Oracle® Applications

Concepts

Release 11i (11.5.9)

Part No. B10642-01

May 2003

.

ORACLE®

Oracle Applications Concepts, Release 11*i* (11.5.9)

Part No. B10642-01

Contributors: Michael Bernstein, Eric Bing, Anne Carlson, Subash Chadalavada, Steven Chan, Ivo Dujmovic, Carole Eubanks, Michael Fiore, Ric Ginsberg, Cliff Godwin, Billy Greene, Jeff Lunn, K.R. Narayanan, Kent Noble, Emily Nordhagen, Lisa Parekh, Andrew Rist, Joan Ryan, Greg Seiden, Yun Shaw, Leslie Studdard, Keith M. Swartz, Millie Wang

Contents

5 Data Model

Glossary

Index

Send Us Your Comments

Oracle Applications, Concepts, Release 11*i* (11.5.9)

Part No. B10642-01

We welcome your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

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

If you find any errors or have any suggestions for improvement, please indicate the document title and part number and the chapter, section, and page number, if available. Send comments to us by e-mail at appsdoc_us@oracle.com. If you would like a reply, please give your name, address, and telephone number.

Preface

How To Use This Guide

Oracle Applications Concepts provides basic information on how the Oracle Applications software and databases are configured, and how you can implement them in a network installation.

This book contains the following chapters:

- Chapter 1 provides an explanation of Oracle Applications architecture.
- Chapter 2 describes enhancements in Release 11*i*.
- Chapter 3 describes aspects of internationalization support.
- Chapter 4 describes the file system that the Rapid Install wizard creates during installation.
- Chapter 5 describes the Oracle Applications data model.
- The Glossary provides definitions to terms you may encounter when installing.

Conventions

The following conventions are used in this book:

Convention	Meaning
UNIX: **Windows:**	Indicates platform-specific information. This guide contains information for both UNIX and Windows platforms.
$ or C:\>	Represents the platform-specific command prompt. Your prompt may differ.
`Monospace text`	Represents command line text. Type this text exactly as shown.
< >	Text enclosed in angle brackets represents a variable. Substitute a value for the variable text. Do not type the brackets.
[ ]	Encloses optional items or indicate a function key. Do not type the brackets.
|	Represents an *or* option among several options. You must enter only one of the options. Do not type the vertical bar.
\	In examples of commands you type online, a backslash at the end of a line signifies that you must type the entire command on one line. *Do not type the backslash.*

Documentation Accessibility

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

For additional information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Other Information Sources

You can choose from many sources of information, including online documentation, training, and support services, to increase your knowledge and understanding of Oracle Applications.

If this guide refers you to other Oracle Applications documentation, use only the Release 11*i* versions of those guides.

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF).

- **Online Help** - The new features section in the HTML help describes new features in 11i. This information is updated for each new release of Oracle Applications. The new features section also includes information about any features that were not yet available when this guide was printed. For example, if your administrator has installed software from a mini-packs an upgrade, this document describes the new features. Online help patches are available on Oracle*MetaLink*.
- **11i Features Matrix** - This document lists new features available by patch and identifies any associated new documentation. The new features matrix document is available on Oracle*MetaLink*.
- **Readme File** - Refer to the readme file for patches that you have installed to learn about new documentation or documentation patches that you can download.

Related User's Guides

You can read the guides online by choosing Library from the expandable menu on your HTML help window, by reading from the Oracle Applications Document Library CD included in your media pack, or by using a Web browser with a URL that your system administrator provides.

If you require printed guides, you can purchase them from the Oracle Store at http://oraclestore.oracle.com.

Guides Related to All Products

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI) available with this release of Oracle Applications. This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

You can access this user's guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Installation and System Administration

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11*i*, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications, the Oracle8 technology stack, and the Oracle8*i* Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user's guides and implementation guides.

Upgrading Oracle Applications

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11*i*. This guide describes the upgrade process and lists database and product-specific upgrade tasks. You must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to Release 11*i*. You cannot upgrade to Release 11*i* directly from releases prior to 10.7.

Maintaining Oracle Applications

Use this guide to help you run the various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to run the AD utilities. This guide also provides information on maintaining the Oracle applications file system and database.

Oracle Applications System Administrator's Guide

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

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Oracle Applications Developer's Guide

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

Oracle Applications User Interface Standards for Forms-Based Products

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

Other Implementation Documentation

Oracle Applications Product Update Notes

Use this guide as a reference for upgrading an installation of Oracle Applications. It provides a history of the changes to individual Oracle Applications products between Release 11.0 and Release 11*i*. It includes new features, enhancements, and changes made to database objects, profile options, and seed data for this interval.

Multiple Reporting Currencies in Oracle Applications

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

Multiple Organizations in Oracle Applications

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

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes, as well as how to monitor the progress of runtime workflow processes.

Oracle Workflow Developer's Guide

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

Oracle Workflow User's Guide

This guide describes how Oracle Applications 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 Applications Flexfields Guide

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

Oracle eTechnical Reference Manuals

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

Oracle Manufacturing APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes API's and open interfaces found in Oracle Manufacturing.

Oracle Order Management Suite APIs and Open Interfaces Manual

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes API's and open interfaces found in Oracle Order Management Suite.

Oracle Applications Message Reference Manual

This manual describes all Oracle Applications messages. This manual is available in HTML format on the documentation CD-ROM for Release 11*i*.

Training and Support

Training

Oracle offers a complete set of training courses to help you and your staff reach full productivity quickly. These courses are organized into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From onsite support to central support, our team of experienced professionals provides the help and information you need to keep Oracle Applications working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle9*i* server, and your hardware and software environment.

1

Oracle Applications Architecture

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

- Overview
- The Desktop Tier
- The Application Tier
- The Database Tier
- Features of the Oracle Applications Technology Layer

Overview

Oracle Applications architecture is a framework for multitiered, distributed computing that supports Oracle Applications products. In this model, various services are distributed among multiple levels, or tiers.

A *Service* is a process or group of processes running on a single machine that provides a particular functionality. The HTTP service, for example, is a process that listens for and processes HTTP requests, and the Forms service is a process that listens for and processes requests for Oracle Forms.

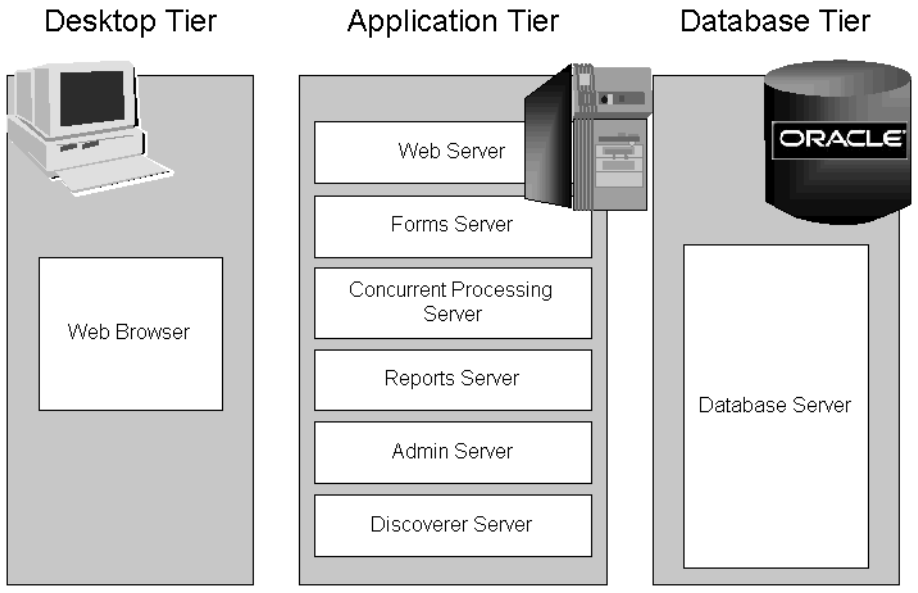
A *tier* is a grouping of services, potentially across physical machines. The tiers that compose Oracle Applications are the database tier, which manages the Oracle database; the application tier, which manages Oracle Applications and other tools; and the desktop tier, which provides the user interface display. Only the presentation layer of Oracle Applications is on the desktop tier in the form of a plug-in to a standard Internet browser.

Note that the tiers are logical groupings, and do not represent physical nodes or machines. Each tier can consist of one or more nodes and each node can

accommodate more than one tier. For instance, the database server can reside on the same node as one or more servers in the application tier.

Because Oracle Applications software is centralized, the need to install and maintain application software on each desktop client is eliminated. The software on the application tier also enables Oracle Applications to scale with load and to keep network traffic low.

Figure 1–1 Oracle Applications Architecture



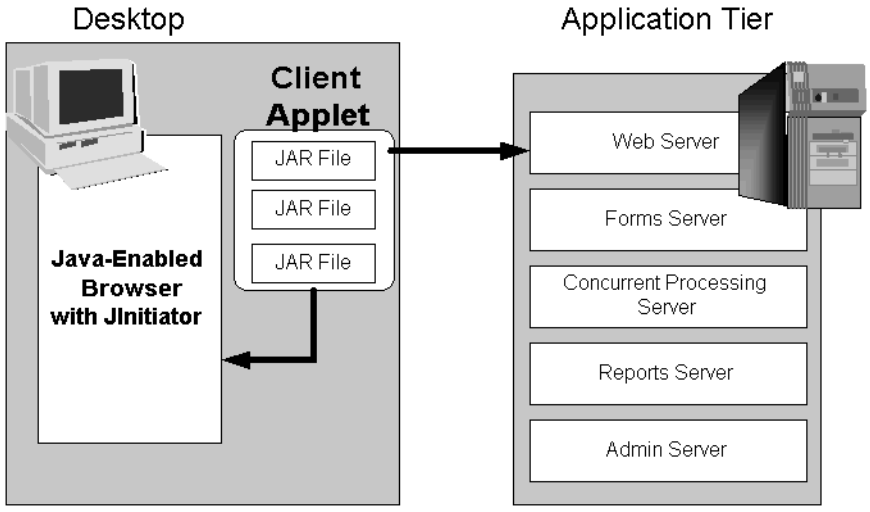
The connection between the application tier and the desktop tier can operate over a WAN. The desktop and application tiers exchange a minimum amount of information, such as field value comparison differences. In a global operation with users at diverse locations, less network traffic means less telecommunications expense and better performance.

