects are dropped.

Database Implementation

Creating a copy of the database part of the running system has been accomplished by taking advantage of the Oracle Database *Edition-Based Redefinition* (EBR) feature. This allows an application to efficiently store multiple copies of its application definition in the same database, and thereby enables online upgrade of the database tier.

The term *edition* refers to the isolation mechanism that allows pre-upgrade and post-upgrade schemas to co-exist. The simplest way to think of an edition is as a separate (isolated) copy of all database code objects that are changed by a patch.

The three types of database edition are:

- **Run**: Online users connect to this edition, which is always the default database edition. The run edition will always exist.

- **Patch**: Patching tools connect to this edition. A child of the run edition, the patch edition exists *only* while a patching cycle is in progress.

- **Old**: When a patch edition is promoted to be the run edition, the previous run edition is now regarded as an old edition. There may be zero or more old editions at a given time. They are discarded when a full cleanup (described later) is performed. You cannot connect to an old edition.

File System Implementation

Creating and using two file systems allows one (*run file system*) to be part of the running system, while the other (*patch file system*) is either being patched or waiting to be patched during the next patching cycle.

> **Note:** The file systems are designated *File System 1* and *File System 2* in Rapid Install. These are abbreviated to *fs1* and *fs2* respectively.

The two file systems are sometimes referred to as a *dual file system*, and swap roles at the end of each patching cycle. That is, the file system that has just been patched is put into use as part of the running system (becoming the new run file system), and the previous run file system takes over the the role of patch file system (in readiness for commencement of the next patching cycle).

> **Important:** The existence of the dual file system has significant
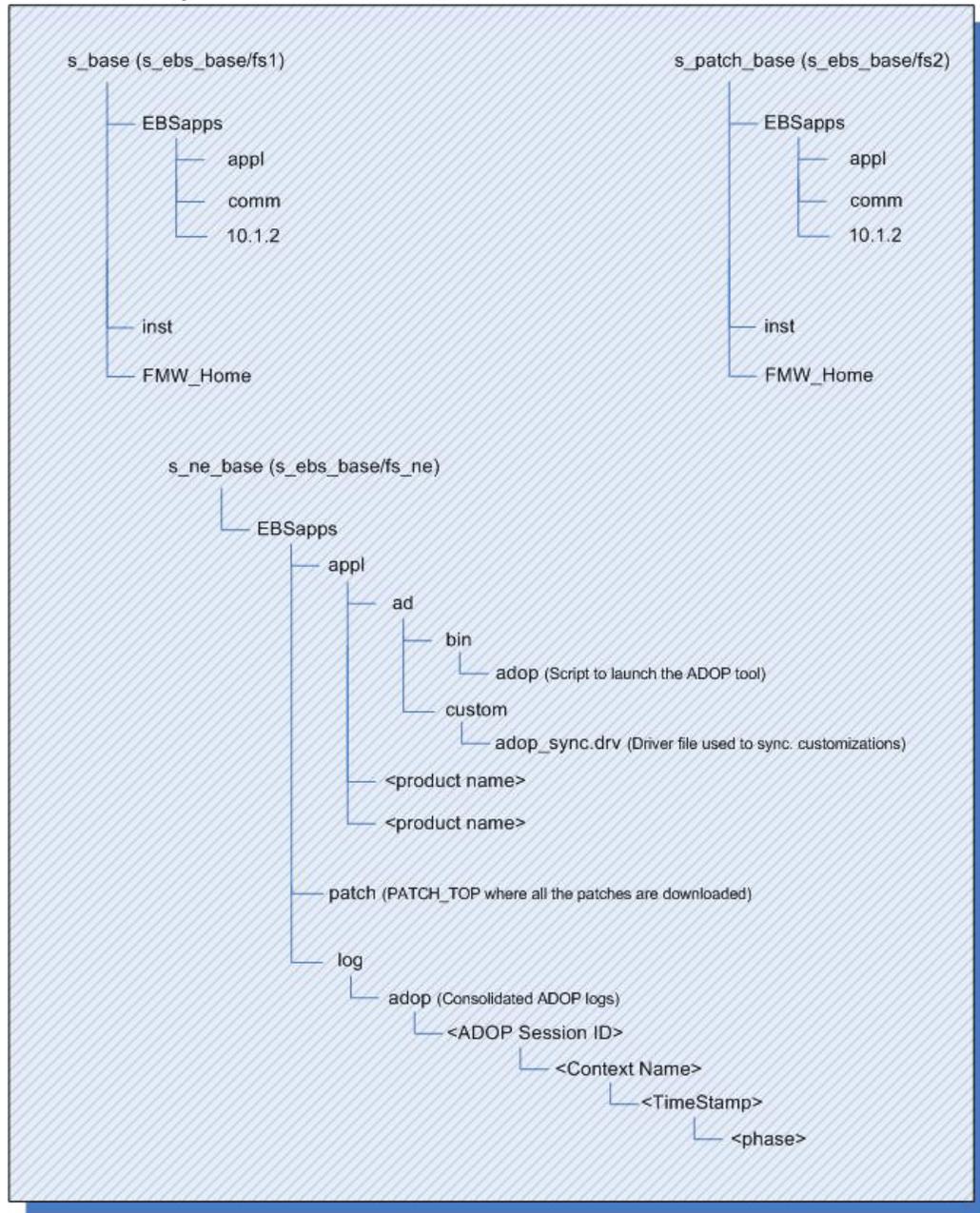
implications for general (non-patching) maintenance activities. For information on understanding how to decide which file system to run AD tools from, refer to: Choosing the Correct File System For Maintenance Tasks, *Oracle E-Business Suite Maintenance Guide* (Chapter 7, page 7-1).

However, two file systems are not sufficient to meet all the practical needs of an online patching environment for Oracle E-Business Suite. A third file system, described in the next section, is also required.

Non-Editioned File System

The dual file system approach caters for application code, but applications also use the file system to read and write business data. In Release 12.2, application data files are stored in a third area, the *non-editioned file system* (fs_ne), which is used to store data that is needed across all file systems. Non-editioned files are not copied or moved during patching: their location remains constant across online patching cycles.

The Three File Systems: fs1, fs2, and fs_ne

```
s_base (s_ebs_base/fs1)                              s_patch_base (s_ebs_base/fs2)

    ├── EBSapps                                          ├── EBSapps
    │      ├── appl                                      │      ├── appl
    │      ├── comm                                      │      ├── comm
    │      └── 10.1.2                                    │      └── 10.1.2
    │
    ├── inst                                             ├── inst
    └── FMW_Home                                         └── FMW_Home


        s_ne_base (s_ebs_base/fs_ne)

            └── EBSapps
                   ├── appl
                   │      ├── ad
                   │      │     ├── bin
                   │      │     │     └── adop (Script to launch the ADOP tool)
                   │      │     └── custom
                   │      │            └── adop_sync.drv (Driver file used to sync. customizations)
                   │      ├── <product name>
                   │      └── <product name>
                   │
                   ├── patch (PATCH_TOP where all the patches are downloaded)
                   │
                   └── log
                          └── adop (Consolidated ADOP logs)
                                 └── <ADOP Session ID>
                                        └── <Context Name>
                                               └── <TimeStamp>
                                                      └── <phase>
```

The non-editioned file system is therefore completely separate from file system 1 and file system 2.

Context Variables for Online Patching File Systems

Several AutoConfig context variables support the file systems used in online patching. For example, the *s_ne_base* context variable stores the location of the non-editioned file system, and AutoConfig sets the environment variable $NE_BASE to the corresponding

value. AutoConfig also sets the environment variables $RUN_BASE and $PATCH_BASE, so you can easily tell which is currently the run file system and which the patch file system.

Files Stored in the Non-Editioned File System

The non-editioned file system is designed to store files that contain transactional data and reference data. Examples include: import, export files, general log files, and concurrent manager log and out files.These are all examples of files that are not modified during an online patching cycle.

More specialized examples include:

- Batch upload and download files.

- Files used to transfer transactional data from processes external to Oracle E-Business Suite (for example, where a third party order entry system delivers orders via order import files).

- Files containing transactions that are needed across all file systems.

> **Note:** The non-editioned file system is not designed to store shared files, because initially identical files can become non-identical during patching life cycles. Nor is it designed to store code, which is editioned (one copy can be in the run file system while the other is in the patch file system).

Concurrent Processing and the Non-Editioned File System

As noted above, concurrent processing in particular makes significant use of the non-editioned file system, with all concurrent manager log and out files being stored there. This has some additional implications, as described below.

Before Release 12.2, there were two choices for the location of log and out files. The first was $APPL_TOP, and the second was $APPLCSF (which pointed to $APPL_TOP/admin/$TWO_TASK). The location used was determined by whether the $APPLCSF environment variable was set.

With Release 12.2, the $APPL_TOP contains only code and configuration files. Therefore, it is mandatory for the $APPLCSF environment variable to be set. Its default value points to the non-editioned file system, specifically to `<s_ne_base>/inst/<EBS_Instance_ID>/logs/appl/conc`: for example, `/u01/R122_EBS/fs_ne/inst/VIS_app101/logs/appl/conc`.

In addition, a new environment variable, $APPLLDM, has been introduced to provide the option of organizing storage of log and out files on a product-specific basis.

In Release 12.2, $APPLLDM environment variable can have multiple possible values. The two main ones are 'single' and 'product'.

If $APPLLDM is set to 'single', concurrent processing and AD Administration log files

will both be stored under $APPLCSF/log. Concurrent processing out files will be stored under $APPLCSF/out. This is the default setting.

If $APPLLDM is set to 'product', the concurrent processing log and out files will be written to product-specific directories under $APPLCSF. This is in contrast to previous releases, where the directories were under $APPL_TOP. For example, concurrent manager log files will be located under $APPLCSF/<product>/log, AD Administration log files will be located under $APPLCSF/log (i.e. the same location as with the 'single' setting), and PO reports will be located under $APPLCSF/po/out.

Beginning with Release 12.2.6, Oracle E-Business Suite provides additional storage strategies for management of large numbers of concurrent processing log and output files. For example, concurrent request log and out files can be organized by date or by user name. These storage strategies are called *schemes*. For more information, refer to the *Oracle E-Business Suite Setup Guide*.

In summary, the default behavior has been designed to be exactly the same as in previous releases: that is to say, if you did not set $APPLCSF before, and you do not set it or $APPLDLM now, you will not notice any difference in concurrent processing log file management or storage.

Editioned and Non-Editioned Objects

EBR automatically manages versioning of objects that support editioning, such as code objects. However, not all objects can be editioned: most notably, this includes transactional data, of which there is only ever one copy.

Oracle E-Business Suite Release 12.2 introduces a new concept, the *logical view* of the data model. This is implemented via *synonyms* and *editioning views*, which isolate the running application from changes to the data model that may be introduced by a patch. That is to say, data model changes are guaranteed to not affect the running application. Data model changes are implemented as new columns on the table, which are exposed to the patch edition of the application via the editioning view. New tables can be introduced by an extension of the same principle.

Crossedition Triggers

A new type of object, the *crossedition trigger*, is used to synchronize data as part of the online patching process. These triggers provide the logic to synchronize and transform data between the run edition and patch edition storage columns. The result is that new transactions entered into the system are patched in place.

More specifically, crossedition triggers allow the run edition to signal that a data update is required. They fire in response to an insert into, delete from, or update of, FND_TABLE. For example, suppose that a patch updates a column called DESCRIPTION from upper case to mixed case. Editioning views project different views of the table to the run and patch editions, so the running application still sees the column data as upper case, while the patched application sees the column data as mixed case. The updated column is maintained by a crossedition trigger.

In summary, crossedition triggers are used to upgrade both existing data and ongoing

changes that occur while the run edition remains in use.

**Editioning Views and the Application Data Model**

Editioning views can be thought of as providing a *cover layer*, or logical representation of the data, on top of the physical representation. All code (Oracle E-Business Suite, custom, or third-party) must access Oracle E-Business Suite data via the cover layer: accessing the data model via the physical layer may result in obsolete data been returned.

Logical View of the Data Model
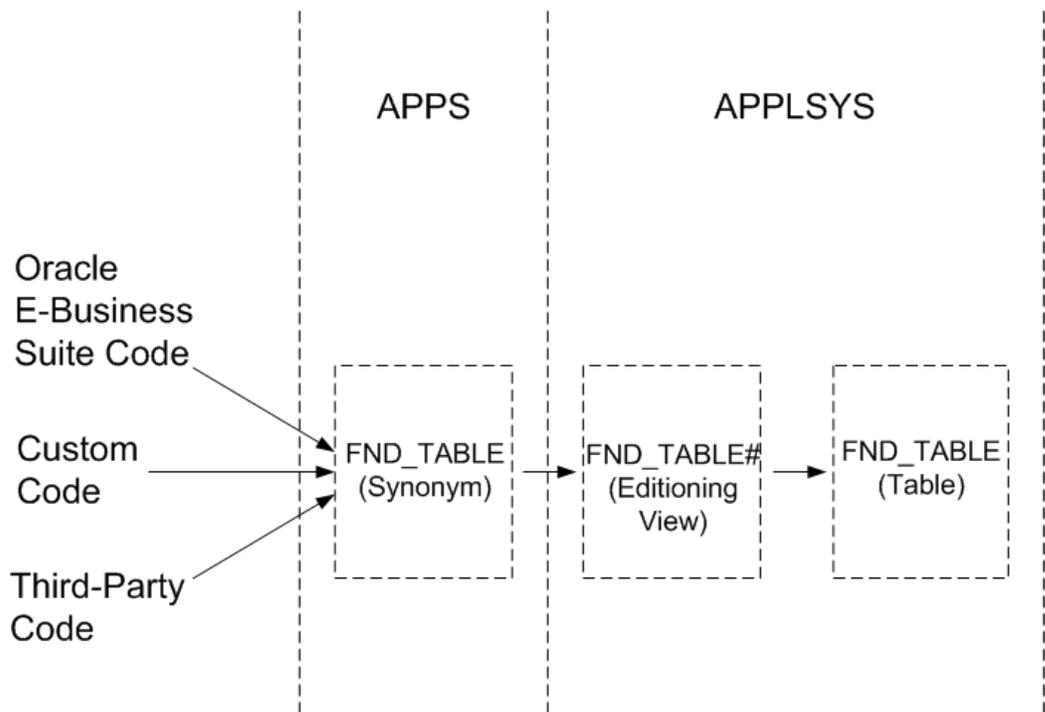
> **Note:** The implementation of EBR for Oracle E-Business Suite requires all code to access the data model via the APPS synonym, which points to the editioning view (logical model).