The Desktop Tier

The client interface is provided through HTML for the Self-Service interface, and a Java applet for the professional, forms-based interface using a Java-enabled Web

browser. The desktop client with Oracle JInitiator downloads the applet on demand and the applet is cached locally for future use.

Figure 1–2 Desktop TIer Architecture



Forms Client Applet

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

The Forms client applet and commonly used JAR files are downloaded from the Web server at the beginning of the client's first session. Other less commonly used JAR files are downloaded as needed. The JAR files remain in the disk cache on the desktop, ready for future sessions, minimizing network traffic.

All updates are installed on the application tier and downloaded to the client automatically through the use of the JInitiator-enabled Web browser.

Oracle JInitiator

The Forms client applet must run within a Java Virtual Machine (JVM) on the desktop. For Oracle Applications, the JVM is supplied by Oracle JInitiator. Oracle JInitiator works in conjunction with the Web browser. It is implemented on the desktop client as a plug-in (Netscape Communicator) or ActiveX component (Microsoft Internet Explorer).

When a user enters the desired Oracle Applications signon URL within the Web browser, Oracle JInitiator is executed. If Oracle JInitiator has not been installed, the Web browser prompts the user to download the necessary installation executable to the desktop client. Once installed, Oracle JInitiator runs the Forms client applet and starts an Oracle Applications session.

The Application Tier

The application tier hosts the servers that process the business logic and provides the communication between the desktop tier and the database tier. This tier is also referred to as the middle tier. Oracle 9*i* Application Server (9*i*AS) provides the technology stack used on the application tier. Six servers comprise the application tier for Oracle Applications:

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

Load Balancing

The application tier supports load balancing among many of its servers to provide higher availability, fault tolerance, reliability, and optimal scalability.

Load balancing occurs when there are multiple installations of the following:

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

Web Server

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

- Web Listener
- Java Servlet Engine
- Portal
- Java Server Pages (JSP)

The Web listener component of the Oracle HTTP server accepts incoming HTTP requests (URLs) from desktop clients (browsers). If possible, the Web server services the requests itself, for example, by returning an HTML 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.

Self-Service Applications Architecture

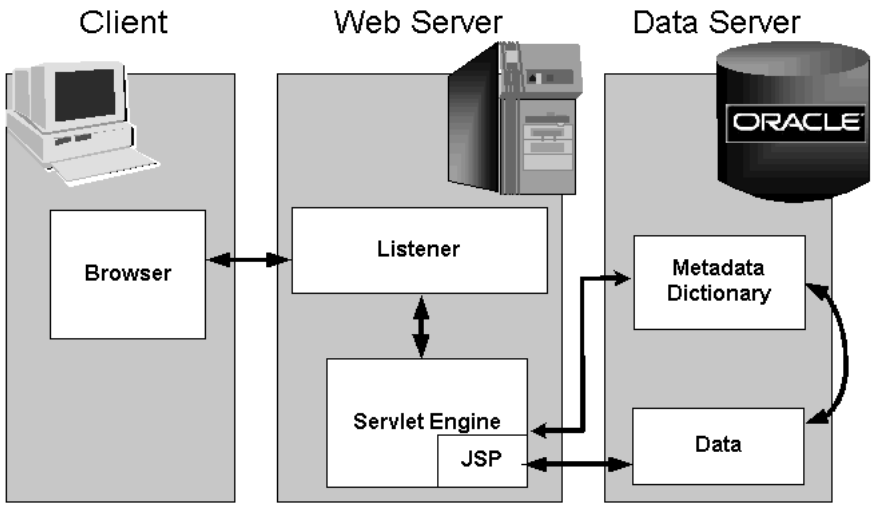
The Oracle Self-Service Applications:

- Do not use Oracle Forms for the interface
- Are designed in pure HTML and JavaScript
- Dynamically generate HTML pages by executing Java code
- Use a metadata dictionary for flexible layout
  - JRAD (Oracle Applications Extensions to JDeveloper) in OA Framework 5.7
- Operate by direct connection to the Web server

OA Framework is the development and deployment platform for self-service applications. It is a 100% Java, middle-tier application framework and services for the rapid development and deployment of HTML-based applications.

OA Framework-based applications logic is controlled by procedures that execute through the Java servlet engine. The Apache JServ module provides the Java servlet engine. For the Self-Service Applications, the Web server communicates with the database using JDBC.

Figure 1–3 Self-Service Applications Architecture



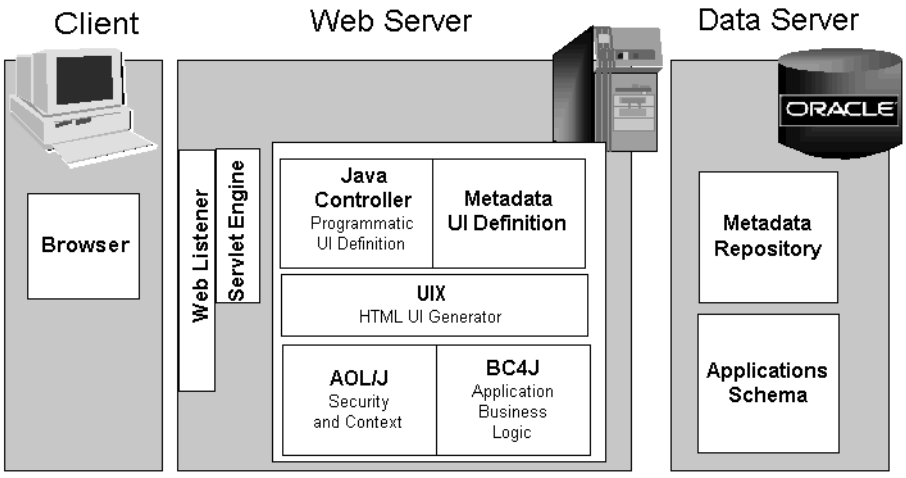
Java Servlet Access with Self-Service Applications When using a Java Server Page (JSP), the Self-Service module uses the following access path:

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

Oracle Applications Framework Architecture Following is a more detailed description of the processing that occurs in Step 4 of the previous section. Oracle Applications Framework uses additional components, including:

- BC4J (Business Components for Java), 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. It allows the separation of the application business logic from the UI.
- AOL/J provides the Oracle Applications Framework with underlying security and applications Java services. It provides the Oracle Applications Framework with its connection to the database and with application-specific functionality, such as flexfields and attachments.

Figure 1–4 Oracle Applications Framework Architecture



Oracle Applications Framework Architecture

1. AOL/J validates user access to the page.
2. The page definition is loaded from the Metadata Repository on the database tier into the applications tier (Metadata UI Definition).
3. The BC4J objects that contain the applications 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 and creates the corresponding HTML in accordance with UI standards and sends the page to the browser.

The Forms Server

The Forms server hosts the Oracle Applications forms and the forms runtime engine that support the professional interface. The Forms server is an Oracle Developer 6*i* component which mediates the communication between the desktop client and the Oracle data server by displaying client screens and initiating changes in the database records based on user actions.

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

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

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

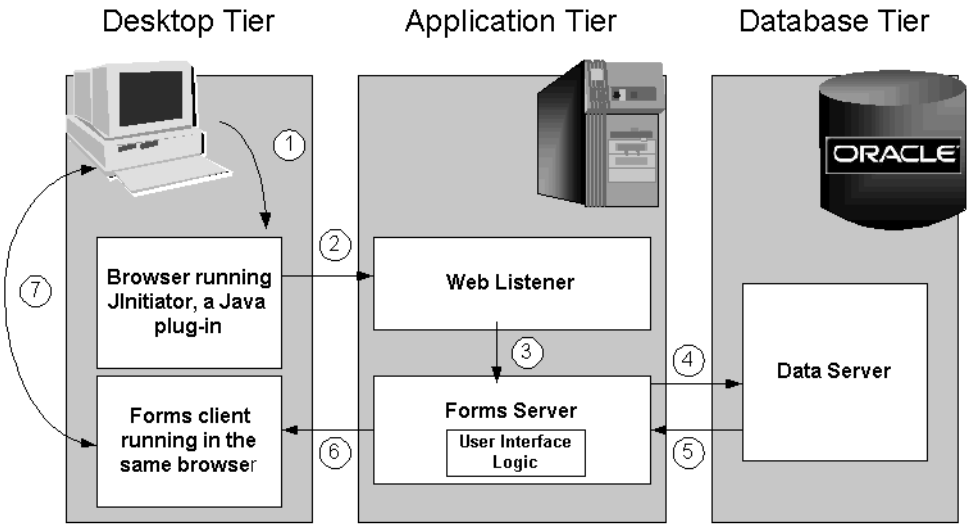
HTTP and HTTPS support enables operation over a firewall. The Forms server communicates with the Oracle data server using Net8.

Forms Server Architecture

When a user initiates an action in the Forms client applet, such as entering data into a field or clicking a button, data is passed to the Forms server on the application tier. The user interface logic runs in the Forms server and determines the appropriate user interface effect based on the user's action. For example, a window may open or another field value may be populated. If necessary, the database tier is

contacted for any data not already cached on the application tier, or for data-intensive processing.

Figure 1–5 Forms Server Architecture



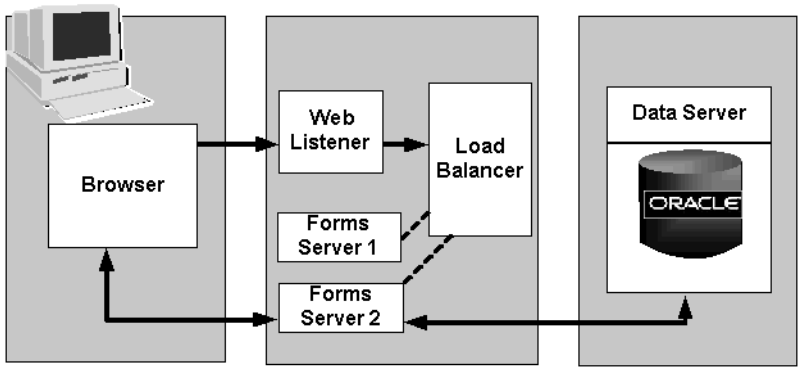
Once a connection has been made, many operations can be performed with little or no interaction with the Forms server.

For example, when a few field values change in response to a user action, there is no need to update the entire screen. In this scenario, only the changed fields are updated with the new values.

Load Balancing Among Forms Servers

Oracle Applications supports automatic load balancing among multiple Forms servers. In a load-balancing configuration, all Forms server requests have a single point of coordination called the Metrics Server, located on one application server. Metrics clients located on the other application servers periodically send load information to the Metrics Server so that it can determine which application server has the lightest load.

Figure 1–6 Forms Server Load Balancing



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

Concurrent Processing Server

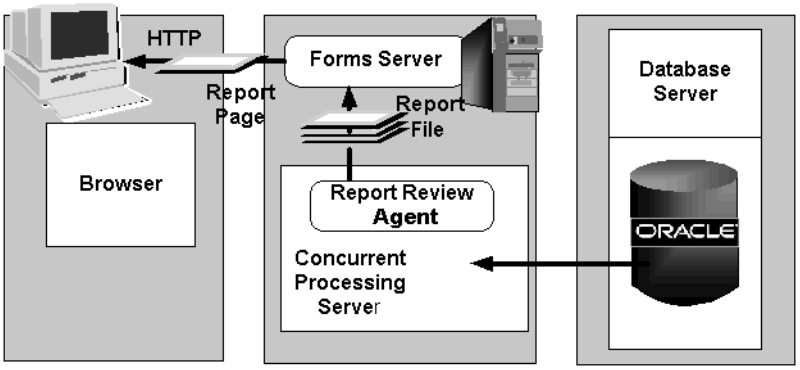
User interactions with Oracle Applications data are conducted through the Self-Service Applications and the Forms-based Professional interface. However, there are reporting programs and data updating programs that run periodically on an ad hoc basis. These programs, which operate in the background while users continue to work on other tasks, may contain a very large number of data-intensive computations. To ensure that they do not interfere with interactive operations, they are run on the Concurrent Processing server.

Processes that run on the Concurrent Processing server are called concurrent requests. When you submit a request, either through Forms or Self-Service Applications, a row is inserted into a database table specifying the program to be run. Concurrent managers read the applicable requests in the table and start the concurrent programs. The Internal Concurrent Manager service monitors the database table for new requests, controls the other concurrent managers, and

determines when a request should be processed and which concurrent manager should carry it out.

Log and output files are generated on the Concurrent Processing server.

Figure 1–7 Viewing Concurrent Processing Output



Accessing Concurrent Processing Output The Concurrent Processing server communicates with the data server using Net8. The resultant log or output file from a concurrent request is passed back to the Report Review Agent, also known as the Web Review Agent. The Report Review Agent passes a file containing the entire report to the Forms server. The Forms server passes the report back to the user's browser one page at a time. You can use profile options to control the size of the files and pages passed through the system.

Business Intelligence System (BIS)

The Business Intelligence System is a decision support solution integrated with Oracle Applications. Using the BIS products, a manager can query the Oracle Applications database to monitor recent business performance across multiple organizations. For example, a manager can set a sales goal and then use BIS to determine how close actual sales amounts are to the goal. A manager can set tolerances and have the system inform people when those tolerances are exceeded. With the BIS Performance Manager Framework, some corrective actions can be performed automatically. If, for example, sales actual amounts are more than 10% below goals, BIS can send automatic notifications to regional sales managers.

BIS products do not use the Forms server or Forms client. Instead, BIS products use the Oracle Discoverer server and Oracle Reports server on the application tier. A

Java applet running on the desktop client communicates with the HTTP server, which connects to the Oracle Discoverer server or Oracle Reports server. The Discoverer server provides ad hoc analysis; the Reports server uses summary tables such as monthly aggregates of data, and returns them to the browser as HTML.

Reports Server

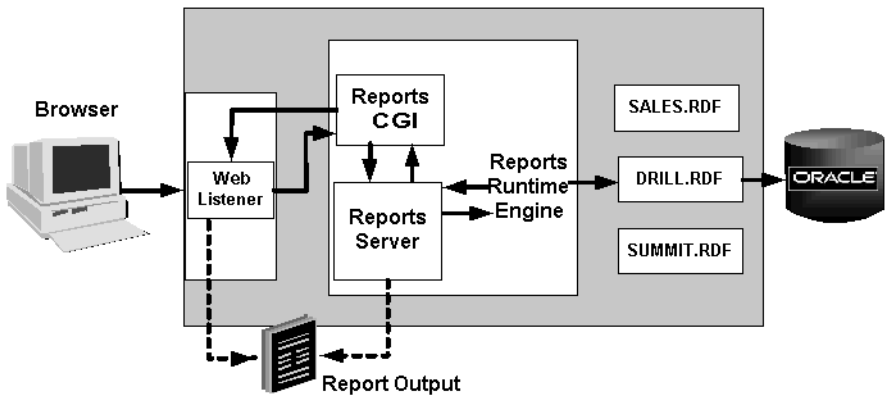
The Reports server is used primarily to produce reports for BIS products. It is automatically installed on the same node as the Concurrent Processing server and its reports are contained in the same directory with the concurrent processing reports. However, reports generated by the Reports server are monitored and administered separately from concurrent processing reports.

The Reports server dynamically selects language at runtime and users see each report in the preferred language.

Reports Server Architecture

The request for an HTML-based report is similar to the flow of a Self-Service Applications request:

Figure 1–8 Reports Server Architecture



1. The user clicks the hyperlink of a function from a browser.
2. The browser makes a URL request to the Web listener.

3. The Web listener contacts the Reports server through the reports Web CGI. The reports Web CGI allows the desktop client to run reports and see the output through the Web browser.
4. The Reports server starts the Reports runtime engine.
5. The Reports runtime engine locates the necessary reports and connects to the database to query the requested information.
6. The queried information is presented to the user in the form of an HTML page.

Clustering and Load Management

When there are a large number of BIS users, it may be beneficial to have multiple Reports servers.

In this configuration, one of the Reports servers is designated as the master server. The master Reports server receives the initial request and distributes it to one of the other Reports servers depending on the load each can handle. The master Reports server is capable of determining how many runtime engines each of the Reports servers can support.

Release 11*i* supports multiple languages on a single Reports server.

Discoverer Server

The Discoverer server complements the Reports server as a tool to perform ad hoc queries and analysis of the resultant query output. It also allows users to perform projections as various business and strategic factors are changed.

End User Layer

The End User Layer is the key to providing ease of use. The End User Layer is a layer of metadata that sits in the Oracle Applications database, but on top of the Oracle Applications data.

Oracle Discoverer hides the complexity of the database and converts it to an easy to understand interface for users. It maps the unfamiliar terms in the database, such as tables and columns, to familiar terms for the user. Therefore, analysts or managers with no database syntax familiarity can access key information using terms common to them, such as customer, name, and revenue.

The End User Layer is one of the key components of the Discoverer process. It is in the database, but is separate from the Oracle Applications data.

There are two important software packages that access the End User Layer. They are:

- Administrator Edition - allows the creation of the End User Layer and manages the administration of responsibilities and roles.
- User Edition - used to create workbooks (saved queries) and view information about the business. Oracle provides predefined workbooks which can be used as bases for ad hoc queries.

In Release 11.0, the User Edition was located on the client. For Release 11*i*, it is located in the Discoverer server. This allows for central administration of the software.

Admin Server

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

- **Upgrading Oracle Applications**

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

  Most patches consist of files and scripts that update the file system and/or database objects. You use the AutoPatch utility (adpatch) to perform these updates. AutoPatch may also be used to apply cumulative patches such as minipacks (for example, 11i.GL.D) and maintenance packs (for example, 11.5.9).
- **Maintaining Oracle Applications data**

  Some features such as Multiple Reporting Currencies require regular maintenance to ensure updates are propagated to the additional tables and schemas used by this feature. The AD Administration utility (adadmin) allows you to do this as well as other file system and database maintenance tasks.

The Database Tier

The database tier contains the database server, which stores all the data maintained by Oracle Applications. It also contains some processing code that is stored in the database to optimize performance. In Release 11*i*, the database includes the Oracle Applications online help information as well.