Database Editioning Views
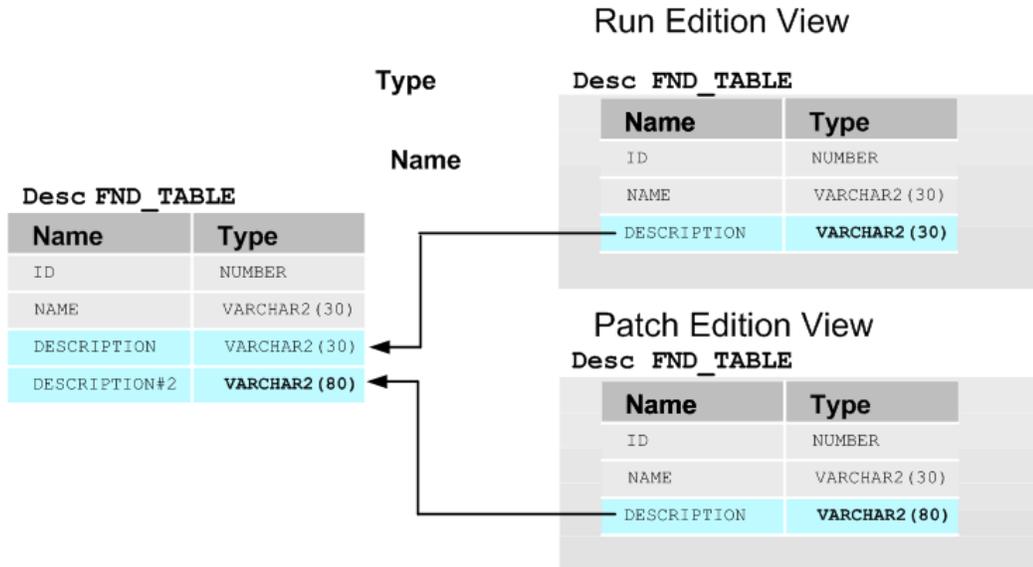
> **Important:** Any code accessing the physical model risks accessing obsolete columns.

Preparing for Online Patching

> **Important:** Upgrading to Oracle E-Business Suite Release 12.2 is *not* done with online patching enabled. Online patching can only be used after you complete the upgrade to Oracle E-Business Suite Release 12.2.

The Release 12.2 upgrade process installs a logical view of the data model that all code must use to access Oracle E-Business Suite, and automatically upgrades seed data tables to support editioned data storage. Several steps can be taken to ease the transition to a Release 12.2 online patching environment. These are divided into pre-upgrade and post-upgrade categories, and mentioned briefly below.

Pre-Upgrade

Ahead to the actual upgrade to Oracle E-Business Suite Release 12.2, it is highly advisable to prepare for the move to an online patching environment by running the three *online patching analysis reports*, Available for all supported pre-12.2 releases, these highlight all cases in Oracle E-Business Suite code or custom code where a non-editioned object references an editioned object.

The reports are:

- **Summary Report**: `ADZDPSUM.sql` provides the following information:

1. Schemas that *will* be editioned upon online patching enablement.

2. Schemas that *will not* be editioned upon online patching enablement.

3. Unregistered schemas that depend on Oracle E-Business Suite objects.

- **Manual Fixes Report**: `ADZDPMAN.sql` lists any manual fixes required.

- **Auto Fixes Report**: `ADZDPAUT.sql` lists any automatic fixes required.

    The reports are run as follows:

```
sqlplus system/manager@DB @$AD_TOP/sql/ADZDPSUM.sql
sqlplus system/manager@DB @$AD_TOP/sql/ADZDPMAN.sql
sqlplus system/manager@DB @$AD_TOP/sql/ADZDPAUT.sql
```

> **Note:** For further details of the online patching analysis reports, see
> *Enabling Online Patching* in the *Oracle E-Business Suite Online Patching*
> chapter of *Oracle E-Business Suite Upgrade Guide: Release 12.0 and 12.2*.

Any compliance issues in Oracle E-Business Suite code are automatically fixed by the upgrade to Release 12.2. With regard to custom code, Oracle recommends running and studying the report before commencing the upgrade, to determine the amount of effort that will be needed for compliance. Most non-compliance issues in custom code can be fixed in the 12.2 upgrade simply by registering the custom schema with Oracle E-Business Suite. Only a small number of issues may require manual intervention.

Post-Upgrade

Having completed the actual upgrade, it is important to bear in mind several key aspects of the online patching environment, including identification of suitable new coding standards to ensure compliance of any new customizations. Also, patch editions should not be treated as a testing environment: patches should still be tested on a test system rather than the production system. New test scenarios may need to be considered, given that users can now be online during patch application.

For full details of the upgrade process, including online patching analysis reports and other pre-upgrade checks and reports, see the applicable upgrade guide: either *Oracle E-Business Suite Upgrade Guide Release 11i to 12.2* or *Oracle E-Business Suite Upgrade Guide Release 12.0 and 12.1 to 12.2*.

Online Patching Tools

Patching is performed by running the *adop* (AD online patching) utility. You run adop instead of the adpatch utility that was used in releases prior to 12.2.

The adop tool orchestrates the entire patching cycle, and can be executed by specifying phases individually or collectively. For example, `adop phase=prepare`, `adop phase=apply`, and `adop phase=cutover` is the equivalent of `adop phase=prepare,apply,cutover`. In both of these examples, only a *single* session id

would be created.

> **Note:** For full details of adop features and options, refer to the Patching part of *Oracle E-Business Suite Maintenance Guide*.

Hotpatch Mode

The *hotpatch* mode of online patching provides the capability to apply changes to the run edition while the run edition is in active use. However, this option can only be used to apply patches that provide explicit support for it. The readme of a patch will state if this is the case.

If hotpatch mode is used with patches that do not support it, the following issues may occur.

- **Possible runtime transaction failure because of invalid objects** - Although some objects may automatically revalidate, a hotpatch operation may not yet have finished making all the necessary object changes. This will result in associated user transactions continuing to fail until the hotpatch is complete.

- **Possible runtime transaction failure due to loss of PL/SQL package state** - Many PL/SQL packages contain global variables which the database considers to be "package state". If a database session has accessed a package containing package state at any time before a hotpatch operation, and then accesses the same package after the package has been invalidated, the session will get the error "ORA-04068: existing state of packages has been discarded" and the transaction will fail.

- **Application code and database objects may be temporarily inconsistent** - If a patch loads new seed data, each uploaded change will be visible to the running application as it is committed. However, while the seed data is being loaded, only some of the necessary changes may have been committed at any given point. This may lead to data integrity problems if the seed data is accessed during the loading operation.

- **Tables that are patched will be temporarily inconsistent** - The process of patching a table involves creating new columns, regenerating the editioning view, populating new columns via cross-edition triggers, and recreating indexes to reference the new columns. Until this sequence of operations is complete, data in the table may appear to be missing or incorrect, or not have correct indexing. Full execution of the table patching steps are spread out over several adop phases, so the table may be in an inconsistent state for the entire duration of the hotpatch operation. It is therefore not safe to access any table from the edition that is being patched.

- **Code and data cached in application tier server memory may be inconsistent with changes made by the hotpatch operation** - To provide optimum performance, pplication tier servers load code objects, seed data and transaction data into

memory caches. After the hotpatch operation is complete, and until the application tier servers are restarted, this cached information may be out of date and lead to various types of failure or corruption.

- **Runtime processing may hold long-term locks on objects, causing the hotpatch operation to fail** - Some database objects in active use by the running application cannot be changed by a hotpatch operation without an exclusive lock first having been obtained on the object (code or data). Such runtime processing includes long-running concurrent programs, or background services that may hold locks on code objects indefinitely. Consequently, the hotpatch operation may fail because it cannot obtain the exclusive lock required to apply some change.

- **Other reasons**

  - In addition to the above issues, it is also possible that a patch will not work in hotpatch mode for unspecified reasons.

In general, Oracle E-Business Suite Development does not test patches in hotpatch mode. Any failures, corruptions, or other errors that result from unauthorized use of hotpatch mode cannot be supported. Consequently, you should only apply a patch in hotpatch mode if support for doing so is explicitly stated in the patch readme.

Online Patching and Seed Data

Oracle E-Business Suite *seed data* is data stored in database tables that affects the behavior of the applications, and is patched as needed by Oracle E-Business Suite Development. In an online patch environment, patches cannot be allowed to modify the seed data that is visible to the running application.

Online patches are prevented from modifying runtime seed data by the use of *editioned data storage*. This involves creation of a (patch) copy of the seed data, which is stored in the same table. The patches that are applied interact only with this copy, while the run edition only interacts with a private copy (which is eventually deleted as part of the cleanup phase).

The running application uses the run edition copy of seed data, while patches may update the patch edition copy of seed data in isolation. The two copies are isolated, except that seed data changes made by the running application are synchronized to the patch edition copy.

In terms of editions and seed data, the *run edition*:

- Always operates on a private copy of the seed data

- Is never modified by patch application

In contrast, the *patch edition*:

- Runs the seed data loader

- Prepares the relevant table for patching

- Copies all table rows

- Loads seed data changes into the (patch) copy

Updates to seed data in the run edition are automatically propagated to the patch edition by the use of crossedition triggers.

Environment Variables

The following table shows examples of the environment variables that are used in online patching. These are all set automatically.

Environment variables and example values

Environment Variable	Example Value on Run File System	Example Value on Patch File System
FILE_EDITION	run	patch
TWO_TASK	vision	vision_patch

APPS_JDBC_URL	`jdbc:oracle:thin:@`
	`(DESCRIPTION=`
	`(ADDRESS_LIST=`
	`(LOAD_BALANCE=YES)`
	`(FAILOVER=YES)`
	`(ADDRESS=PROTOCOL=tcp)`
	`(HOST=testsys1.`
	`example.com`
	`(PORT=1521)))`
	`(CONNECT_DATA=`
	`(SERVICE_NAME=vision))`
	`)`
		`jdbc:oracle:thin:@`
		`(DESCRIPTION=`
		`(ADDRESS_LIST=`
		`(LOAD_BALANCE=YES)`
		`(FAILOVER=YES)`
		`(ADDRESS=`
		`(PROTOCOL=tcp)`
		`(HOST=testsys1.`
		`example.com)`
		`(PORT=1521)))`
		`(CONNECT_DATA=`
		`(SERVICE_NAME=ebs_patc`
		`h)`
		`(INSTANCE_NAME=vision)`
		`))`

AD_APPS_JDBC_URL	`(DESCRIPTION=`
	`(ADDRESS_LIST=`
	`(LOAD_BALANCE=YES)`
	`(FAILOVER=YES)`
	`(ADDRESS=`
	`(PROTOCOL=tcp)`
	`(HOST=testsys1.`
	`example.com)`
	`(PORT=1521)))`
	`(CONNECT_DATA=`
	`(SERVICE_NAME=vision))`
	`)`
		`(DESCRIPTION=`
		`(ADDRESS_LIST=`
		`(LOAD_BALANCE=YES)`
		`(FAILOVER=YES)`
		`(ADDRESS=`
		`(PROTOCOL=tcp)`
		`(HOST=testsys1.`
		`example.com)`
		`(PORT=1521)))`
		`(CONNECT_DATA=`
		`(SERVICE_NAME=ebs_patc`
		`h)`
		`(INSTANCE_NAME=vision)`
		`))`

Environment Variable	Example Value on Run File System	Example Value on Patch File System
RUN_BASE	/u01	/u02
PATCH_BASE	/u02	/u01

> **Note:** These variables and values are all created automatically. No user action is required to set them, and they should not be changed manually.

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 application tier file system (shared APPL_TOP in earlier releases) must be employed to ensure the files are created in a single, centralized location.

Using Distributed AD

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

While using adop 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 adop 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).

> **Additional Information:** See *Oracle E-Business Suite Maintenance Guide* 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.

Oracle Applications Manager Features

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

Among other Oracle 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 into Oracle E-Business Suite, and can be used in conjunction with a number of optional tools and features described later in this chapter: including *Oracle Application Management Pack for Oracle E-Business Suite*.

The *Applications Dashboard* provides an overview of the key features of an Oracle 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.

Oracle Applications Manager Dashboard

Applications Dashboard   Site Map

Applications Dashboard

Navigate to Application Services [▼] [Go]

[**Overview**] Performance   Critical Activities   Business Flows   Security   Software Updates

Applications System Status

Data Retrieved: 20-Jul-2019 02:06:59

Host	Platform	Host Status	Admin	Database	Concurrent Processing
RWS	Linux x86-64 (64-bit)	✔	✔	✔	✔

Configuration Changes (last 24 hours)

Data Retrieved: 20-Jul-2019 02:06:59

Patches Applied	0
Site Level Profile Options	0
Applications Context Files Edited	0

System Alerts

Data Retrieved: 20-Jul-2019 02:06:59

New Alerts	7
New Occurrences	7
Open Alerts	0
Open Occurrences	0

Web Components Status

Data Retrieved: 20-Jul-2019 08:00:00

Servlet Agent	✔	Up
JSP Agent	✔	Up
TCF	✔	Up

User Initiated Alerts

Data Retrieved: 20-Jul-2019 02:06:59

New Alerts	0
New Occurrences	0
Open Alerts	0
Open Occurrences	0

☑ TIP The information shown above (with the exception of Web Components Status section) is retrieved from the system periodically. To retrieve up-to-the-minute data, please use the refresh icon for the desired section. Please see Help for more details.

A more detailed view of the Oracle E-Business Suite 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. Previous configuration settings can easily be compared with the current settings, allowing changed settings to be identified and rolled back as necessary.

Site Map - Administration

Applications Dashboard | **Site Map**

Site Map

| **Administration** | Monitoring | Maintenance | Diagnostics and Repair |

System Configuration

Hosts
AutoConfig
License Manager
Fusion Middleware Control
Weblogic Administration Console

Application Services

Generic Services
Request Processing Managers
Transaction Managers
Parallel Concurrent Programming Setup

Workflow

Home
Work Item Metrics
Agent Activity
Background Engines
Notification Mailer
Service Components
Purge

Concurrent Requests

Submit New
Pending
Running
Completed (Last Hour)

Service Fulfillment Manager

Service Fulfillment Manager

Others

Applications Manager Log
Knowledge Base

☑ **TIP** Only the items to which you have access are clickable.

Additionally, Oracle Applications Manager helps you track configuration changes, such as recently altered site-level profile option settings. It can also helps 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, Forms listeners, 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.

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.

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.