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

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

Features of the Oracle Applications Technology Layer

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

- Applications DBA (AD)
- Applications Object Library (AOL/FND)
- Applications Utilities (AU)
- Common Modules (AK)
- Workflow
- Alert (ALR)

Oracle Applications DBA (AD)

The Applications DBA product provides a set of tools that help in the administration of the Oracle Applications file system and database. AD provides tools for installing, upgrading, maintaining, and patching the Oracle Applications system. The AD utilities include:

- AutoUpgrade
- AutoPatch
- AD Merge Patch
- AD Administration
- License Manager
- Rapid Install

Oracle Common Modules (AK)

The Common Modules is an active data dictionary that enables you to define Oracle Applications components for the Self-Service Applications and generate many of the Oracle Applications' characteristics at runtime.

AK is heavily used to develop inquiry applications for the Self-Service Applications without programming. AK allows storing language-translated labels for all the attributes on a transaction page, thus providing Multiple Language Support.

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

Note: AK is replaced by JRAD in Framework 5.7.

Oracle Applications Utilities (AU)

The Applications Utilities product is used to maintain the Oracle Applications system.

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

The Oracle Applications Java files are stored in AU_TOP, as well as in JAVA_TOP and <PROD>_TOP. A collection of all Java files in Oracle Applications is stored in a file called apps.zip. This file is maintained under AU_TOP and copied to JAVA_TOP for runtime operation.

Oracle Workflow

Oracle Workflow allows for communication of data between products and accomplishes three important business requirements:

- Routes information
- Defines rules
- Delivers electronic notifications

Self-Service Web Expenses utilizes Workflow in its approval process. This process can be tailored for each site.

By use of Workflow, electronic notifications can be delivered via e-mail to any Oracle Applications or Internet user.

Oracle Workflow is also available as a standalone product outside of Oracle Applications.

Oracle Workflow Builder is an optional product that lets you graphically define the rules of your process, which are expressed as automated activities using PL/SQL.

Oracle Alert (ALR)

Oracle Alert allows you to electronically mail system notifications to users when an exception or event has occurred. Some products are delivered with predefined alerts, which allow you 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 e-mail to key database administrators when a tablespace in the Oracle Applications database does not have adequate free space.

Oracle Application Object Library (AOL/FND)

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

Using the Application Object Library ensures that the processing of flexfields or the procedure for report submission, for instance, does not vary from one product to another.

The Application Object Library also provides capabilities for developers to extend the operation of Oracle Applications by creating custom programs that interact with the base modules.

AOL provides many features to make system administration of Oracle Applications easier, such as security setup and maintenance, and management of concurrent processing.

AOL End User Features

Standard User Interface The Application Object Library supports the integration of Oracle Applications by providing standardized functionality and capabilities across all products so that the look and feel does not vary from product to product.

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

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

Applications Online Help The presentation of Applications Online Help is also standardized across all products and is accessed through a browser.

AOL Developer Features

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

GUI and coding standards The same coding and Graphical User Interface (GUI) standards used by Oracle Applications developers are available to custom developers.

Standard Report Submission Custom reports can be integrated into Standard Report Submission so that they can be submitted and monitored using the same procedures as other Oracle Applications reports. Developers can set up certain menus and responsibilities to access custom reports or standard objects.

Flexfield Development Flexfields used on custom forms can take advantage of existing flexfield capabilities such as value sets, validation, and security rules.

Custom menus and responsibilities Custom menus and responsibilities can be seamlessly integrated with Oracle Applications.

AOL Features for System Administrators

The Application Object Library provides many features to make administration of Oracle Applications easier. The Applications system administrator can:

Set up new users Register new Oracle Applications users and give them access to only those Forms, functions, and reports they need to do their jobs.

Manage and control security Decide which users have access to each product, and within a product, which Forms, functions, and reports a user can access.

Audit user activity Monitor what users are doing and when they do it. Choose who to audit and what type of data to audit.

Set user and system profiles A profile is a set of changeable options that affects the way Oracle Applications looks and behaves. A system administrator can set profile values at the site, application, responsibility, and user levels.

Manage concurrent processing Concurrent processing is an Oracle Applications facility that lets long-running, data-intensive tasks run simultaneously with online operations, taking full advantage of multitasking and parallel processing. A system administrator can monitor and control concurrent processing using a few simple interfaces, including Forms and Oracle Enterprise Manager (OEM).

Application Object Library Security

Application Object Library controls access to the data in Oracle Applications through user signons and responsibilities. Each user must have a user name and password to gain access to Oracle Applications.

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

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

When you install Oracle Applications, a standard Applications user called SYSADMIN is created for you. Several default responsibilities are also created. Since the SYSADMIN signon 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.

2

Release 11*i* Enhancements

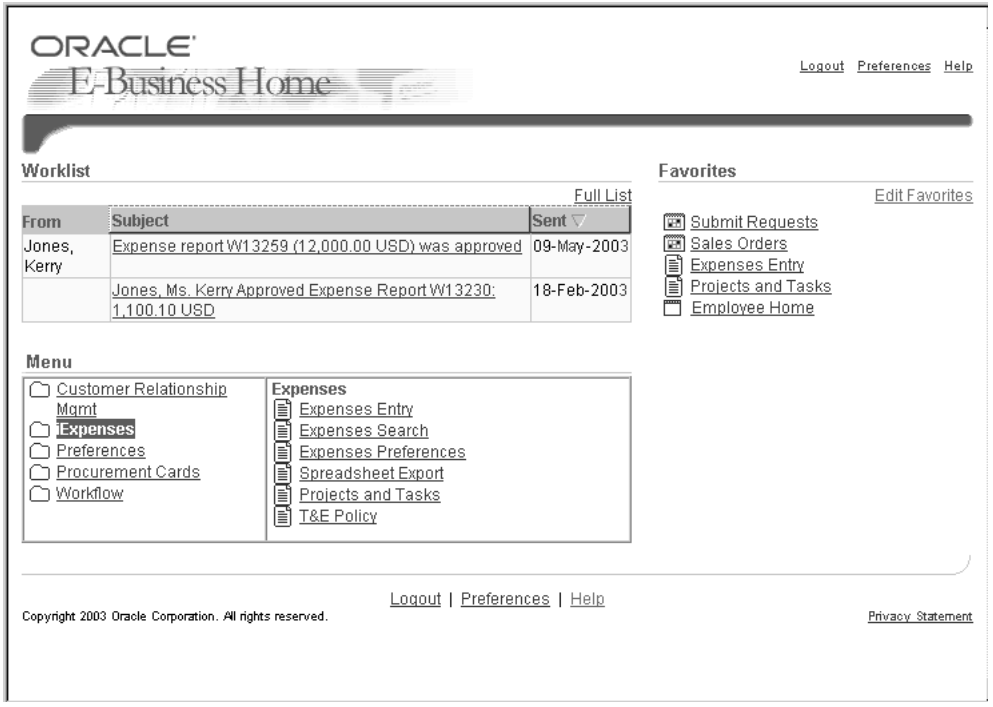
Release 11*i* provides enhancements that improve usability, increase performance, and simplify the configuration and management of Oracle Applications. The E-Business Suite Home page, new in Release 11*i*, provides users with a single point of access to all Oracle Applications products. Release 11*i* leverages the power of the Oracle technology platform to substantially increase performance and reduce network traffic. The Oracle Applications Manager provides administration of concurrent managers and centralized monitoring of the entire Oracle Applications environment.

> **Note:** Not all Release 11*i* new features are covered in the following sections. New features are embedded throughout the product suite, country-specific functionality, and supporting technologies. Additional features and enhancements are discussed in the *Oracle Applications Product Update Notes*.

The E-Business Suite Home Page

In Release 11*i*, each user logs in to Oracle Applications through the E-Business Suite Home on the desktop client. The E-Business Suite Home page provides unified navigation to access Forms-based, HTML-based Applications, or Business Intelligence products. Once logged into the E-Business Suite Home, you need not sign on again to access other parts of the system. Oracle Applications does not prompt again for user name and password, even when you navigate to other tools and products. Oracle Applications also retains preferences as you navigate through the system. For example, if you registered in the E-Business Suite Home that French is your preferred language, this preference carries over whether you access Forms-based or HTML-based products.

Figure 2–1 The E-Business Suite Home Page



Oracle Applications Manager

The Oracle Applications Manager (OAM) provides the tools to configure, monitor, and diagnose your Oracle Applications system, as well as the ability to manage all your applications tier services and processes. These tools are furnished by the following features: Oracle Workflow Manager, Patch Advisor, Patch Impact Analyzer, Patch History Database, Applications Context Editor, License Manager, and Service Fulfillment Manager.

With OAM, you can view information on general system activity including the statuses of the database, concurrent managers and other services, concurrent requests, and Oracle Workflow processes. System administrators can also start or stop services, submit concurrent requests, and view configuration information, such as initialization parameters and profile options.

Using Oracle Workflow Manager you can control Workflow system services, such as background engines, the Notification Mailer, agent listeners, queue propagation, and obsolete Workflow data purge.

Use the Patch Advisor to determine which recommended patches you should apply to keep your system current. Use the Patch Impact Analyzer to report the impact that applying these patches will have. Impact data includes: Applications covered, new files introduced by the patch, existing files changed by the patch, files in the patch that are ignored, and files in the system unchanged by the patch, but having dependencies on files that are changed by the patch.

Features Enabled by the Oracle Applications Technology Layer

Oracle9*i* Application Server and Oracle Applications

Oracle9i Application Server (Oracle9iAS) offers the industry's fastest, most complete and integrated J2EE-certified application server. Oracle9iAS has revolutionized the application server market by being the first to integrate all the technology required to develop and deploy e-business portals, transactional applications, and Web services into a single product.

Oracle Applications with Oracle 9iAS Portal

Oracle9i Application Server enables you to rapidly build and deploy personalized portals using Oracle9iAS Portal. With its intuitive Web interface, Oracle9iAS Portal allows you to integrate Web-based resources such as applications, Web pages, business intelligence reports, and syndicated content feeds within standardized information components called portlets. You can build a personalized portal to deliver all of your business information and applications with Oracle9iAS Portal.

For information on configuring Oracle Portal for integration with Oracle Applications, see Oracle*MetaLink* Note 146469.1.

Enterprise-Wide Single Sign-On

Oracle E-Business Suite now supports the use of Single Sign-On functionality via Oracle Portal, Oracle Login Server, and Oracle Internet Directory. Single Sign-On functionality allows users to access Oracle E-Business Suite applications and other applications and services from Oracle Portal through a single user ID, without having to log in to each application separately.

User information in external, third-party user directories can be synchronized with Oracle Internet Directory using Lightweight Directory Access Protocols (LDAP).

With Oracle Internet Directory, customers can manage and publish user information in a central location that various application systems, including the Oracle E-Business Suite, can reference.

For more information on implementing Single Sign-On with Oracle Login Server authentication using Oracle Internet Directory, see the *Oracle Applications System Administrator's Guide*.

Features Enabled by the Oracle Database Platform

Many improvements in Release 11*i* are built on enhancements in the underlying database technology. Release 11*i* leverages the following Oracle features to improve performance, business intelligence, and scalability.

Performance Features

Cost-Based Optimization

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

Using CBO, the optimizer considers the available access paths and factors in statistical information for the tables and indexes that the SQL statement will access. CBO also considers hints, which are optimization suggestions placed in a Comment of the SQL statement. First, the optimizer creates a set of potential execution plans for the SQL statement based on its available access paths and hints. Then the optimizer estimates the *cost* of each execution plan based on statistics in the data dictionary for the data distribution and storage characteristics of the tables, indexes, and partitions. The optimizer compares the costs of the execution plans and chooses the one with the smallest cost.

For some operations, such as batch processing, Release 11*i* uses CBO to achieve the best throughput, or the minimal resource use necessary to process all rows accessed by the statement. For other operations, such as accessing forms and communication with the desktop client, Release 11*i* uses CBO to achieve the best response time, or the minimal resource use necessary to process the first row accessed by a SQL statement.

Other Oracle9*i* performance enhancements used in Release 11*i*, such as partitioned tables, also require CBO.

> **Additional Information:** The Optimizer, *Oracle9i Concepts;* Cost-based Optimization, *Oracle Applications System Administrator's Guide*

Database Resource Manager

The Database Resource Manager gives the system administrator more control over processing resources on the database node. With the Database Resource Manager, the system administrator can distribute server CPU based on business rules, and thereby ensure that the highest priority business activities always have sufficient CPU resources. Administrators can also limit the impact of any inefficient adhoc queries on the overall system.

Using the Database Resource Manager, the system administrator might, for example, limit ad hoc queries on the database to consume no more that 5% of CPU usage. The system administrator can 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.

Partitioned Tables

Partitioning helps support very large tables and indexes by dividing them into smaller and more manageable pieces, which are called partitions. Once partitions are defined, SQL statements can access and manipulate them rather than entire tables or indexes. Partitioning reduces access time, and partitions are especially useful in data warehouse applications, which often store and analyze large amounts of historical data.

For example, operations that involve copying or deleting data are now improved because Release 11*i* products use partitioned tables. Creating and deleting all rows of a partitioned table is a much faster operation than selectively inserting rows into and selectively deleting rows from an existing table. Operations in some products that, in earlier releases, could potentially take hours are now reduced to seconds.

Invoker Rights

In earlier releases, if you had Multiple Reporting Currencies (MRC) or Multiple Sets of Books Architecture (MSOBA), several copies of Oracle Applications packages existed in the database. This took more database space and required extra time to upgrade and maintain.

Release 11*i* uses the Invoker Rights functionality of Oracle 9*i* to ensure that most packages are installed only in the APPS schema. Other schemas, such as the MRC schema, have synonyms to the packages in the APPS schema, and the corresponding packages in the APPS schema have grants to the MRC schema. PL/SQL routines use Oracle9*i* Invoker Rights to access the package in the APPS schema. In an MRC database, this can very markedly decrease the size of the database and shorten the time spent by upgrade, patch, and maintenance tasks.

Additional Information: *PL/SQL User's Guide and Reference*

Business Intelligence

Materialized Views

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

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

In distributed environments, materialized views are used to replicate data at distributed sites and synchronize updates done at several sites with conflict resolution methods. As replicas, they provide local access to data which otherwise would have to be accessed from remote sites.

Scalability

Real Application Clusters

Real Application Clusters (RAC) harnesses the processing power of multiple interconnected computers. RAC software and a collection of hardware known as a *cluster* unite the processing power of each component to create a robust computing environment.

In RAC environments, all active instances can concurrently execute transactions against a shared database. RAC coordinates each instance's access to the shared data to provide data consistency and data integrity.

Harnessing the power of clusters offers obvious advantages. A large task divided into subtasks and distributed among multiple nodes is completed sooner and more efficiently than if you processed the entire task on one node. Cluster processing also provides increased performance for larger workloads and for accommodating rapidly growing user populations.

With RAC, you can scale applications to meet increasing data processing demands without changing the application code. As you add resources such as nodes or storage, RAC extends the processing powers of these resources beyond the limits of the individual components.

3

Internationalization Support

In Release 11*i*, the Oracle E-Business Suite enables you to create a single global instance that meets the complex requirements of a worldwide enterprise. Key to meeting those requirements is the level of internationalization support and flexibility that release 11i provides. Internationalization includes support for many languages and territories, flexible date and number formats that conform to local custom, and other country-specific functionality such as multiple reporting currencies and compliance with local statutory requirements.

Language Support

In Release 10.7 you could install Oracle Applications in only one language per instance. If you needed to run Oracle Applications in more than one language in the same instance, Oracle Consulting offered a customized solution. With Release 11 you could install and run Oracle Applications in more than one language, but the set of languages you could install was limited to those supported by the database character set.

Textual parts of Oracle Applications such as forms, reports, messages, help text, menu prompts, and lists of report names were available in all installed languages, but most data at the product level was still available only in the base, or primary language. This meant, for example, you could enter payment terms only in the base language, even though forms would be displayed in a non-base language. For additional multilingual support, Oracle Consulting continued to provide a customized solution.

The majority of Oracle Applications products have been restructured in Release 11*i* to provide multilingual support at the product data level. In Release 11*i*, support for the Unicode UTF8 character set removes the limitation on the number of supported languages that can be run in a single instance. The Unicode character set supports all characters in common use in all of the world's modern languages. The additional

multilingual support available from Oracle Consulting is incorporated in Release 11*i*.

> **Additional Information:** Set Up National Language Support (NLS) in Finishing Your Installation, *Installing Oracle Applications*

Languages and Character Sets on the Database Tier

The Oracle database tier is installed in the US7ASCII character set by default, but can be converted to run in any other supported character set. You choose the database character set when running Rapid Install, and Rapid Install converts the database to the new character set.

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 run Oracle Applications in English and French, you might choose WE8ISO8859P15 as the database character set when running Rapid Install. WE8ISO8859P15 is a superset of US7ASCII, supports both English and French, and contains the euro symbol. If you need to support English, French, Japanese, and Arabic, you must choose the UTF8 character set, because this is the only one that supports these four languages.

> **Additional Information:** The *Oracle9i Globalization Support Guide* provides information on supported character sets, languages supported by each character set, and tips on choosing a database character set.

Warning : Before installing Oracle Applications, you should carefully consider the future language requirements of your installation. The character set you choose during installation determines the languages that you can support. Review the *Oracle 9i Globalization Support Guide* for information on all possible character sets before choosing the character set for your installation. Changing character sets after installation is an involved and expensive process, and is best avoided by initially choosing a character set that will meet your long-term needs.

You cannot change the character set during an upgrade from an earlier release to Release 11*i*. You must first upgrade to Release 11*i* using the existing character set and, after the upgrade, change the character set.

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

> **Additional Information:** Overview of an Upgrade, *Upgrading Oracle Applications*

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

Language and Character Sets on the Application Tier

The application tier is installed in the US7ASCII character set by default, but can be converted to run in any supported character set. Rapid Install recommends the application tier character set based on the the languages chosen. To prevent data loss, character sets on all tiers must either be the same, or be character sets that can be converted from one to another. Some character sets allow a conversion with no data loss because the character representation in one corresponds to an appropriate character representation in the other. For example, JA16SJIS and JA16EUC are both Japanese language character sets and allow for conversion with no data loss. If a target character set does not contain all characters in the source data, replacement characters will be used and data is thereby lost.

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

The HTTP servers on the application tier must use a character set supported by the browsers on the desktop tier. Not all character sets available for the HTTP server are supported by the browsers. This is the only compatibility requirement between the desktop tier and application tier. All other application tier servers, such as the

Concurrent Processing server, can be configured with any other character set that is compatible with the database server.

> **Attention:** As noted earlier concerning the database tier character set, the character set on the application tier should also meet your worldwide language requirements in the future. Changing character sets for the application tier after installation is a difficult process.