The Monitoring tab provides key performance metrics related to online, batch, and workflow activity. Additionally, system administrators can now subscribe to receive automated alerts when concurrent requests run longer or wait longer than specified thresholds.

Site Map - Monitoring

Monitoring support is provided for application tier services such as concurrent managers and Web services. 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.

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.

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.

Site Map - Maintenance

The *Patch Wizard* tool built into Oracle Applications Manager can recommend patches for your Oracle E-Business Suite system. The wizard takes patch data downloaded from Oracle, analyzes that data against the Oracle E-Business Suite 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.

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 (AD) utilities: access to other parts of Oracle E-Business Suite is not allowed.

Diagnostics and Repair

Oracle Applications Manager for Oracle E-Business Suite Release 12 provides troubleshooting wizards that 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, guides the administrator through the steps needed to bring GSM back up.

Site Map - Diagnostics and Repair

The Applications Dashboard provides a gateway into Oracle Diagnostics for Oracle E-Business Suite, offering graphical views of diagnostic test executions and failures, summarizing test results, and providing access to detailed test reports. Keyword searches can be performed on the test repository to locate relevant tests, which can then be grouped and executed directly from the same interface.

Additionally, system administrators can schedule and run diagnostics tests as batch programs, and output from all diagnostic interfaces can be captured and packaged for shipment to Oracle Support quickly 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.

Finally, the system alerting infrastructure has been enhanced with a new flood control mechanism to prevent multiple instances of a given alert from overwhelming the alert system. Additionally, alerts have been enhanced to allow summary and context information to be retrieved in XML format via standard business event integration mechanisms.

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.

Oracle Applications Manager Support Cart

Applications Dashboard | Site Map

Support Cart Contents

| **Description** | Applications Signature | Other Information Collected |

TAR Number [＿＿＿＿＿＿＿＿＿＿＿＿]

Details [text area]

☑ TIP Please use this section to provide the following information:
(1) Problem description
(2) Steps to reproduce the issue
(3) System, Platform, Environment specific information and
(4) Error messages.

Update   Reset

Restore Cart

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.

Extending Oracle Applications Manager Functionality

Oracle Applications Manager can be used in conjunction with *Oracle Enterprise Manager Grid Control* via the supporting *Oracle Application Management Suite for Oracle E-Business Suite*, to create a comprehensive, top-down approach to system monitoring and administration.

Oracle Application Management Suite for Oracle E-Business Suite extends Oracle Enterprise Manager Grid Control to help monitor and manage Oracle E-Business Suite systems. Features include simplified administration, hot cloning (where the database does not need to be shut down), and management of advanced topologies.

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 E-Business Suite products has increased significantly, with many Oracle E-Business Suite 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 though 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 application tier service management infrastructure has several benefits:

- The service processes no longer need to be manually and individually started and monitored by Oracle E-Business Suite 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

5

Technical Configuration Tools

Introduction

Configuring a new installation of Oracle E-Business Suite 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 E-Business Suite environment. A fresh install of Release 12.2 includes AutoConfig as a standard (and required) configuration management tool. AutoConfig can also be used with earlier releases of Oracle E-Business Suite.

The Applications Context

Where the *System Identifier* (SID) was traditionally used to identify a file as belonging to a particular Oracle E-Business Suite environment, an *Applications context* is used in an AutoConfig-managed environment such as Oracle E-Business Suite Release 12.2. 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 Oracle E-Business Suite system

- Allows easier startup and shutdown of applications services

- Permits services to be installed or deinstalled independently of others

- Integrates seamlessly with a shared application tier file system (see Chapter 9)

- Enables use of Oracle Applications Manager for configuration management (see Chapter 7)

- Facilitates support for Oracle Real Application Clusters (see Chapter 4)

Before AutoConfig and the Applications context 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 Oracle E-Business Suite 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 Oracle E-Business Suite patches.

There are separate context files for the application and database tiers of an Oracle E-Business Suite system.

Applications Context File - The Applications context file, $INST_TOP/appl/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 Oracle E-Business Suite configuration files and update relevant database profiles.

Information stored includes:

- Name and location of the database

- Port numbers for Forms and Web services

- 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 server, the relevant variable will be set to execute the scripts that start the service.

The following fragment illustrates the structure, format, and content of the Applications context file:

```
<!-- Environment on this Node --<
   <oa_environments<
     <adconfig<
        <adconfig_file oa_var="s_adconfig_file"
>/u01/R122_EBS/EBSapps/appl/admin/adconfig.txt</adconfig_file>
        <APPL_TOP oa_var="s_at_adconfig>"
>/u01/R122_EBS/EBSapps/appl</APPL_TOP>
        <APPL_TOP_CSET oa_var="s_at_cset">AL32UTF8</APPL_TOP_CSET>
        <APPS_ENV_NAME oa_var="s_appsEnvName">VISION</APPS_ENV_NAME>
        <TIER_ADADMIN oa_var="s_isAdAdmin">YES</TIER_ADADMIN>
        <TIER_ADWEB oa_var="s_isAdWeb">YES<TIER_ADWEB>
        <TIER_ADFORMS oa_var="s_isAdForms">YES</TIER_ADFORMS>
        <TIER_ADNODE oa_var="s_isAdConc">YES</TIER_ADNODE>
        <APPL_TOP_NAME oa_var="s_atName">testsys100</APPL_TOP_NAME>
        <TIER_ADFORMSDEV oa_var="s_isAdFormsDev">YES</TIER_ADFORMSDEV>
        <TIER_ADNODEDEV oa_var="s_isAdConcDev">YES</TIER_ADNODEDEV>
        <TIER_ADWEBDEV oa_var="s_isAdWebDev">YES</TIER_ADWEBDEV>
        <STAGING_DIRECTORY oa_var="s_staging_directory"
</u01/R122_EBS/inst/apps/vision_test100/appltmp/updates</STAGING_DIRECTO
RY>
     </adconfig<
     <oa_environment type="generic_service">
        <DISPLAY oa_var="s_display" osd="unix">LOCALHOST:5.0</DISPLAY>
        <java_awt_headless oa_var="s_java_awt_headless"
>TRUE</java_awt_headless>
     </oa_environment>
     <oa_environment type="rapid_install"<
        <APPS_BASE oa_var="s_base"</u01/R122_EBS</APPS_BASE>
        <PATCH_BASE oa_var="s_patch_base"</u02/R122_EBS</PATCH_BASE>
        <INST_BASE oa_var="s_inst_base"</u01/R122_EBS/inst</INST_BASE>
        <HTML_TOP oa_var="s_html"</u01/R122_EBS/FMW_Home/Oracle_EBS-
app1/applications/oacore/html</HTML_TOP>
        <JDK_TOP oa_var="s_jdktop" osd="LINUX_X86-64"
</u01/R122_EBS/FMW_Home/jrockit_160_22_D1.1.1-3</JDK_TOP>
        <JRE_TOP oa_var="s_jretop"
</u01/R122_EBS/FMW_Home/jrockit_160_22_D1.1.1-3/jre</JRE_TOP>
```

> **Note:** The s_base and s_inst_base context variables were
> introduced in Release 12.2 to support Oracle WebLogic Server, and, like
> other context variables, should not be edited manually.

Database Context File - The database context file, <s_base>/11.2.0
/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 on UNIX (command files on Windows) include:

- **adautocfg.sh** - Wrapper script that passes the name of the specific environment

context file to adconfig.sh. Located in $INST_TOP/admin/scripts on the application tier, and <s_base>/11.2.0/appsutil/scripts/<CONTEXT_NAME> on the database tier.

- **adconfig.sh** - Invoked by `adautocfg.sh`, this script is a wrapper for adconfig.pl. Located in $AD_TOP/bin on the application tier, and <s_base>/11.2.0/appsutil/bin on the database tier.

- **adconfig.pl** - Invoked by `adconfig.sh`, this Perl script calls the Java API to carry out the actual configuration tasks. Located in $AD_TOP/bin on the application tier, and <s_base>/11.2.0/appsutil/bin on the database tier.

In addition to these configuration scripts, several additional AutoConfig scripts are used to help manage the Oracle E-Business Suite 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.

AutoConfig Directories

Directory Name	Directory Contents
$INST_TOP/admin/install	Install scripts
$INST_TOP/admin/scripts	Control scripts
$INST_TOP/admin/log	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 can 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.

AutoConfig does not undertake all aspects of configuration management, such as changes at operating system level that may have implications outside the context of Oracle E-Business Suite. In addition, it does not manage technology components that have their own native management tools: typically, these are components that are not specific to Oracle E-Business Suite, such as Oracle WebLogic Server.

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

Application Tier CVM Files

File Location	Description
$AD_TOP/bin/adcvm.sh	Main CVM script
$AD_TOP/admin/template/adcvmat.xml	Stores CVM-related data for the application tier

Database Tier CVM Files

File Location	Description
<s_base>/11.2.0/appsutil/bin/adcvm.sh	Main CVM script
<s_base>/11.2.0/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 $<PRODUCT>_TOP/admin/template directories on the application tier, and in the <s_base>/11.2.0/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 Oracle E-Business Suite database.

Templates for Management Scripts - To run all the standard processes required by Oracle E-Business Suite, 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 $<PRODUCT>_TOP/admin/driver directory on the application tier, and in the <s_base>/11.2.0/appsutil/template directory on the database tier.

Configuration Files

AutoConfig configuration files, with a *.conf* extension, are created when AutoConfig instantiates the corresponding template files. The values in the configuration files correspond to site-specific settings. AutoConfig creates numerous configuration files in various directories.

Relationship Between AutoConfig Files
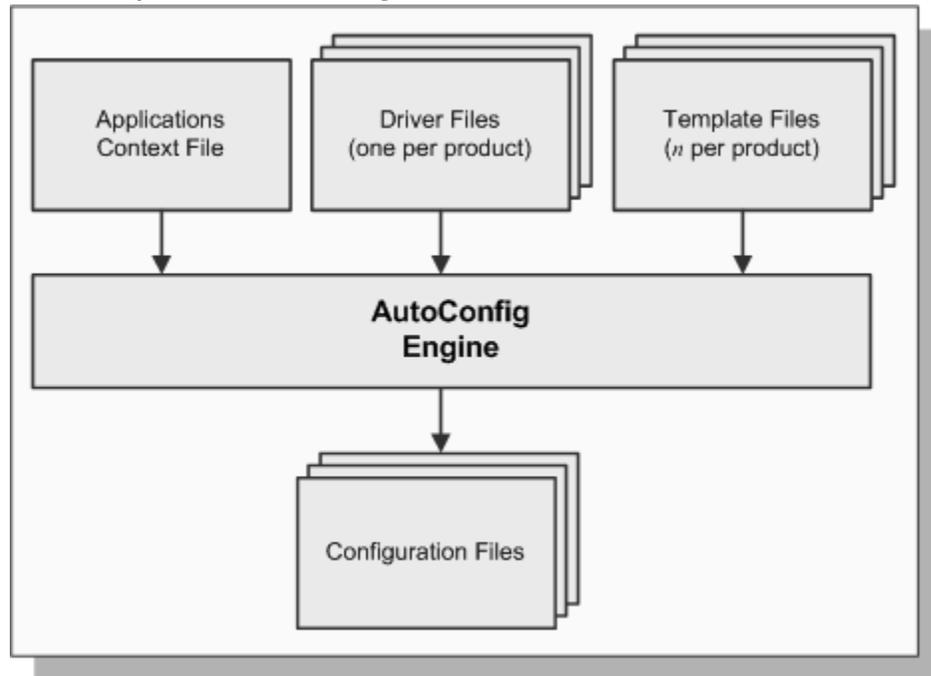
Instantiation

As mentioned, 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 *template* files to determine the basic settings needed. There is one template file for each configuration file. 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 $<PRODUCT>_TOP/admin/driver directories looking for driver files such as adtmpl. drv, fndtmpl.drv, and icxtmpl.drv.

> **Additional Information:** For further details of AutoConfig operation, refer to Chapter 3, Technical Configuration, of *Oracle E-Business Suite Setup Guide*.

Execution of Scripts

As well as its instantiation activities, AutoConfig may execute other scripts, depending on the requirements of the specific Oracle E-Business Suite system.

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 Oracle E-Business Suite installation.

Management Tasks

There are several areas in which an administrator can use AutoConfig to update, control, and monitor an Oracle E-Business Suite 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 Release 12.2 will also be updated by applying an Oracle E-Business Suite patch that adds or modifies variables in the Applications context file, AutoConfig template files, or AutoConfig driver files.

On occasion, you may need to undo configuration changes that have been made. The previous configuration can be restored 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.

The backup files are located in $INST_TOP/admin/out/<MMDDhhmm> on the application tier, and <s_base>/11.2.0 //appsutil/out/<CONTEXT_NAME><MMDDhhmm> on the database tier, where the

<MMDDhhmm> 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.

Build Context Utility

The `adblxml` utility, which was introduced in Release 12.1, can be used to create a new context file on the database tier. The new file acts as a central repository for database configuration information, and is essential in enabling AutoConfig on the database tier of an Oracle E-Business Suite instance.

A new context file on the database tier can be created using the following commands:

```
$ cd <s_base>/11.2.0/appsutil/bin
$ perl adbldxml.pl \
[template=<contextfile_template>] \
[out=<contextfile_name>
```

Where <contextfile_template> is the location of the context template (default is <s_base>/11.2.0/appsutil/template/adxdbctx.tmp), and <contextfile_name> is the absolute path of the context file to be generated (default is <s_base>/11.2.0 /appsutil/<context_name>.xml).

> **Note:** The `adbldxml` utility is only supported on the database tier, not on the application tier.

Controlling the System

AutoConfig utilizes a number of application tier control scripts, located in $INST_TOP/admin/scripts.

Key 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 <s_base>/11.2.0 /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 facilitate system management in an AutoConfig-managed environment such as Release 12. For example, you can identify the effects of proposed changes before you make them.

Configuration Check Utility

`adchkcfg.sh` is located in <AD_TOP>/bin on the application tier, and in <s_base>/11. 2.0/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. In Release 12.1, the tool was enhanced to report important database updates, generate a report for database changes, and improve readability of reports.

> **Additional Information:** For further details on managing system configuration parameters, see *Oracle E-Business Suite Maintenance Guide*.

AutoConfig Profiler

Introduced in Release 12.1, the profiling feature provides a consolidated HTML report of an AutoConfig run. The top level of the report displays a summarized view that lists all the product tops, along with the total instantiation and execution time of the templates within each product top. You can drill down into the report to view additional details, including the source and target location of each template, the time taken to instantiate and execute individual template scripts, and the execution report for each template script.

AutoConfig can be run in profile mode on the application tier using the following command:

```
$AD_TOP/bin/adconfig.pl contextfile=<CtxFile> \
[product=<product-top>] -profile
```

Where <CtxFile> is the absolute path to the application tier context file, and <product_top> is the short name of the product to configure.

AutoConfig can be run in profile mode on the database tier using the following command:

```
perl <s_base>/11.2.0/appsutil/bin/adconfig.pl \
contextfile=<CtxFile> -profile
```

Where <CtxFile> is the absolute path to the database tier context file.

Enhanced Service Dependency Management

In addition to the above utilities, AutoConfig's service management infrastructure has been enhanced with dependency information between service groups and services. This

dependency information is stored as metadata in the context file.

The following XML attributes were added in Release 12.1:

- **services**: contains a comma-separated list of the services that belong to a particular service group.

- **incompatible_with** (optional): contains a comma-separated list of service groups and services that are mutually exclusive with the service group or service for which they are defined. For example, Forms Socket mode and Forms Servlet mode are mutually incompatible.

- **requires** (optional): contains a comma-separated list of the required service groups and services.

Running AutoConfig in Parallel

Introduced in Release 12.1, the *parallel mode* feature enables AutoConfig to be run simultaneously across multiple nodes of an Oracle E-Business Suite system. When running in parallel mode, AutoConfig uses the `dbms_locks` PL/SQL package to ensure that configuration of one node does not interfere with configuration of other nodes: this is required because certain AutoConfig configurations of one node depend on the configurations of other nodes.

> **Additional Information:** During the instantiation phase, locking is done at script level. During the execution phase, locking is done at product_top level.

AutoConfig can be run in parallel mode on the application tier using the following command:

```
perl $AD_TOP/bin/adconfig.pl contextfile=<CtxFile> \
[product=<product_top>] -parallel
```

Where <CtxFile> is the absolute path of the application tier context file and <product_top> is the short name of the product to be configured.

AutoConfig can be run in parallel mode on the database tier using the following command:

```
perl <s_base>/11.2.0/appsutil/bin/adconfig.pl \
contextfile=<CtxFile> -parallel
```

Where <CtxFile> is the full path to the database tier context file.

> **Note:** When using parallel mode, AutoConfig must be invoked with the `-parallel` option on *all* nodes of the relevant tier.

6

Cloning

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 Oracle E-Business Suite 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 Oracle E-Business Suite 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

This chapter examines the principles and strategies involved in cloning an Oracle E-Business Suite Release 12.2 system using the *Rapid Clone* tool.

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.

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

Cloning Options

The following list summarizes the cloning options currently available with Rapid Clone.

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

- Single node to Single node

- Recloning (of database only)[1]

- Clone existing Clone[2]

- Multi-node to Multi-node

- Single node to Multi-node[3]

- Multi-node to Single node [4]

Footnotes on List

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 Oracle E-Business Suite patches have been applied to the source system, the rest of the components (APPL_TOP, COMMON_TOP and ORACLE_HOMEs) 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 procedure is often referred to as merging APPL_TOPs.

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 <s_base>/11.2.0/appsutil/template on the database tier.

> **Important:** Rapid Clone requires all the expected binary files to be in place on the source system, and may fail to operate correctly if any are missing.

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.

Rapid Clone and Oracle Universal Installer

The Oracle Universal Installer's *global inventory* is simply a list of pointers to each *local inventory* location. The inventory is located in <s_base>/11.2.0/inventory, which contains all the patch information for the ORACLE_HOME.

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.

Additional Cloning Features

Several features are 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 on the target system.

> **Note:** This list of actions is not exhaustive.

Database tier:

- Creates the Database context file

- Registers the ORACLE_HOME in the Global Inventory

- Relinks the ORACLE_HOME

- Configures the ORACLE_HOME

- Recreates the database control files

- Starts the database

- Configures the database

- Starts the database listener

Application tier:

- Creates the Applications context file

- Registers the OracleAS 10.1.2 and FMW ORACLE_HOMEs in the Global Inventory

- Relinks the OracleAS and FMW ORACLE_HOMEs

- Configures the OracleAS and FMW ORACLE_HOMEs

- Configures the APPL_TOP

- Creates the INST_TOP

- Starts application tier server processes

For a complete description of cloning options and associated actions, refer to My Oracle Support Knowledge Document 1383621.1, *Cloning Oracle E-Business Suite Release 12.2 with Rapid Clone*.

7

Security

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.

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 Oracle 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. This section describes a number of factors that should be taken into account when designing and implementing a security policy.

> **Additional Information:** For further details of recommended security practices and strategies, see My Oracle Support Knowledge Document 403537.1, *Best Practices For Securing Oracle E-Business Suite Release 12*.

Basic Strategies

There are several basic steps that can be taken to enhance the security of an Oracle E-Business Suite database, as outlined below.

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.

DBAs should generally use named accounts at the OS level, and the use of privileged operating system accounts such as root, oracle, applmgr should be tightly regulated. Where named accounts are not feasible, limited access accounts should be used.

Common tasks that require elevated privileges should wherever possible be performed via delegation mechanisms such as Oracle Enterprise Manager or sudo. Where this is not feasible, auditing should be used to track performance of privileged tasks.

For example, Oracle E-Business Suite requires the APPS password and access to the applmgr operating system account to start and stop application tier services such as the Concurrent Processing server. An Oracle Enterprise Manager script or sudo script can be used to start and stop the Concurrent Processing server without requiring the user to know the APPS password or have direct access to the applmgr account.

> **Note:** For further details, see My Oracle Support Knowledge Document 950018.1, *Using Database Vault with Oracle E-Business Suite*.

Patching

Applying patches to Oracle E-Business Suite requires the person performing the patching (normally, the DBA) to provide the passwords for the APPS account and (if on AD-TXK Delta 13 or later) the EBS_SYSTEM account or (if on an earlier AD-TXK release update pack) the SYSTEM account. All 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 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 should not be relied on for complete protection of data: the DBA could still carry out unauthorized actions, immediately before or after the patching operation. Therefore, additional safeguards such as auditing and keystroke logging may be advisable during patching operations.

Auditing DBA Activity

Oracle Database includes features that can be used to track 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 extensive 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.

The Human Aspect

While technical measures such as those already mentioned can and should be used to restrict DBA actions and selectively audit DBA activity, it is important to remember 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. A company's data is just as sensitive and valuable as 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.

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.

If regulatory compliance or other reasons require DBA access to be restricted, optional features such as Oracle Database Vault and Transparent Data Encryption (both described elsewhere in this chapter) may be employed as part of a comprehensive approach to security that includes policies and practices to meet auditing, segregation of duties, and other business requirements.

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.

> **Additional Information:** For further details of Audit Trail, see *Oracle E-Business Suite Security Guide*.

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 Oracle 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.

Oracle User Management

Oracle User Management (UMX) 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.

With Oracle User Management, instead of exclusively relying on a centralized administrator to manage all its users, an organization can, if desired, 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.

A login assistance mechanism is easily accessed from the Oracle E-Business Suite Login Page. A user simply clicks on the "Login Assistance" link located below the Login and Cancel buttons, and can then go to a Forgot Password section or Forgot User Name section to have the necessary action taken automatically, without the need for an administrator to become involved.

Another feature allows users with the relevant privileges to enable other users to act on their behalf, as delegates, without having to share the account password. For example, managers may need to grant peers or subordinates limited authority to act on their behalf while they are out of the office. This *Proxy User* feature allows control over the pages, functions, and data security policies that can be granted, and includes an on-screen display that indicates when a user is acting on behalf of another user.

Role Based Access Control

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 E-Business Suite

- 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.

Database Security Features

The Oracle database has always included mechanisms to protect its contents from unauthorized access, without hindering access by legitimate users. Details of these can be found in the standard database documentation.

Optional database features can be used to enhance and customize security to meet the specific needs of a site:

- *Transparent Data Encryption*, - a feature that can be used either to encrypt all content stored in a particular tablespace, or only the data stored in specific database columns.

- *Oracle Database Vault* - a feature that can be used to help address security issues such as insider threats, regulatory compliance requirements, and enforced separation of duties.

These two features will be discussed next.

Oracle Database Vault

Oracle E-Business Suite Release 12.0.4 and higher support *Oracle Database Vault*, a feature that can be used to help address security issues such as insider threats, regulatory compliance requirements, and enforced separation of duties. A number of flexible options enable you to apply fine-grained access control to sensitive data. In particular, Oracle Database Vault can be employed to protect sensitive data from unauthorized access by privileged users, while still allowing them to maintain your Oracle databases.

Oracle Database Vault uses realms (groupings of Oracle objects) to provide an internal firewall that prevents unauthorized access to specific application data. These realms can be supplemented with command rules - for example, classes of commands (such as INSERT, UPDATE, DELETE, DROP) can be prohibited. The command rules can, in the

form of rule sets, help ensure that DBAs do not perform DML within the realm.

Effective use of Oracle Database Vault requires an initial evaluation of the privileges required for the various roles involved in administering the database and applications. These roles should be chosen to minimize the requirement to use generic database accounts (APPS, APPLSYS, EBS_SYSTEM, SYSTEM, and the product schema accounts). Use of Oracle Database Vault should be combined with organization-specific processes to provide a more complete security solution. For example, DBAs should use specific named accounts wherever possible, which assists with the provison of realms and simplifies the ability to audit activities.

Some DBAs will require little or no access to applications data.These include roles that focus on administration of third-party software, or database administration and tuning. The default realms shipped with Oracle E-Business Suite prevent DBAs from viewing any applications objects.

Oracle E-Business Suite DBAs who need access to a defined subset of applications data (for example, the FND tables) can be given access to a realm that provides access to this subset of tables. The are two strategies for accomplishing this. The most controlled way to do this is by defining specific objects in the Oracle E-Business Suite application to which DBAs are allowed access so they can perform particular roles. The named DBA accounts should be granted access to a realm that provides access to that data specifically.

An alternative approach, which is easier to maintain if the DBAs need access to a large number of applications tables, involves modification of the shipped realms to create a list of tables that contain sensitive information to which DBAs are *not* permitted to have access. However, this strategy is less rigorous as by default it allows rather than denies access to objects.

Oracle E-Business Suite delivers pre-defined realms for the integration with Oracle Database Vault. With AD-TXK Delta 13 and later, integration with Oracle Database Vault is simplified with two predefined realms. With earlier AD-TXK release update packs, five predefined realms were recommended.

> **Additional Information:** For further details, see the following My Oracle Support knowledge documents as applicable:
>
> - Doc ID 2981476.1, *Integrating Oracle E-Business Suite Release 12.2 with Oracle Database Vault 23ai Using EBS_SYSTEM*
>
> - Doc ID 2727580.1, *Integrating Oracle E-Business Suite Release 12.2 with Oracle Database Vault 19c Using EBS_SYSTEM*
>
> - Doc ID 2617770.1, *Integrating Oracle E-Business Suite Release 12.2 with Oracle Database Vault 19c*
>
> - Doc ID 2131435.1, *Integrating Oracle E-Business Suite Release 12.2*

Transparent Data Encryption

For fields containing sensitive information (such as credit card numbers), it may be desirable to encrypt the data stored.

Currently, Oracle E-Business Suite does not automatically encrypt data. Optionally, *Transparent Data Encryption* (TDE) functionality can be used to encrypt selected areas of the database. As the encryption is transparent to the application, code does not have to be rewritten to provide the encryption, and existing SQL statements should work without modification.

Two encryption strategies are available:

- **TDE Tablespace Encryption** is used to encrypt all content stored in a particular tablespace.

- **TDE Column Encryption** is used to encrypt only specific database columns, typically those that contain sensitive data.

    Note: For further details, see My Oracle Support Knowledge Documents 828229.1, *Using TDE Tablespace Encryption with Oracle E-Business Suite Release 12*, and 732764.1, *Using TDE Column Encryption with Oracle E-Business Suite Release 12*.

8

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.

> **Additional Information:** For additional authentication and authorization documentation, refer to My Oracle Support Knowledge Document 1934915.1, *Oracle E-Business Suite Release 12.2 Technology Stack Documentation Roadmap*. In particular, see Document 2278051.1, *Integrating Oracle E-Business Suite Release 12.2 with Oracle Unified Directory 12c* and Document 2003483.1, *Integrating Oracle E-Business Suite Release 12.2 with Oracle Unified Directory 11g Release 2*.

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 Discoverer.

> **Additional Information:** If applicable, refer to My Oracle Support Knowledge Document 2277369.1, *Oracle E-Business Suite Support Implications for Discoverer 11gR1* .

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.

> **Note:** *Oracle Directory Services* refers to both *Oracle Internet Directory* and

Oracle Unified Directory. Procedures documented for implementing
Oracle Directory Services apply to both these directories.

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 Directory Services to multiple application accounts
in Oracle E-Business Suite Release 12.2.

> **Important:** Use of the advanced authentication features described in
> this chapter, such as Single Sign-On, are optional with Oracle E-
> Business Suite Release 12.2. If you wish to use them, you must carry out
> the requisite additional setup procedures as noted later.

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

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
two classes of component, each of which performs a distinct but related function. The
first class is an LDAP directory such as *Oracle Unified Directory (OUD)*. The second class
is a single sign-on product such as *Oracle Access Manager*, which is used in conjunction
with *Oracle E-Business Suite AccessGate*, a Java Enterprise Edition application that maps
a single sign-on user to an Oracle E-Business Suite user, and creates the Oracle E-
Business Suite session for that user.