By default, Rapid Install installs American English on all servers in the application tier. When you later install a National Language Support (NLS) release, you must install it on all servers. You cannot, for example, install French only on the forms server with the assumption that you will not run the reports or concurrent processing server in French. All application tier servers must have the same set of languages installed.

Character Sets on the Desktop Tier

Language support, which includes support for data input methods and required character sets and fonts, must be available in the desktop client's operating system. If Unicode UTF8 is installed on the applications tier, the desktop client operating system must support Unicode.

The desktop browser must be configured to input data in the required language and must handle any 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.

The character set in the browser is set by the HTTP server. Users must not change the character set in the browser during an Oracle Applications session.

External Documents

External documents are those documents intended for customers and trading partners, such as bills of lading, commercial invoices, and packing slips. In Release 11*i*, you can produce many external documents in any of the active languages, simultaneously and with a single request. A customer in Italy, for example, can receive invoices printed in Italian, and a customer in Poland can receive their invoices printed in Polish. You can also print the documents to different printers based on language, and route completion notifications to different people according to the requested language. For instance, you can route all French external

documents to printer A and all others to printer B. You can send completion notifications for Spanish documents to one user, and perhaps all notifications, including Spanish, to another. See "Multilingual External Documents" in the *System Administrator's Guide* for a list of external documents provided in Release 11*i*.

Globalization and Localization

Globalization is the process of designing software that meets all of the needs of a global enterprise in a single instance. One way that globalization is achieved is through national and regional extensions called "localizations." Each localization meets the statutory, legal, and/or cultural practices of a given locale. Because all localizations are compatible with each other, the installation of all localizations together results in a fully globalized implementation.

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

Dates and Numbers

You can enter and view dates in any valid format, such as 11/25/01 or 11-25-2001. Any format for which SQL provides a mask is valid. The only exception to flexible date formats is that Oracle Reports will always display DD-MON-RRRR.

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

> **Additional Information:** Date and Time Parameters, Numeric Parameters, *Oracle9i Globalization Support Guide*

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

Multiple Reporting Currencies

The Multiple Reporting Currencies (MRC) feature allows you to report and maintain accounting records at the transaction level in more than one functional currency. You do this by defining one or more reporting sets of books, in addition to your primary set of books.

In your reporting sets of books, you maintain records in a functional currency other than your primary functional currency. Primary functional currency is the currency you use to record transactions and maintain your accounting data within Oracle Applications. The primary functional currency is generally the currency in which you perform most of your business transactions and the one you use for legal reporting. A reporting functional currency is a currency, other than your primary functional currency, that you need for reports.

MRC is based on Multiple Organization Architecture (see *Multiple Organizations in Oracle Applications*), and requires a primary set of books and a reporting set of books. In the primary set of books, the functional currency is always the primary functional currency. The reporting set of books is a financial reporting entity associated with a primary set of books. The reporting set of books has the same chart of accounts and accounting calendar as the primary set of books, but usually has a different functional currency. The reporting set of books allows you to report in a different functional currency than that of your primary set of books.

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

> **Additional Information:** *Multiple Reporting Currencies in Oracle Applications*
>
> *Multiple Organizations in Oracle Applications*

NLS-Independent Application Servers

In Release 11, an application tier server was required for each language and territory configuration a user might have. For example, to process French and German forms requests, you needed to start one forms server for French and one for German. Even if two users both ran French, but one set the territory to France and the other to Switzerland, you would need to install two forms servers and two reports servers to support these two users. In Release 11*i*, you no longer need to set

up a server for each permutation of NLS preferences. All application tier server processes can start with any NLS configuration.

Application tier processes must be started with the same character set that was chosen for the server in Rapid Install. All other user NLS settings (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 those NLS settings.

NLS Settings

Earlier releases relied on operating system environment settings for runtime NLS requirements. In Release 11*i*, user runtime NLS settings are stored as profile option values in the database.

The profile options for language and territory are configured at site level when running Rapid Install. The language you choose for the base language is used for the language profile option. The default user territory you choose is used for the territory profile option. Rapid Install does not set date and numeric formats. Based on the territory profile setting, default Oracle database date and numeric formats are used. Although the system administrator can reset date and numeric formats after Rapid Install, we recommend that you accept the defaults provided by the territory setting.

The site level profile values provide the default NLS settings for all end users. Users inherit these values the first time they log on to Oracle Applications using the E-Business Suite Homepage. A user can continue to use the default values or change any of the NLS settings to alternate values. The updated values are stored in the database at the profile user level. The user's current session is reset to use the updated user level profile values and all future sessions will be started with the new values.

4

File System

Oracle Applications product files, technology stack files, environment files, and common files are held in the *file system* on the application tier. In Release 11*i*, only the downloaded JAR files are stored on the desktop client and the database server holds only database files. Environment settings indicate the location of files in the file system.

This chapter discusses the environment settings and file systems in detail, including the following topics:

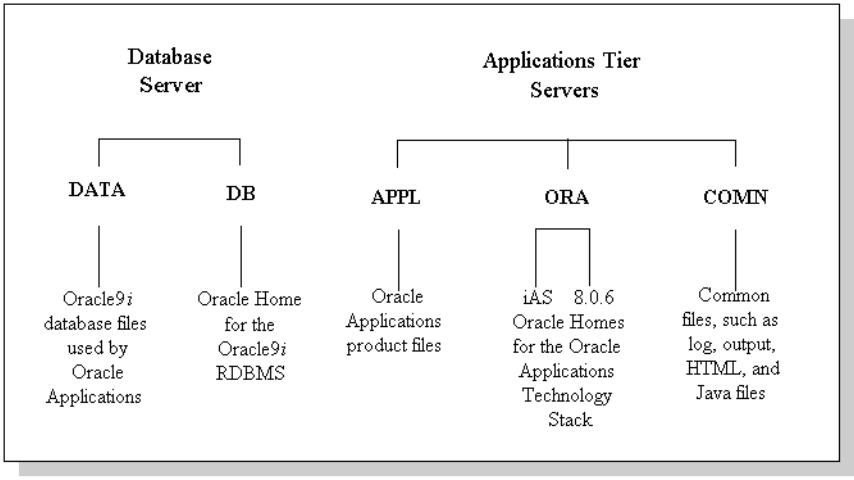
- Top-Level Directories
- Product Directories
- Environment Settings

Top-Level Directories

The Oracle Applications 11*i* system uses components from many Oracle products. It stores these product files within several top-level directories as follows :

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

Figure 4–1 Top-Level Directories



- The <*dbname*>DATA or DATA_TOP directory is located on the database server and it contains the system tablespaces, redo log files, data tablespaces, index tablespaces, and database files.
- The <*dbname*>DB directory is located on the database server and it contains the Oracle Home for the Oracle9*i* RDBMS.
- The <*dbname*>APPL or APPL_TOP directory contains the product directories and files for Oracle Applications.
- The <*dbname*>ORA directory contains the Oracle Homes for the Applications technology stack components.
- The <*dbname*>COMN or COMN_TOP (or COMMON_TOP) directory contains directories and files used across products.

The <*dbname*>DATA Directory

The <*dbname*>DATA file system contains the .dbf files of the Oracle9*i* database. Rapid Install installs all the system, data, and index files in up to four different mount points on the database server. You can specify the directory names of the mount points on the database server during installation.

<*dbname*>DB and <*dbname*>ORA Directories

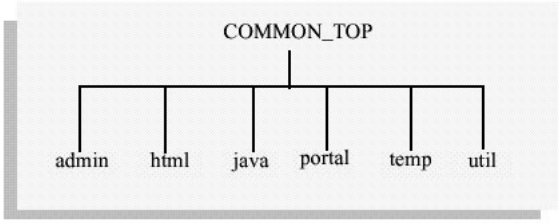
Oracle Applications supports running a database of one version, while linking Oracle Applications programs using the tools from a second or third version of the database server. This is known as *multiple Oracle Homes*. This model allows Oracle to support features in later database server versions and still maintain compatibility with an earlier release. Release 11*i* has three Oracle Homes.

The 9.2.0 ORACLE_HOME (or database home) is located in the <*dbname*>DB directory. It contains the files for running and maintaining the Oracle9*i* database on the database server.

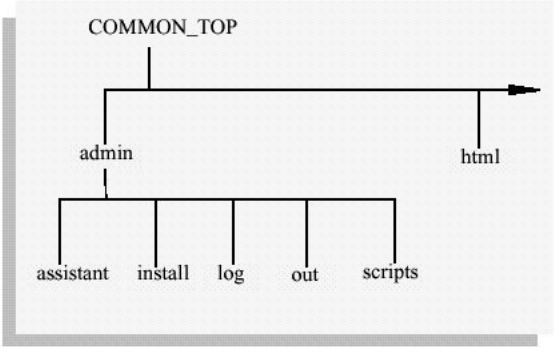
To take advantage of the latest features of the applications technology stack components, there are two ORACLE_HOME directories under the <*dbname*>ORA directory. The 8.0.6 directory contains the ORACLE_HOME for the Developer 6*i* products (Forms, Reports, and Graphics). The iAS directory contains the ORACLE_HOME for the Oracle HTTP Server.

The <*dbname*>COMN Directory

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



The admin directory in the COMMON_TOP directory contains the log and output directories for concurrent managers. When the concurrent managers run Oracle Applications reports, they write the output files, as well as diagnostic log files and temporary files, to the log and out directories in this 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. The default, however, is to write the files to the log and out directories in the COMMON_TOP/admin/log and COMMON_TOP/admin/out directories.