The following diagram illustrates the high-level structure of a typical integration.
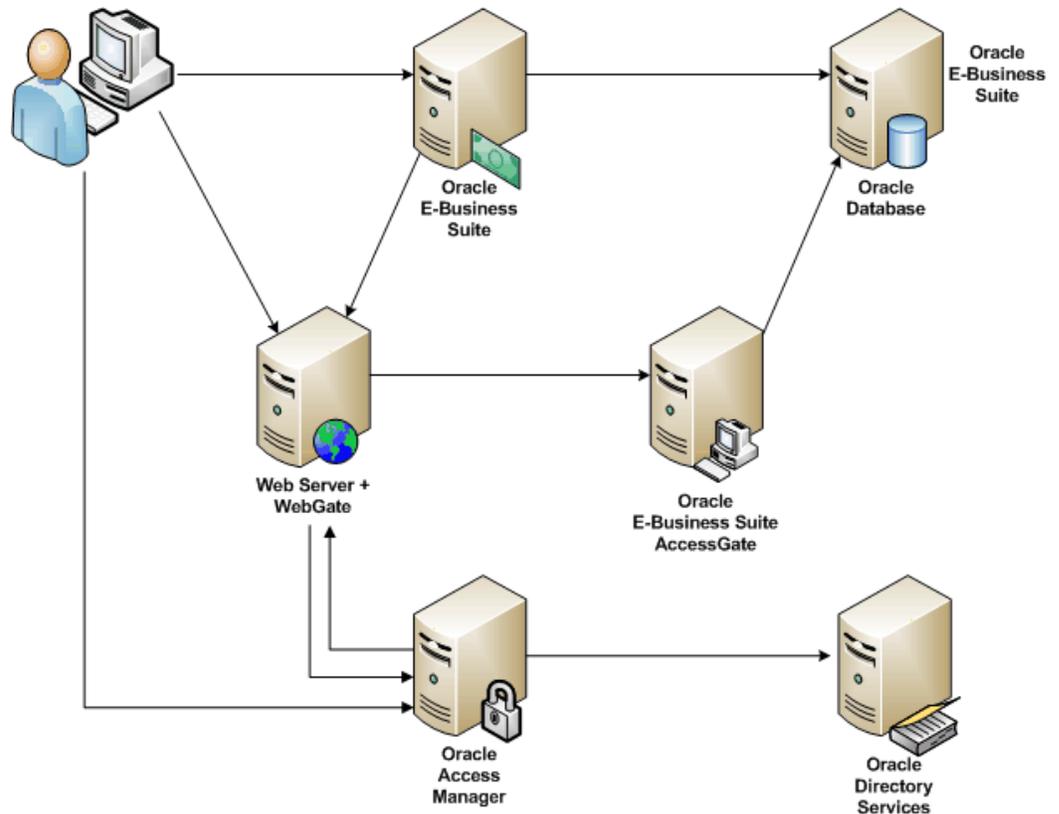
Implementing a single sign-on solution involves significant changes to the mechanism by which Oracle E-Business Suite Release 12.2 users are authenticated. Instead of authentication being performed natively, via the FND_USER table, this functionality is delegated to Oracle Single Sign-On, which can either:

- Perform user validation itself, against information stored in Oracle Directory Services.

- Delegate validation to a third-party single sign-on server.

With either of these solutions, Oracle E-Business Suite Release 12.2 accepts identities vouched for by the single sign-on mechanism. Oracle Directory Services complements this by acting as an integration point that enables Oracle E-Business Suite Release 12.2 to participate in enterprise level user management.

The sequence of actions is illustrated in the following diagram.

Steps shown in diagram:

7. User redirected to original URL with authenticated EBS session and resource is returned to browser

1. User requests protected resource

Oracle E-Business Suite

Oracle Database

2. User redirected to OAM-protected AccessGate

5. OAM WebGate securely passes user identifier to EBS AccessGate

6. AccessGate identifies the EBS user that is linked to the authenticated Oracle Directory Services user

Web Server + WebGate

Oracle E-Business Suite AccessGate

3. Per OAM policies, WebGate intercepts request, collects credentials, and relays them to Access Manager server

4. OAM verifies credentials against user directory and creates user session

Oracle Access Manager

Oracle Directory Services

> **Additional Information:** Note that where a third-party single sign-on server is in use, Oracle Access Manager and Oracle Directory Services are still required, to provide a bridge between Oracle E-Business Suite Release 12.2 and the third-party single sign-on solution.

Each Oracle 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 Directory Services 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 Directory Services and Oracle E-Business Suite.

The key role of Oracle Directory Services is to link the separate namespaces used by Oracle Access Manager and Oracle E-Business Suite. Linking the namespaces ensures that a particular user logging in via Oracle Access Manager is the same user that is represented within Oracle E-Business Suite's FND_USER repository. The linking is done by associating externally-managed Oracle Access Manager users with internally-managed Oracle E-Business Suite users via the GUIDs, which are generated by Oracle Directory Services.

In addition to this role, Oracle Directory Services has another significant part to play. While Oracle E-Business Suite is typically used mainly within an organization, certain application modules such as iRecruitment need to be available to outside users without accounts being created manually and responsibilities being assigned. This means application modules that support self-registration must create user accounts *synchronously* (in Oracle E-Business Suite and the external directory at the same time) and on demand. Oracle E-Business Suite uses specific Oracle Directory Services function calls to handle these synchronous account creation tasks.

Another requirement in such a single sign-on 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 Directory Services and a third-party LDAP server.

- Support for automatic propagation of user information across an enterprise, via a *provisioning* process between Oracle Directory Services and Oracle E-Business Suite Release 12.2.

User information in external, third-party user directories can be synchronized with Oracle Directory Services using the LDAP protocol. With Oracle Directory Services, customers can manage and publish user information in a central location that various application systems, including 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 Directory Services and Oracle E-Business Suite during the transition to an integrated single sign-on environment.

> **Additional Information:** For more information on implementing single sign-on with Oracle Single Sign-On and Oracle Directory Services, see Chapter 6 of *Oracle E-Business Suite Security Guide*, and My Oracle Support Knowledge Document 1388152.1, *Overview of Single Sign-On Integration Options for Oracle E-Business Suite.*.

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 Access Manager and Oracle Directory Services 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.

> **Additional Information:** 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 12.2 installation, plus a new Oracle Access Manager and Oracle Directory Services installation on a different machine.

Oracle Directory Services has no currently existing users, apart from pre-seeded users. The requirement is to integrate Oracle E-Business Suite Release 12.2 with Oracle Access Manager and Oracle Directory Services.

Key Goals

- Oracle E-Business Suite Release 12.2 will delegate user sign-on and authentication to Oracle Access Manager

- Oracle Access Manager and Oracle E-Business Suite AccessGate will authenticate user credentials against user entries in Oracle Directory Services

- Oracle Directory Services will store every user's single sign-on account id and password

Deploying E-Business Suite with Oracle Access Manager and Oracle Directory Services

User Management Options

Existing Oracle E-Business Suite Release 12.2 application accounts are migrated to single sign-on accounts in Oracle Directory Services 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 12.2, then provisioned into Oracle Directory Services.

- Oracle E-Business Suite Release 12.2 is configured as a *provisioning integrated application* with Oracle Directory Services

- System administrators configure the provisioning integration via *provisioning profiles*

Provisioning User Information from Oracle E-Business Suite to Oracle Directory Services

Oracle E-Business Suite	Directory Integration Platform	Oracle Directory Services
Userid = "New.User1"	Provisioning Profile	Userid = "New.User1"
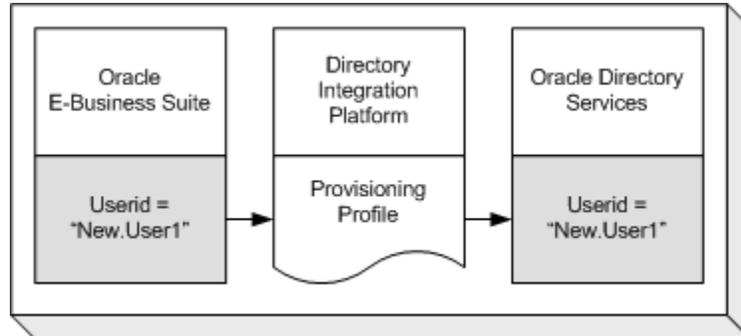
The creation of a new application account in Oracle E-Business Suite Release 12.2 will automatically trigger the creation of a new single sign-on account in Oracle Directory Services. Some of the user attributes from the application account may be provisioned in the single sign-on account in Oracle Directory Services during account creation.

Option 2

All user information is created in Oracle Directory Services, then provisioned into Oracle E-Business Suite Release 12.2:

- Oracle E-Business Suite Release 12.2 is configured as a provisioning integrated application with Oracle Directory Services

- System administrators configure the provisioning integration via *provisioning profiles*

Provisioning User Information from Oracle Directory Services to E-Business Suite
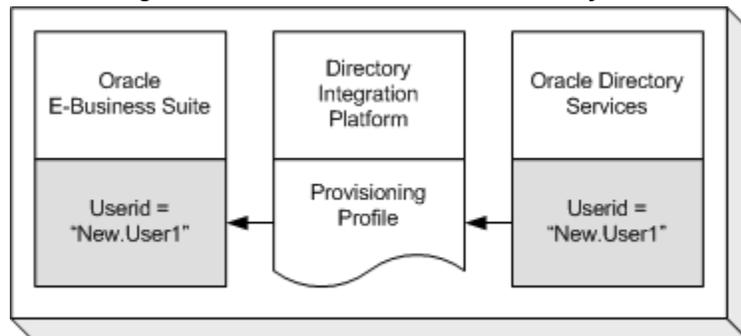
Oracle E-Business Suite	Directory Integration Platform	Oracle Directory Services
Userid = "New.User1"	Provisioning Profile	Userid = "New.User1"

The creation of a new single sign-on account in Oracle Directory Services will automatically trigger the creation of a new application account in Oracle E-Business Suite Release 12.2. Some of the user attributes from the single sign-on account may be

provisioned in the application account in Oracle Directory Services during account creation.

Option 3

All user information is created in either Oracle Directory Services or Oracle E-Business Suite Release 12.2, then provisioned into the other system:

- Oracle E-Business Suite Release 12.2 is configured as a provisioning integrated application with Oracle Directory Services

- System administrators configure the provisioning integration via provisioning profiles

Provisioning User Information Between Oracle E-Business Suite and Oracle Directory Services
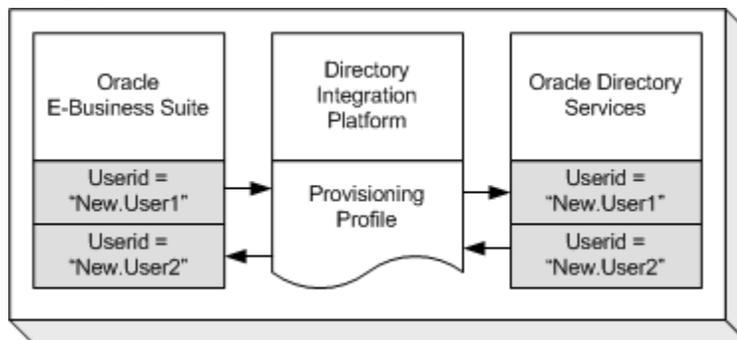
The creation of a new application account in Release 12.2 will automatically trigger the creation of a new single sign-on account in Oracle Directory Services, and the creation of a new single sign-on account in Oracle Directory Services will automatically trigger the creation of a new application account in Release 12.2.

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

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

Signing On

Attempting to gain access to an Oracle E-Business Suite Release 12.2 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 12.2.

Signing Out

When a user logs out of an Oracle E-Business Suite instance, the user is also logged out of Oracle Access Manager and E-Business Suite AccessGate, as well as any *partner applications* that have been integrated with Oracle Access Manager. The user will see a logout page that lists all the applications he 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 Access Manager, 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 Access Manager security cookie has expired. Since the application session timeout value takes precedence over the Oracle Access Manager timeout setting, Oracle recommends setting the application session timeout value to be equal to or less than that of Oracle Access Manager.

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 Access Manager with Oracle E-Business Suite Release 12.2

Starting Environment

- Multiple new Oracle E-Business Suite Release 12.2 environments have been installed

- Other than the default administrative accounts, no user accounts have been registered yet

- No single sign-on infrastructure in place

Solution

- Oracle Access Manager and Oracle Directory Services are needed for the integration required

- Oracle E-Business Suite Release 12.2 will delegate user sign-on and authentication to Oracle Access Manager

- Oracle Access Manager authenticates user credentials against user entries in Oracle Directory Services

- Oracle Directory Services contains every user's single sign-on account ID and password

Either Oracle Directory Services or one Oracle E-Business Suite Release 12.2 instance can be designated as the source of user enrollment, with the following implications:

- If Oracle Directory Services 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 Directory Services, and then to the other Oracle E-Business Suite instances.

Optionally, user profile information in an Oracle E-Business Suite Release 12.2 instance can be kept synchronized with the information in Oracle Directory Services.

Scenario 2

Requirement - Need to integrate new installation of Oracle E-Business Suite Release 12.2 with existing third-party single sign-on and user directory infrastructure

Starting Environment

- Oracle E-Business Suite Release 12.2 has been newly installed using Rapid Install.

- Other than the default administrative accounts, no user accounts have been registered yet.

- A third-party single sign-on solution is in use as a corporate single sign-on solution.

- A third-party LDAP directory is in use as a corporate user directory.

Solution

- Oracle Access Manager and Oracle Directory Services are needed for the integration.

- Oracle E-Business Suite and Oracle Access Manager must be set up so that Oracle E-Business Suite delegates authentication to Oracle Access Manager , which in turn delegates the functionality to the third-party single sign-on server in use.

- Oracle Directory Services 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 Directory Services also needs to be set up to provision users in Oracle Directory Services to Oracle E-Business Suite.

Integrating E-Business Suite with Third-Party Single Sign-On and User Directory
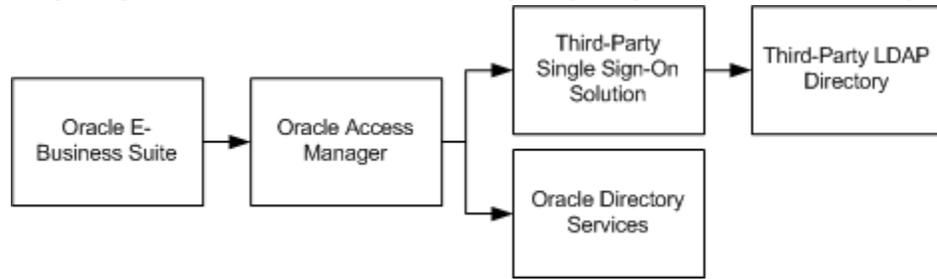
Existing users in the third-party LDAP directory can be bulk migrated into Oracle Directory Services, and then bulk migrated into Oracle E-Business Suite.

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 12.2 with existing third-party single sign-on and user directory infrastructure

Starting Environment

- Oracle E-Business Suite Release 12.2 is in use, and has an up to date user repository.

- A third-party corporate single sign-on solution is in use and is to be retained.

- A third-party LDAP directory 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 12.2 and the third-party LDAP directory, with either the same user name in both or a different user name in each.

Solution

- Oracle Access Manager and Oracle Directory Services are needed for the integration.

- Oracle E-Business Suite and Oracle Access Manager need to be set up so that Oracle E-Business Suite delegates authentication to Oracle Access Manager , which in turn delegates the functionality to the third-party single sign-on server.

- Oracle Directory Services 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 Directory Services.

- 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 Directory Services, 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 or LDAP directory is involved. There is only an existing Oracle E-Business Suite Release 12.2 installation, plus an Oracle Access Manager and Oracle Directory Services 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 12.2 installations where no Oracle Single Sign-On infrastructure is currently in place

Starting Environment

- Multiple Oracle E-Business Suite Release 12.2 instances are implemented, and each has an existing user population.

- No existing Oracle Access Manager infrastructure is in place.

Solution

- Oracle Access Manager and Oracle Directory Services are needed for the integration.

- Each Oracle E-Business Suite instance delegates user sign-on and authentication to Oracle Access Manager .

- Oracle Access Manager authenticates user credentials against user entries in Oracle Directory Services.

- Oracle Directory Services 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 Directory Services.

- 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 Directory Services.

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 Directory Services will correspond to a single application account in Oracle E-Business Suite Release 12.2. However, in special cases a user may need to have a single sign-on account in Oracle Directory Services and multiple application accounts in Oracle E-Business Suite Release 12.2.

Single Sign-On Account with Multiple Application Accounts
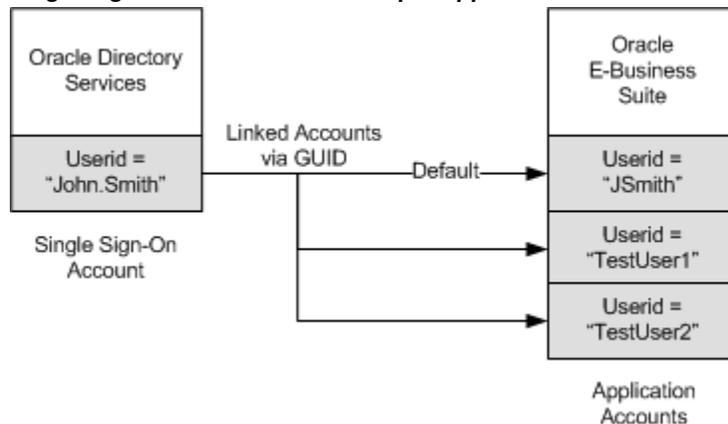
If required, this feature can be enabled by system administrators via the profile option 'Applications SSO Allow Multiple Accounts'.

9

High Availability

Introduction

The subject of High Availability covers a range of standard features and additional options that help to minimize planned and unplanned downtime, or facilitate recovery after a period of downtime. They include:

- Patching Hints and Tips

- Shared Application Tier File System

- Distributed AD

- Nologging Operations

- Disaster Recovery

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.

> **Note:** High availability is greatly facilitated by the *online patching* capability introduced in Oracle E-Business Suite Release 12.2. This key feature is described in Chapter 5.

High Availability Principles

In addition to taking full advantage of online patching, other strategies for high availability include:

- **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.

- **Apply the latest software updates** - 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.

- **Consolidate multiple patches** - Merging multiple Oracle E-Business Suite patches into a single patch minimizes the scope for error that would arise in applying a number of separate patches. This can now be undertaken by the adop utility, although the traditional AD Merge Patch utility is still available for use in specialized cases.

- **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 either Oracle Applications Manager or the Rapid Clone tool to create a copy of your production system for tests.

Where applicable, these strategies are described further below.

> **Note:** For full details of carrying out patching and maintenance operations, see the Patching and Managment Tools chapter of this book, and the relevant chapters of *Oracle E-Business Suite Maintenance Guide*.

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.

- A hardware or system failure or malfunction, such as a media failure, or operating system problem that corrupts the actual data in the database

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* and *Flashback Database*.

- Data Guard is used to 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 unavailable following a significant problem, or via a switchover operation that is executed to allow service to continue during planned maintenance of the environment's

platform or building services.

- Flashback Database is used to "rewind" a database to a prior point in time, making it possible to recover from major logical corruptions of a database without requiring a complete restore.

You must also install any other hardware and software required to run your standby environment as a production environment after a failover, ensuring that any changes on the primary are matched on the standby. Examples include tape backup equipment and software, system management and monitoring software, and other applications.

Oracle Data Guard and Oracle E-Business Suite Release 12.2

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 *Redo Apply* (often referred to as *Physical Standby*) and *SQL Apply* (often referred to as *Logical Standby*). . Both of these use the primary database's redo information to propagate changes to the standby database.

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

- Logical standby employs the Oracle LogMiner utility to build SQL statements that recreate changes made to the data. The logical standby mechanism is not currently utilized with Oracle E-Business Suite.

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 traffic. The other requirement is that the servers at the secondary site are the same type as at the primary site, in sufficient numbers to provide the required level of service; depending on your organization's needs, this could either be a minimal level of service (supporting fewer users), or exactly the same level of service as you normally provide.

Data Guard's reliance on redo 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 faster throughput. Oracle recommends turning on the *force logging* feature at the database level to simplify your backup and recovery, and standby database maintenance procedures. In cases where the nologging feature is used in Release 12.2, and you have chosen not to use force logging, insufficient redo information will be generated to make the corresponding changes on the standby database. You may then be required to take manual steps to refresh the standby (or recreate the relevant objects) to ensure it will remain usable.

Finally, based on your organization's business requirements, choose one of the following protection modes:

- **Maximum protection**: This protection mode ensures that no data loss will occur if the primary database fails. To provide this level of protection, the redo data needed to recover each transaction must be written to both the local online redo log and to the standby redo log on at least one standby database before the transaction commits. To ensure data loss cannot occur, the primary database shuts down if a fault prevents it from writing its redo stream to the standby redo log of at least one transactionally-consistent standby database.

- **Maximum availability**: This protection mode provides the highest level of data protection that is possible without compromising the *availability* of the primary database. Like maximum protection mode, a transaction will not commit until the redo needed to recover that transaction is written to the local online redo log, and to the standby redo log of at least one transactionally-consistent standby database. However, unlike maximum protection mode, the primary database does not shut down if a fault prevents it from writing its redo stream to a remote standby redo log. Instead, the primary database switches to maximum performance mode until the fault is corrected, and all gaps in redo log files are resolved. When all gaps have been resolved, the primary database automatically resumes operating in maximum availability mode. This strategy ensures that no data loss will occur if the primary database fails, unless a second fault prevents a complete set of redo data from being sent from the primary database to at least one standby database.

- **Maximum performance**: This protection mode (the default) provides the highest level of data protection that is possible without affecting the *performance* of the primary database. This is accomplished by allowing a transaction to commit as soon as the redo data needed to recover that transaction is written to the local online redo log. The primary database's redo data stream is also written to at least one standby database, but that redo stream is written asynchronously with respect to the transactions that create the redo data. When network links with sufficient bandwidth are employed, this mode provides a level of data protection that approaches that of maximum availability mode, with minimal impact on primary database performance.

Flashback Database

Oracle recommends you enable the Flashback Database feature, to:

- Help protect against logical data corruption

- Allow you to reinstantiate the production database as a standby after a failover to your secondary site

- Create database restore points to which you can flash back in case an upgrade or major application change encounters a serious problem

Flashback Database enables you to rewind the database to a previous point in time without restoring backup copies of the data files. This is accomplished during normal operation by Flashback Database buffering and writing before images of data blocks

into the *flashback logs*, which reside in the *flash recovery area*.

Flashback Database can also flashback a primary or standby database to a point in time prior to a Data Guard role transition. In addition, a Flashback Database operation can be performed to a point in time prior to a resetlogs operation, which allows administrators more flexibility to detect and correct human errors.

For further information, refer to My Oracle Support Document 2966999.1, *Using Active Data Guard Reporting with Oracle E-Business Suite Release 12.2 and Oracle Database 23ai*, Document 2608030.1, *Using Active Data Guard Reporting with Oracle E-Business Suite Release 12.2 and Oracle Database 19c*, or Document 1944539.1, *Using Active Data Guard Reporting with Oracle E-Business Suite Release 12.2 and an Oracle 11g or 12c Database*.

Additionally, see Document 2246690.1, *Business Continuity for Oracle E-Business Suite Release 12.2 Using Logical Host Names with an Oracle 12c Physical Standby Database*, Document 2088692.1, *Business Continuity for Oracle E-Business Suite Release 12.2 using Virtual Hosts with Oracle 12c Physical Standby Database*, or Document 1963472.1, *Business Continuity for Oracle E-Business Suite Release 12.2 Using Oracle 12c (12.1.0.2) Physical Standby Database*.

10

Load Balancing

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 Oracle E-Business Suite 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 very 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 the following two reasons:

- Load balancing can involve the entire infrastructure of an Oracle 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.

A key load balancing area strategy is to share the application tier file system across multiple nodes, allowing servers and services to be enabled or disabled on particular nodes and thus making optimum use of the available hardware. This is particularly useful in cases where new nodes are added or existing ones removed.

Other load balancing options apply to particular technologies or components of the Oracle E-Business Suite system. These include:

- Domain Name Server (DNS)

- Web (HTTP) Services

- Forms Services

- Concurrent Processing Layer

- Database Layer

This chapter describes these load balancing strategies, to facilitate informed evaluation of the applicability of a particular load balancing option in achieving the desired business requirements.

Shared Application Tier File System

> **Important:** The current status of support for a shared application tier file system in Oracle E-Business Suite Release 12.2 can be found in My Oracle Support Knowledge Document 1375769.1, *Sharing The Application Tier File System in Oracle E-Business Suite Release 12.2*.

Historically, a traditional multi-node installation of Oracle E-Business Suite required each application tier node to maintain its own file system. 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 was referred to as a *Shared APPL_TOP File System*, usually abbreviated to *Shared APPL_TOP*.

A subsequent refinement provided the capability 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 be shared between them all.

These concepts were subsequently extended 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 application tier file system in an Oracle E-Business Suite Release 12.2 environment. Current restrictions are also noted where applicable.

For practical considerations of using a shared file system in Release 12.2 online patching environments, refer to *Oracle E-Business Suite Maintenance Guide*.

Shared Application Tier File System Features

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.

Benefits of using a shared application tier file system include:

- Overall disk space requirements are greatly reduced, as there is only a single copy of the relevant Oracle E-Business Suite code.

- Most administrative tasks need only be carried out once, and take effect immediately on all nodes.

Current restrictions on using a shared application tier file system include:

- An application tier file system can only be shared across machines running either identical or binary compatible operating systems.

- Sharing file systems between internal and external application tiers is not supported. This is true even for external application tiers that have reverse proxies in the DMZ.

- Shared application tier file system functionality is not currently available on Windows.

Shared Disk Resources

A shared application tier file system can reside on any standard 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.

High Availability Features of Shared Application Tier File System

Utilizing a 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.

Load Balancing Definitions

Load balancing is the means by which network traffic directed to a particular web site is divided between one or more machines in a cluster of servers. Typically, the servers will all be running the same application, and use of load balancing presents the outside world with the appearance of a single server rather than a cluster. The relevant hardware device, often called a *server load balancer*, receives the relevant network traffic

for the site, and distributes it to the various servers in the cluster based on the load balancing methodology that it supports. By sending requests to different nodes within the server cluster, system performance is optimized, scalability is simplified, and application availability (a key requirement of web-based applications) is greatly enhanced.

Additional features of load balancing include *monitoring server availability* and *context-based load distribution*.

Monitoring server availability maintains a watch on the server pool, and periodically checking that all machines are responding to the traffic; if one is not (perhaps because of a network problem) it is taken out of the pool of servers, so that traffic can be routed to the servers that remain available.

Context-based load distribution is required for applications such as Oracle E-Business Suite that need to maintain *session persistent-connections*, whereby cookies are created when the session is initially established. To support this requirement, the load balancer reads and updates the network packet header information sent with each request made by the client, and routes the request to the node in the cluster with which the corresponding session was originally established and is subsequently being maintained.

Categories of Load Balancer

Load balancers are not installed as part of Oracle E-Business Suite. However, Oracle E-Business Suite can easily be configured for use with them when required.

The two main categories of load balancer are:

Session-Persistent Load Balancers - Using this type of load balancer ensures that after a client's HTTP connection is established with a 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 - This type of load balancer uses a round-robin strategy to balance incoming HTTP requests, and does not maintain session-persistent client connections. After the initial HTTP client-server connection, subsequent HTTP requests from a client will be not necessarily be directed to the same server.

Various options exist when setting up a load balancer for use with Oracle E-Business Suite. The most important ones are discussed in the next section.

Load Balancing Options

Different load balancing strategies can be employed with Oracle E-Business Suite. The choice will heavily depend on the specific needs and existing infrastructure of the site.

Domain Name Server (DNS) Round Robin Load Balancing

This load balancing strategy distributes end-user requests across multiple server nodes, based on dynamic assignments of IP addresses to a given hostname and associated domain name. Utilizing a feature known as *DNS round robin* allows the association of

more than one IP address with a given hostname, thereby facilitating distribution of the traffic across the list of IP addresses associated with that hostname.

DNS Layer Load Balancing
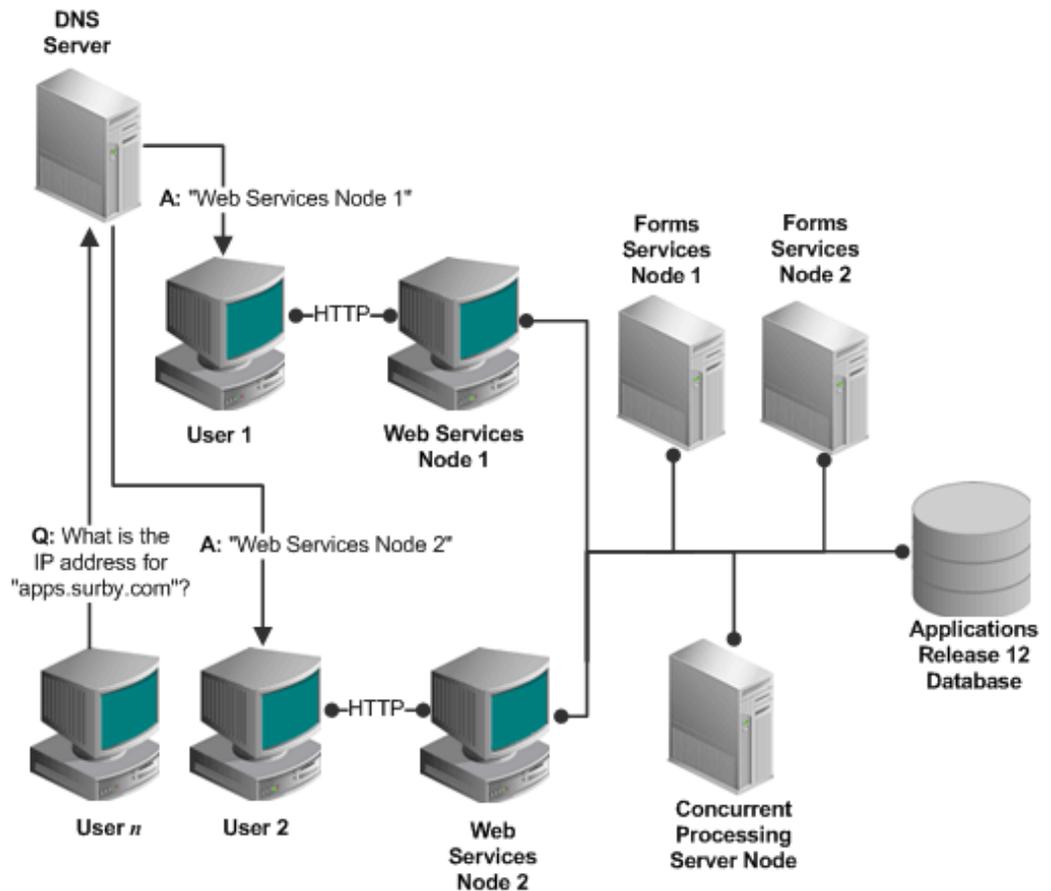
Although DNS round robin load balancing is a simple and effective way to distribute traffic among several servers, it has several limitations. These include unpredictable load distribution across servers (depending on user activities), and DNS caching issues with browsers.