Additional Information: Overview of Concurrent Processing, *Oracle Applications System Administrator's Guide*

The admin/install directory contains scripts and log files used by Rapid Install during installation. The admin/scripts directory contains scripts to start and stop services such as listeners and concurrent managers.

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

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

Most Java code used by Oracle Applications is version-controlled in the apps.zip file contained in the AU_TOP directory. Patches, for example, update individual classes in apps.zip under the AU_TOP directory, and from this apps.zip file JAR files are generated both in the JAVA_TOP and the <*prod*>_TOP directories. The same apps.zip file exists in both the AU_TOP and JAVA_TOP directories.

The portal directory The portal directory contains the Rapid Install Portal files. The Rapid Install Portal is a web page that includes the post-install tasks that may be necessary for your installation, Server Administration scripts, installation documentation, and online help. Using a browser, you can view the Rapid Install Portal after you run Rapid Install.

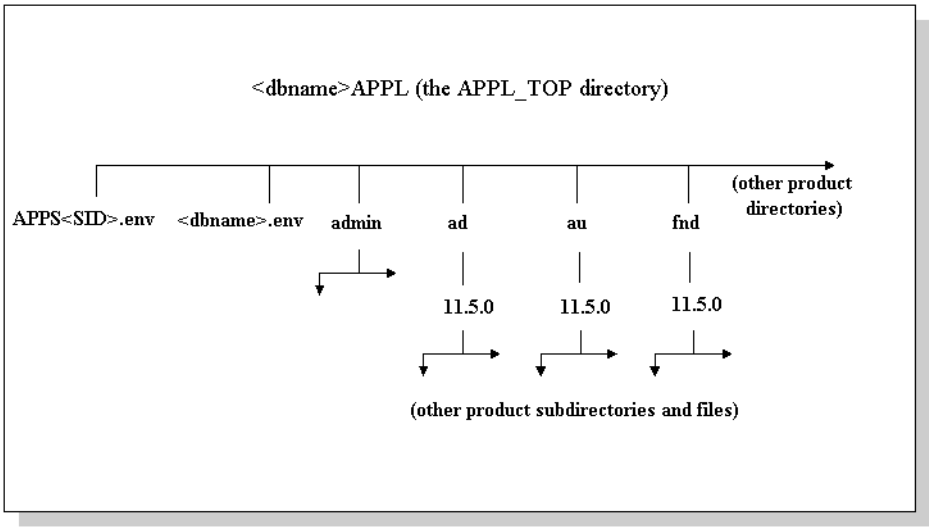
> **Additional Information:** Accessing the Rapid Install Portal, *Installing Oracle Applications*

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

The util directory The util directory contains the third-party utilities licensed to ship with Oracle Applications. These include, for example, JRE, JDK, and the unzip utility.

The <*dbname*>APPL Directory

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



The APPL_TOP directory contains:

- The main applications environment file, called the <*dbname*>.env file
- The consolidated environment file, APPS<SID>.env (where SID is the name of the database)
- The core technology directories
- The product directories for all products

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

> **Attention:** For Release 11*i*, all Oracle Applications products are installed in the database and the file system, regardless of registration. Do not attempt to manually remove files for unregistered products.

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

Each Applications top directory is associated with a single Oracle Applications database on the Oracle9*i* server. If you install both a Vision Demo system and a test system, you must use Rapid Install to lay down two file systems: one for each system.

Product Directories

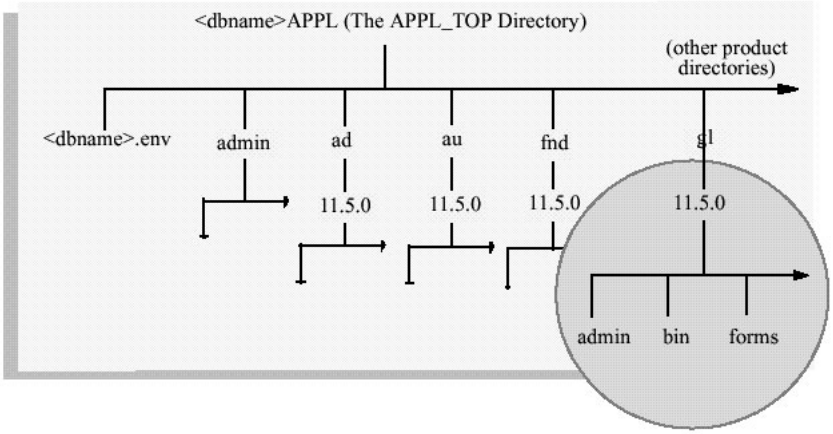
Each product has its own subdirectory under APPL_TOP named for its standard abbreviation, such as gl for Oracle General Ledger. Within each product directory, is a subdirectory named for the base release version number, such as 11.5.0. Within this directory are the various subdirectories for the product files.

The <PROD>_TOP

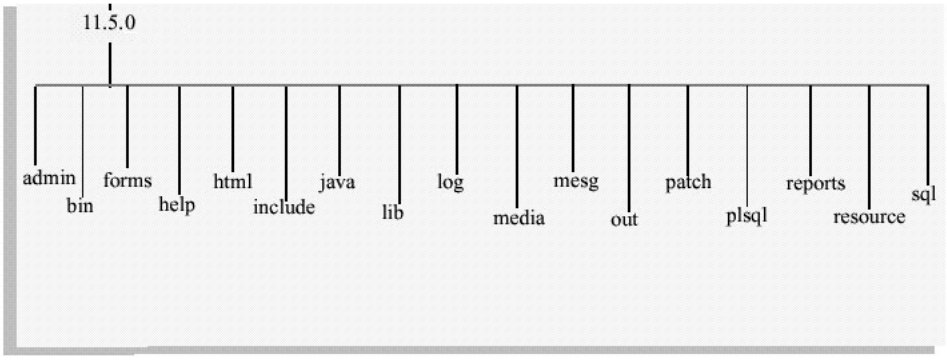
The <APPL_TOP>/<prod>/<version> path is known as the product top (<PROD>_TOP) and it is stored in the <*prod*>_TOP environment setting. For example, if APPL_TOP=/d01/oracle/prodapps, then the value contained in the AD_TOP environment setting is /d01/oracle/prodapps/ad/11.5.0 and the AD_TOP environment setting points to the <APPL_TOP>/ad/11.5.0 directory. Similarly, the value of AU_TOP is /d01/oracle/prodapps/au/11.5.0, and the AU_TOP environment setting points to the <APPL_TOP>/au/11.5.0 directory. This is the same for all directories except for the admin directory.

Product Files

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



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 (also listed in the following table).



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

table.

Table 4–1 Applications Directory and File Types

Subdirectory Name	Description
admin	PROD_TOP/admin contains files used by AutoUpgrade to upgrade each product. Note that this directory is not the APPL_TOP/admin directory. The APPL_TOP/admin directory contains upgrade-related files for all products. This PROD_TOP/admin directory contains product-specific upgrade files.
■ driver	Contains .drv files (driver files). AutoUpgrade processes in several phases, and each phase is controlled by a driver file.
■ import	Contains DataMerge files used to upgrade seed data.
■ odf	Contains object description files (.odf files) used to create tables and other database objects.
■ sql	Contains SQL*Plus scripts used to upgrade data, and .pkh, .pkb, and .pls scripts to create PL/SQL stored procedures.
bin	Contains concurrent programs, other C language programs and shell scripts for each product.
forms	Contains Oracle Forms generated runtime (.fmx) files (Oracle Forms form files).
help	Contains the online help source files. Within this directory are subdirectories for each language you choose to install.
html	HTML, Javascript, and Java Serve Page files, primarily for Self-Service 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 JAVA_TOP.
lib	Contains files used to relink concurrent programs with the Oracle server libraries. These files include: ■ object files (.o) with compiled code specific to one of the product's programs; ■ a library file (.a) with compiled code common to the product's programs; ■ a makefile (.mk) that specifies how to link executables.

Table 4–1 Applications Directory and File Types

Subdirectory Name	Description
log and out	Contains output files for concurrent programs: ■ .mgr (master log file for concurrent manager) ■ .req (log file for a concurrent process) **Note**: Log and out subdirectories under a product directory are not used if you set up a common directory for log and output files (except in FND_TOP).
media	The desktop client displays text and graphics from the .gif files in this directory.
mesg	Forms display messages at the bottom of the screen and in popup boxes. Concurrent programs also print messages in the log and output files. These messages are translated and stored in message files separate from the forms and concurrent programs. This directory contains the .msb files (binary message files used at runtime), and language-specific message files (such as a US.msb file for American English and a D.msb file for German.)
patch	Updates to the data or data model use this directory to store the patch files.
plsql	Location where .pll files (PL/SQL library files for Oracle Reports) are unloaded, later in the installation they are moved to the plsql subdirectory in the AU_TOP directory.
reports	Contains Oracle Reports .rdf files for each product, which are platform-specific binary report files. 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.

Language Files

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

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

Additional Information: *Oracle9i Globalization Support Guide*

Core Technology Directories

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

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

- The adovars.env environment file that defines certain file and directory locations
- Text files read by AutoUpgrade
- Scripts run during the upgrade
- A preupg directory containing product-specific preupgrade scripts
- Log and out directories for upgrade log and output files
- A restart directory where AD programs create restart files

The ad directory (Applications DBA) contains the installation and maintenance utilities such as AutoUpgrade (adaimgr), AutoPatch (adpatch), and AD Administration (adadmin).