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*.

HTTP Layer Load Balancing
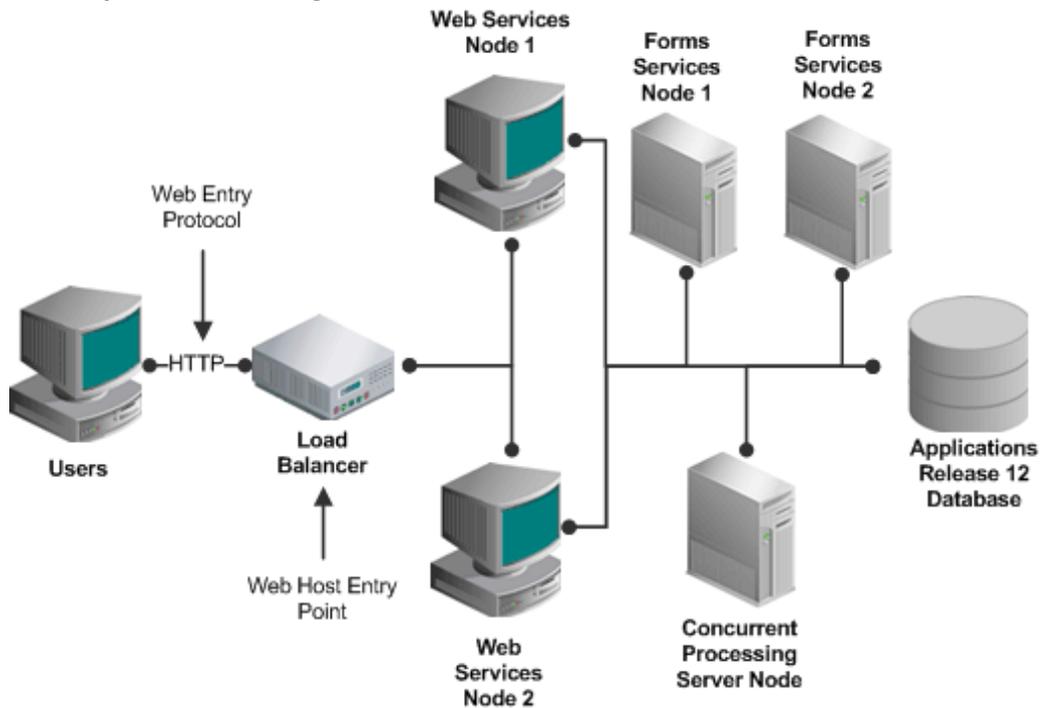
Hardware-based HTTP load balancers must be configured to use session-persistent connections for all traffic through all ports for Oracle E-Business Suite Release 12.2 environments. This is required as various Oracle E-Business Suite products and components need to maintain session state. If session-persistent connections are not configured, users may experience transaction state loss errors while using Oracle E-Business Suite.

> **Additional Information:** To learn more about load balancing options and setup procedures, see My Oracle Support Knowledge Document 1375686.1, *Using Load-Balancers with Oracle E-Business Suite Release 12.2*.

11

Globalization Support

Introduction

Oracle E-Business Suite Release 12.2 is designed for ease of deployment in a single global instance that meets the complex requirements of a worldwide enterprise. *Globalization* is the process of designing and deploying software that supports a global enterprise. Key globalization features provided by Release 12.2 include support for a wide variety of languages and territories, flexible date and number formats to suit local custom, reporting currencies, and other country-specific functionality to provide compliance with local statutory requirements.

The majority of Release 12.2 Oracle E-Business Suite products 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 means there is no inherent limitation on the number of supported languages that can be run in a single database.

> **Additional Information:** To learn more about about character sets, see *Oracle Database Globalization Support Guide*. For more detailed information about globalization, see My Oracle Support Knowledge Document 393861.1, *Globalization Guide for Oracle Applications Release 12*.

National Language Support (NLS) refers to the ability to run an Oracle E-Business Suite instance in a supported language, including specific regional number and date formats.

Multiple Language Support (MLS) refers to the ability to support multiple languages in the same Oracle E-Business Suite instance. Most products in Oracle E-Business Suite Release 12.2 are MLS-enabled. Using a Unicode character set with the MLS architecture allows Oracle E-Business Suite to use any combination of supported languages from one database instance. Because it covers the widest range of characters, a Unicode character set such as AL32UTF8 is recommended as the character set for installations that support, or plan to support, multiple languages. Oracle supports over thirty languages in Release 12, which are installed via NLS patches.

> **Additional Information:** For a full list of supported languages, see
> *Oracle E-Business Suite Installation Guide: Using Rapid Install*.

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 a Unicode character set such as AL32UTF8. 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 E-Business Suite, 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, for example, you need to support English, French, Japanese, and Arabic, you must choose a Unicode character set (such as AL32UTF8) to be able to support all these languages.

The extended multilingual support present in the Oracle E-Business Suite Release 12 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 *Oracle E-Business Suite Installation Guide: Using Rapid Install*.

Using a multibyte character set such as Unicode AL32UTF8 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 My Oracle Support Knowledge Document 393861.1, *Globalization Guide for Oracle Applications Release 12*.

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 AL32UTF8 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 AL32UTF8 is a superset of all other supported character sets, there are no other fully compatible character sets. If you use AL32UTF8 on the database tier, you must use AL32UTF8 on all other 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 Client 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 E-Business Suite 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.

User Preferences

In Release 12.2, 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 E-Business Suite using the Oracle 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.

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. Application tier processes must be started with the character set of the server, as determined when Rapid Install was run.

Date and Number Formats

You can enter and view dates in any valid format, such as 10-20-11, 20/10/11, or 2011-10-20. The only exception is with Oracle Reports, which always uses the format DD-MON-RRRR, for example 20-OCT-2011.

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

> **Note:** For further details of date and number formats, see My Oracle Support Knowledge Document 393861.1, *Globalization Guide for Oracle Applications Release 12*.

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.

Global Application Design

This section describes the key criteria involved in designing and implementing applications that will be used in multiple countries, perhaps all over the world.

Multiple Time Zone Support

Release 12 of Oracle E-Business Suite includes as standard a feature called *User-Preferred Time Zone Support*. In most existing E-Business Suite implementations, all users interact with the system in the "corporate" time zone, which will normally be the time zone of the headquarters of the implementing company, and the time zone in which the database runs. This means that remote users have to be aware of the time difference between their location and that of the corporate headquarters.

Employing the user-preferred time zone feature enables users to specify their local time zone for both display and entry of date-with-time fields. Key consequences of this are:

- Users see date-with-time fields in their preferred (local) time zone, and can enter dates with time in this time zone

- Date fields without a time component are not affected by this feature

- The data in the database continues to be stored in the standard corporate time zone

Time Zone Concepts

Conceptually, there are two types of date fields:

- Dates *with* a time component – used to indicate a specific point in time within a particular day

- Dates *without* a time component – used to denote a particular day, but not a specific point in time within that day

Date fields with a time component can be represented in any time zone, and thus displayed in whichever time zone is most meaningful to the end user. Generally, users prefer to view dates in their own (local) time zone. With the user-preferred time zone feature enabled, date with time fields will be converted to the user's preferred time zone for display.

Date fields without a time component cannot be represented in different time zones, because no meaningful conversion is possible for a date that does not include a specific time. Such a date is entered with respect to one time zone, and in general must be viewed as a day in that time zone, regardless of the location (and possibly different time zone) in which it is being viewed. Oracle E-Business Suite typically uses the corporate time zone for these day definitions: dates without a time component represent the day with respect to the corporate headquarters (corporate days).

There are some exceptions to the above rule. For example, dates without a time component may be held as *ANSI dates,* to represent dates independently of the time zone in which they are being viewed. In such a case, a benefit that starts on 1st January will start on that date wherever in the world it is made available; that is, it will apply to anyone who is in a time zone where it is 1st January.

Many dates without a time component represent pointers to a financial period. These dates are not meant to indicate the exact hour and minute that a transaction occurred, but rather the financial period into which the transaction is accounted. This is a financial bucketing from the perspective of the implementing company. For example, the invoice dates on Payables or Receivables invoices never change based on who is looking at them: they are classified as invoices for that day (and thus that period), regardless of the viewer's local time zone.

> **Additional Information:** For details of setting up Oracle E-Business Suite to use multiple time zones, see *Oracle E-Business Suite Setup Guide*. Also see My Oracle Support Knowledge Documents 402650.1, *User Preferred Time Zone Support in Oracle E-Business Suite Release 12*, and 563019.1, *Complying with Daylight Saving Time (DST) and Time Zone Rule Changes in E-Business Suite 12*.

Reporting Currencies

The *Reporting Currencies* feature allows you to report on and maintain accounting records in more than one functional currency. This can be done at the subledger journal level, General Ledger journal level, or balance level. You define one or more reporting currencies, and assigning them to a *primary ledger* or *secondary ledger* using General

Ledger's Accounting Setup Manager.

The following terms are fundamental to the principles described in this section.

Ledgers

Primary Ledger	A financial reporting entity in which you conduct business. The primary ledger acts as the main, record-keeping ledger and uses a particular chart of accounts, accounting calendar, currency, and subledger accounting method.
Secondary Ledger	An optional, additional ledger that is associated with the primary ledger for an accounting setup. Secondary ledgers can be used to represent the primary ledger's accounting data in another accounting representation that differs in one or more of the following from the primary ledger:

- Chart of accounts

- Accounting calendar and period type combination

- Currency

- Subledger accounting method

- Ledger processing options such as Suspense Posting

Currencies

Ledger Currency	The currency used to define a ledger, which is also the base currency used to record your business transactions and accounting data within the General Ledger module of Oracle E-Business Suite.
	The primary ledger currency is generally the currency in which you perform most of your business transactions, and the one you use for legal reporting.

Reporting Currency	The currency used for an organization's financial statements. Shares the same chart of accounts and accounting calendar as the source ledger (either the primary ledger or secondary ledger), but need not be the same currency as that of the primary or secondary ledger.
Functional Currency	The main currency used by a organization. The functional currency can be different from the ledger currency that is assigned to the primary and secondary ledgers.

For example, you may choose Japanese Yen (JPY) for your ledger currency, even though the functional currency for the accounting purposes of your integrated business group is actually US Dollars (USD). The determination of the functional currency is based on a number of factors, discussed in SFAS #52 and IAS 21. |

You must define a separate reporting currency for each additional currency representation you need. You assign one or more reporting currencies to your primary ledger or secondary ledger using General Ledger's Accounting Setup Manager.

Unlike secondary ledgers, reporting currencies must share the same chart of accounts, accounting calendar/period type combination, subledger accounting method, and ledger processing options as their source ledger.

As a general rule, always use reporting currencies instead of secondary ledgers if you only need to maintain an accounting representation that differs in currency alone. You can assign reporting currencies to both primary and secondary ledgers. Reporting currencies are maintained at one of the following currency conversion levels:

- **Subledger**: Maintains a currency representation of your subledger journals, General Ledger journal entries, and balances. When using the subledger level reporting currency, you must define subledger accounting rules using Subledger Accounting. These rules provide instructions on how to convert subledger data entered into the source ledger to one or more subledger level reporting currencies. You must also define journal conversion rules that General Ledger Posting uses to automatically convert specific journals, such as manual journal entries, to one or more subledger level reporting currencies.

  > **Note:** Subledger level reporting currencies can only be assigned to primary ledgers, not secondary ledgers.

- **Journal**: Maintains General Ledger journal entries and balances in another currency representation. Journal level reporting currencies are maintained using the General Ledger Posting Program. Every time a journal is posted in the source ledger, such as the primary ledger or secondary ledger, the journal is automatically converted to the respective journal level reporting currency based on the journal conversion rules defined.

- **Balance**: Maintains balances in another currency. The General Ledger Translation program is used to convert balances from the source ledger to the balance level reporting currency. When you run Translation in your primary or secondary ledger and specify a target currency, the translated balances are reflected in the balance level reporting currency.

  > **Note:** If a balance level reporting currency is not assigned to the ledgers in the accounting setup, a balance level reporting currency is automatically created the first time Translation is run. The name of the balance-level reporting currency is the same as its source ledger, except that its currency code, such as GBP, is appended to its name.

The subledger level and journal level reporting currencies act similarly to ledgers. You must open and close the periods for these reporting currencies before you can enter transaction and journal entries. You can also enable journal approval for these reporting currencies if planning to enter manual journal entries directly to these reporting currencies.

> **Note:** For further details about reporting currencies, see *Journal or Subledger Level Reporting Currencies in Oracle General Ledger User's Guide*, and *Reporting Currencies* in*Oracle Financials Implementation Guide.*.
>
> For further details about secondary ledgers, see *Secondary Ledgers* in *Oracle Financials Implementation Guide*.

Country-Specific Functionalities

One requirement for successful globalization is to meet the statutory, legal, and cultural practices of a given locality. In Oracle E-Business Suite Release 12.2, this is achieved is 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.

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 12.2 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.

> **Additional Information:** See *Oracle E-Business Suite Setup Guide* for a list of external documents provided in Release 12.2.

12

Multiple Organization Architecture

Introduction

The *Oracle E-Business Suite Multiple Organizations Architecture* (hereafter referred to as *Multiple Organizations*), provides support for multiple organizations in a single installation of Oracle E-Business Suite, with relationships you define. Multiple Organizations specifies how the different organizations interact, and how transactions flow between them. These organizations can be ledgers, business groups, legal entities, operating units, or inventory organizations.

Overview

When you run any Oracle E-Business Suite product in a Multiple Organizations environment, you first choose an organization, either implicitly (by choosing a responsibility), or explicitly (by selecting an operating unit when entering data in transaction entry pages, or by running reports). Inventory organizations can be selected in the Choose Organization window; each subsequent window and report displays information for the chosen organization only.

The following terms are fundamental to the Multiple Organizations architecture:

Ledger (Set of Books in previous releases)	A financial reporting entity that uses a particular chart of accounts, ledger currency, accounting calendar, and subledger accounting method. Oracle General Ledger secures transaction information (such as journal entries and balances) by ledger. A single Oracle General Ledger responsibility can access, process, and report on data for one or more ledgers that is assigned to its Data Access Set.
Operating Unit	An organization used by Oracle subledgers, such as Oracle Cash Management, Oracle Order Management and Shipping Execution, Oracle Payables, Oracle Purchasing, or Oracle Receivables. In this context, the organization may be a sales office, a division, a business unit, or a department. An operating unit is associated with a legal entity and primary ledger.
	Information is secured by operating unit. A single application responsibility can access, process, and report on data for one or more operating units that is assigned to its security profile.

> **Note:** For further details, see: *Oracle E-Business Suite Multiple Organizations Implementation Guide*, *Oracle General Ledger Implementation Guide*, and *Oracle Financials Implementation Guide*.

Multiple Organizations Partitioned Objects

Tables that contain Multiple Organizations data can be identified by the suffix "_ALL" in the table name. These tables include a column called ORG_ID, which partitions Multiple Organizations data by organization..

Every Multiple Organizations table has a corresponding view that partitions the table's data by operating unit. Multiple Organizations 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 is a similar way to the LANGUAGE variable, which returns the language of the current session.

> **Note:** If accessing data from a Multiple Organizations 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 Multiple Organizations partitioned object.

Converting to Multiple Organizations

When you install a production Oracle E-Business Suite system, the data model is identical whether you implement Multiple Organizations or not. The Multiple Organizations views used to partition data are incorporated into the normal install, and use predefined text. When you convert to Multiple Organizations, tables are renamed to use the "_ALL" suffix to denote that they now contain Multiple Organizations partitioned data.

In Release 12.2, the AD Administration utility is used for the initial conversion of seed data and transaction data to Multiple Organizations 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 E-Business Suite Maintenance Guide*.

Glossary

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.

See also: tier

Applications Database Administrator

The Applications Database Administrator (Applications DBA) is the person responsible for preparing the application tier for installation of Oracle E-Business Suite, and subsequently maintaining the application tier and patching application code when needed. The Applications DBA has access to highly privileged Oracle E-Business Suite database accounts including EBS_SYSTEM, APPS, APPLSYS, APPS_NE, and all product schemas.

applmgr

The account used to install and upgrade Oracle E-Business Suite. This account owns the Oracle E-Business Suite product files.

APPS schema

An ORACLE schema that has access to the complete Oracle E-Business Suite 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 E-Business Suite Release 12 environment, AutoConfig includes a number of scripts and other files that simplify the process of making updates to a system. A key file is the *Applications context file*.

Automatic Undo Management (AUM)

A database feature, 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 mulitlingual support.

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. The client-server architecture is not used by Oracle E-Business Suite Release 12.2.

clone

A copy of an Oracle E-Business Suite system, typically used for testing purposes. A clone can be created by using the Rapid Clone tool, or from Oracle Applications Manager.

codelevel

A rollup patch with a name that looks like a version number. Higher codelevels include more changes, and include all fixes from lower codelevels.

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 E-Business Suite 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 E-Business Suite server that runs time-consuming, non-interactive 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 to process 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. Oracle E-Business Suite Release 12 uses cost-based optimization exclusively.

See also: hint

customization

Enhancements to an Oracle E-Business Suite 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 Database Administrator (DBA) is the person who prepares the Oracle database server, performs installation of the Oracle Database, and subsequently carries out management and maintenance tasks as needed. The DBA has highly privileged access to the database via the SYSTEM and SYS accounts. With AD Delta 13 and later, the roles of the DBA and Applications DBA can be separated. Following the introduction of the EBS_SYSTEM schema with AD Delta 13 and later, the Applications DBA no longer requires access to the SYS and SYSTEM accounts.

database connection tagging

A mechanism by which Oracle E-Business Suite utilizes Oracle Database session attributes to record the current use to which a database connection is being put.

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 Oracle E-Business Suite system. The database server stores the data maintained by Oracle E-Business Suite, and processes SQL and Java requests from Concurrent Processing servers, Forms services, and Web services.

[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 E-Business Suite products. It is used primarily for system testing and user training.

See also: product group

dependent product

An Oracle E-Business Suite product that is not licensed, but whose files are shared in part by a fully installed Oracle E-Business Suite 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

distributed concurrent processing

See: parallel concurrent processing

distributed directory structure

Oracle E-Business Suite product files installed in more than one file system, as when there is insufficient disk space in a single file system for all Oracle E-Business Suite 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

EBS_SYSTEM Schema

Designed to provide improved separation of duties, the EBS_SYSTEM schema was

introduced with AD Delta 13 as the Applications DBA administration account for the Oracle E-Business Suite database. It is analogous to the SYSTEM schema in the Oracle Database.

Edition-Based Redefinition

The Oracle database feature that is used to implement online patching.

See also: online patching

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

An operating system parameter that help specify the behavior of Oracle E-Business Suite and other programs installed on your system. Environment settings are stored as environment variables on UNIX and as registry keys or environment variables on Windows.

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

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. Data can be entered by typing information into the relevant fields.

Forms client

A Java applet that runs on a desktop client and provides the user interface and interaction with Forms.

Forms server

See: forms services

Forms services

An application component that hosts the Oracle Forms engine. Traditionally referred to as *Forms server*, a term that does not accurately reflect the architecture used in Release 12.2. Forms services mediate 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. Release 12.2 uses the *Forms Listener Servlet* by default.

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

HTTP server

A server that sends out Web page content in response to HTTP requests from remote browsers. Often referred to as a *Web server*.

See also: Web browser

HTTPS

A version of HTTP that provides additional features needed for secure data transmission.

See also: HTTP

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

parameters affect many aspects of database activity, including performance, and should in general only have their values modified 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 E-Business Suite 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.

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

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

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.

JavaServer Page

See: JSP

JDBC (Java Database Connectivity)

A Java programming interface that enables Java programs to access the Oracle database server.

JRE

A plug-in component, JRE enables use of the Oracle JVM on a desktop client, instead of the browser's own JVM. The JRE Plug-in is installed when first required, the browser prompting the user to download the installation executable.

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 Directory Services, Oracle Internet Directory, Oracle Unified Directory

ledger currency

The currency used to record transactions and maintain Oracle E-Business Suite accounting data using primary and secondary ledgers.

See also: reporting currency

load balancing

The mechanism for distributing tasks to the least-busy server of the servers that are handling a particular workload.

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.

Maintenance Mode

A mode of operation in which the Oracle E-Business Suite system is accessible only for patching activities. Maintenance Mode status is controlled from within Oracle Applications Manager.

See also: Restricted Mode

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 down to 1,000,000 bytes for estimates of space needed.

Multi-Org

See: Multiple Organization Architecture

Multiple Organization Architecture

A single installation of any Oracle E-Business Suite product that can support any number of organizations and different ledgers. The data contained in product schemas is for all organizations, and is partitioned by the ORG_ID column in tables.

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

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.

online patching

A means of patching an Oracle E-Business Suite system with no downtime except for a restart ("bounce") of the servers and services.

See also: Edition-Based Redefinition

operating system

The computer software that performs basic tasks such as scheduling CPU time, allocating memory, and supervising communications between different computers.

Oracle Access Manager

The preferred Oracle single sign-on product for use with Oracle E-Business Suite Release 12.2. Replaces Oracle Single Sign-On. Used in conjunction with Oracle E-Business Suite AccessGate.

See also: Oracle E-Business Suite AccessGate.

Oracle Applications Manager

A sophisticated tool that enables system administrators to monitor and manage an Oracle E-Business Suite system from an HTML-based central control console.

Oracle Applications Tablespace Model (OATM)

A storage model used as standard in Oracle E-Business Suite Release 12, OATM uses locally managed tablespaces to provide more flexible storage options then the dictionary-managed tablespaces used in the traditional tablespace model.

See also: locally managed tablespaces

Oracle BI Publisher

An Oracle E-Business Suite product that uses standard technologies and tools to facilitate the rapid and easy development of reports in the end-user's preferred format. Originally known as Oracle XML Publisher.

See also: XML

Oracle Database

The database management system used by Oracle E-Business Suite. An Oracle database is made up of various different types of file.

See also: instance

Oracle Database Vault

An optional Oracle database feature that can be used to help address security issues such as insider threats, regulatory compliance requirements, and enforced separation of duties.

Oracle Directory Services

Generic term for a user directory service such as Oracle Internet Directory or Oracle Unified Directory.

See also: Oracle Internet Directory, Oracle Unified Directory

Oracle E-Business Suite AccessGate

A Java Enterprise Edition application that maps a single sign-on user to an Oracle E-Business Suite user, and creates the Oracle E-Business Suite session for that user. Used in conjunction with Oracle Access Manager.

See also: Oracle Access Manager

Oracle HTTP Server

Provides the foundation for the Oracle Application Server Web services. 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 Internet Directory

A user directory service for an Oracle E-Business Suite single sign-on solution such as Oracle Access Manager . Superseded by Oracle Unified Directory.

See also: Oracle Directory Services, Oracle Unified Directory

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. Sometimes referred to by its original name of SQL*Net.

Oracle RAC

See: Oracle Real Application Clusters

Oracle Real Application Clusters

Oracle Real Application Clusters (Oracle 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. Oracle RAC also offers resilience, allowing processing to continue in the event of one or more machines being unavailable because of planned or unplanned downtime.

ORACLE schema

See: schema

Oracle Unified Directory

A user directory service for an Oracle E-Business Suite single sign-on solution such as Oracle Access Manager. Supersedes Oracle Internet Directory.

See also: Oracle Directory Services, Oracle Internet Directory

ORACLE user ID

A user name employed to access an ORACLE instance. Used in conjunction with a *password*.

Oracle WebLogic Server

An enterprise-ready Java EE application server that supports the deployment of mission-critical applications in a robust, secure, highly available, and scalable environment. Utilized by Oracle E-Business Suite Release 12.2.

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_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

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 E-Business Suite 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

PL/SQL

A procedural extension of the SQL programming language that provides programming constructs such as blocks, conditionals, and functions.

platform

The type of computer system (hardware and operating system), on which application programs run. Some Oracle E-Business Suite functionality is platform-specific, meaning its availability and behavior may vary between different platforms.

portal

A software component that offers controlled access to an information source, possibly summarizing the content.

See also: portlet

portlet

A reusable component that forms the fundamental building blocks of a page in a portal such as Oracle Webcenter.

See also: portal

primary ledger

A financial reporting entity in which business is conducted. The primary ledger acts as the main record-keeping ledger, and uses a specific chart of accounts, accounting calendar, currency, and subledger accounting method.

See also: reporting currency

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 family release update pack

An aggregation of patches created for a specific product family after the initial release, and designated by the release number followed by the product short name and a letter.

See also: release update pack

product group

A set of Oracle E-Business Suite 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.

Rapid Install

The installer for Oracle E-Business Suite. Normally, a new version is released to accompany a maintenance pack. Using Rapid Install provides the same Oracle E-Business Suite code as could be obtained by applying the associated maintenance pack to an earlier release level. It also provides the latest certified technology stack. In contrast, applying a maintenance pack provides Oracle E-Business Suite code only.

See also: maintenance pack

registry

On a Windows machine, a central repository that holds configuration information. During installation, Oracle E-Business Suite 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

release update pack

A cumulative aggregation of product family release update packs created across Oracle E-Business Suite since the initial (base) release.

See also: product family release update pack

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 E-Business Suite to view concurrent processing files online.

Reporting Currencies

An Oracle E-Business Suite feature that allows you to create, maintain, and report on accounting data in multiple currencies at one of the following levels: Subledger, Journal, and Balance.

reporting currency

A currency, other than your ledger currency, for which you need to report. The reporting currency shares the same chart of accounts and accounting calendar as the source ledger (either the primary ledger or secondary ledger), but typically uses a different currency. The reporting currency allows you to report in a different currency than that of your primary or secondary ledger.

See also: ledger currency

reserved word

A word that has a special meaning to a computer program, for example END. Custom-built programs that integrate with Oracle E-Business Suite 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 Oracle E-Business Suite administrators to carry out privileged tasks such as viewing the progress of a patching session.

See also: Maintenance Mode

rollback segment

Historically, an 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 and later database server releases.

See also: undo tablespace

rollback tablespace

Historically, a tablespace created for rollback segments. Superseded by *undo tablespace* in Oracle9i and later database server releases.

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.

secondary ledger

An optional, additional ledger that is associated with the primary ledger. A secondary ledger can be used to represent the primary ledger's data in another accounting representation, which differs from the primary in one or more of the following: chart of accounts, accounting calendar/period type combination, currency, and subledger accounting method.

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 that provides a particular functionality. For example, the HTTP server responds to HTTP requests. In Release 12, *server* is, where applicable, being replaced by *services*, to reflect the architectural trend away from implementation via a single process.

[2} A computer dedicated to a particular role. For example, a database server.

service

[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 E-Business Suite 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 E-Business Suite 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 E-Business Suite 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.

SQL*Net

The original name for the Oracle Net software.

See also: Oracle Net

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 rules by which commands, qualifiers, and parameters are organized to form valid instructions to an operating system or application program.

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 E-Business Suite, 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 E-Business Suite product's database tables, and another for its indexes. Release 12 uses 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.

Transparent Data Encryption

An optional Oracle database feature that can be used to encrypt selected columns of database files.

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, Oracle Net

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 E-Business Suite system. Every username is associated with a password. In Oracle E-Business Suite, 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 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 BI Publisher

Index

See target process
Workflow Engine
   definition, 1-29
worklist
   Workflow, 1-29
work shift
   Concurrent Processing, 1-17
World Wide Web Consortium, 1-31

X

XSL-FO, 1-31