The au directory (Applications Utilities) directory contains product files that are consolidated in a single location for optimal processing. These files include:

- PL/SQL libraries used by Oracle Reports, in the plsql subdirectory
- PL/SQL libraries used by Oracle Forms, in the resource subdirectory

- Oracle Forms source files, in the forms subdirectory
- A copy of all Java files used by JInitiator when regenerating the desktop client JAR files, in the java subdirectory
- Certain reports needed by Discoverer or BIS, in the reports subdirectory

The fnd directory (Application Object Library) contains the scripts and programs that are used as the foundation for all Applications products to build data dictionaries, forms and C object libraries.

Distributing the APPL_TOP Across Disks

The Oracle Applications file system on the application tier requires a significant amount of disk space. If you choose multiple mount points in Rapid Install, you can distribute the APPL_TOP files across as many as four disk drives. You can also install the OA_HTML, JAVA_TOP, and COMMON_TOP on different mount points than the APPL_TOP.

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

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

Environment Settings

Oracle Applications uses environment settings to control program execution. The environment settings are defined when you install Oracle Applications. Many settings are defined by information you provide when running Rapid Install, though other settings have constant values for all installations.

Rapid Install creates four different environment setup files that set up the Oracle9*i* database, the Oracle 8-based technology stack, the Oracle HTTP server, and the Oracle Applications environments. The environment setup files are called <*dbname*>.env in UNIX or <*dbname*>.cmd in Windows, where <*dbname*> is the name of your database. The location of these files is shown in the following table:

Filename	Location	Environment
<dbname>.env or <dbname>.cmd	9.2.0 ORACLE_HOME	Oracle9*i* Enterprise Edition
<dbname>.env or <dbname>.cmd	*i*AS ORACLE_HOME	HTTP Server
<dbname>.env or <dbname>.cmd	8.0.6 ORACLE_HOME	Oracle 8-based technology stack
<dbname>.env or <dbname>.cmd	APPL_TOP	Applications
APPS<SID>.env or APPS<SID>.cmd	APPL_TOP	Consolidated setup file

For convenience, Oracle Applications provides a consolidated environment file APPS<SID>.env in UNIX or APPS<SID>.cmd in Windows that sets up both the Oracle Applications and Oracle 8-based technology stack environments.

The main Oracle Applications environment file is a script that sets the environment for running Oracle Applications. When you install Oracle Applications, Rapid Install creates this script in your APPL_TOP directory. Some of the variables can be changed during the install process.

Key parameters in the Oracle Applications main environment file are described in the following table:

Parameter	Description
APPLFENV	The name of this environment file, <dbname>.env. If you rename the environment file, change this parameter.
PLATFORM	The execution platform. The value should match the value in the APPL_TOP/admin/adpltfrm.txt.
APPL_TOP	The top-level directory for this Oracle Applications installation.
FNDNAM	The name of the ORACLE schema to which the System Administration responsibility connects. The default is APPS.
GWYUID	The public ORACLE username and password that grants access to the Oracle Applications initial signon form. The default is APPLSYSPUB/PUB.
FND_TOP	The path to the Application Object Library directory.
AU_TOP	The path to the Applications Utilities directory.
<PROD>_TOP	The path to a product's top directory. There is one entry for each Oracle Applications product.
PATH	Sets the directory search path, primarily FND_TOP and AD_TOP.
APPLDCP	Specifies whether distributed concurrent processing is being used. Distributed concurrent processing distributes processing load across multiple concurrent processors.
APPCPNAM	Indicates whether the format of the concurrent manager log and output files follow 8.3 file name conventions (maximum of 8 characters to the left of the dot and 3 to the right, for example, alogfile.log). If this parameter is set to "REQID", the concurrent manager uses file names that meet 8.3 naming requirements.
APPLCSF	Identifies the top-level directory for concurrent manager log and output files if they are consolidated into a single directory across all products. For example, if the path is <COMN_TOP>/admin, all log and output files are placed in a subdirectory under this defined directory.
APPLLOG	The subdirectory for concurrent manager log files. The default is "log".
APPLOUT	The subdirectory for concurrent manager output files. The default is "out".
APPLTMP	Identifies the directory for Oracle Applications temporary files. The default is "usr/tmp" for UNIX and "C:\temp" for NT.

Parameter	Description
APPLPTMP	Identifies the directory for temporary PL/SQL output files. The possible directory options must be listed in the init.ora parameter "utl_file_dir".
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".
NLS_NUMERIC_CHARACTERS	The National Language Support numeric separators. The default is ".,".
FORMS60_MAPPING	Specifies the Oracle Applications forms Web server host name and the port on which it is running. For example: http://ap9000sun.oracle.com:8000/OA_TEMP
REPORTS60_TMP	The directory for temporary files used by Oracle Reports.
GRAPHICS60_PATH	Specifies the path to the Oracle Graphics files.

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

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

Rapid Install sets both APPLPTMP and the utl_file_dir parameter to the same default directory. As the temporary files placed in this directory may contain context sensitive information, it should be a secure directory on the database server node with read and write access for the database server owner. In a multi-node system, the directory defined by APPLPTMP does not need to exist on the

application tier servers. During an upgrade with AutoUpgrade, you must provide the utl_file_dir parameter value for the APPLPTMP environment setting.

> **Attention:** Some Oracle Applications utilities use your operating system's default temporary directory even if you define the environment settings listed in the previous paragraph. Be sure to have available disk space for these default directories as well as those denoted by APPLTMP, REPORTS60_TMP, and APPLPTMP.

Other Environment Files

The adovars.env File 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 (<dbname>.env). The adovars.env file is self-documented and provides comments on what each variable does and how it should be configured. You can manually edit most of the variables in this file. Any additional custom variables should be stored here and not in the main environment file, so that you do not have to reapply the custom variables if you have to recreate your main environment file.

The adovars.env file contains:

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

The adconfig.txt File AD utility programs perform a variety of database and file management tasks. These utilities need to know certain configuration information to run successfully. This configuration information is specified when Oracle Applications is installed and subsequently stored in the adconfig.txt file, which is

located in the APPL_TOP/admin directory. Once created, this file is used by other Oracle Applications utilities.

> **Note:** This file is not the same as the configuration file created by Rapid Install (config.txt). The adconfig.txt file is created during the creation of the APPL_TOP file system.

The fndenv.env File sets additional environment variables used by the Application Object Library. For example, it sets APPLBIN as the name of the subdirectory where product executable programs and shell scripts are stored ("bin").

This file should not be modified. The default values should be applicable for all customers. The file is located in the FND_TOP directory.

The devenv.env File sets variables that let you link third-party software and your own custom-developed applications with Oracle Applications. This script is located in FND_TOP and is automatically called by fndenv.env so that you can compile and link custom Oracle Forms and Reports user exits and concurrent programs with Oracle Applications.

5

Data Model

This chapter describes the Oracle Applications data model. It includes a discussion of the following topics:

- Basic Data Model
- Multiple Organization (Multi-Org) Architecture
- Multiple Reporting Currencies
- Oracle9i Server Organization and Requirements

Basic Data Model

A database can contain only one installation of Oracle E-Business Suite products. Generally, product data objects are stored in product Oracle schemas and product code objects are stored in the APPS schema.

Oracle User IDs

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

> **Attention:** We recommend you do not change the default user IDs, but you should change the password.

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, there are necessary grants and synonyms between the APPS schema and the base product schemas.

APPS Schema

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

APPS and Base Product Schemas

There is one schema for each product—a base product schema—that owns the data objects for that product. There is one schema—APPS—that owns all the code objects for the E-Business Suite and has access to all data objects. There is one APPS schema for every product installation group. The following code objects are installed in the APPS schema:

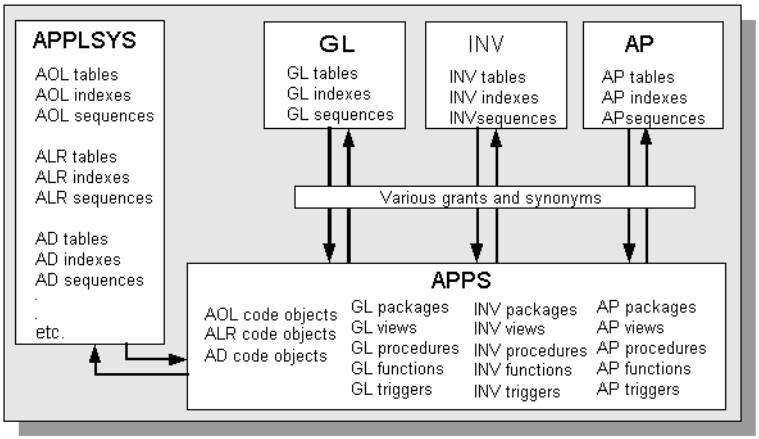
- Views
- Packages
- Procedures
- Functions
- Triggers

The following objects are installed in the base product schemas:

- Tables
- Sequences
- Indexes
- Constraints

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

Figure 5–1 APPS Schema and Base Product Schemas



APPS Schema Benefits

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

Custom Schema Access

In some circumstances you may wish to create a schema that has limited or read-only access to Oracle Applications data. For example, you might do this so your users can query Oracle Applications using a data query tool like SQL*Plus.

> **Warning: Since the APPS schema has all privileges to all Oracle Applications objects, you should not give users direct access to this schema.**

Grant access on objects to the user schema from the base product schema.

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

Data Access

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

If you connect directly to a schema, the rows returned by the view may be different from those returned if you are running in an Oracle Applications environment. For example, some views reference profile options. When accessed from SQL*Plus directly, the site value of the profile option is used, not the setting for a particular applications user.

Multiple Organization (Multi-Org) Architecture

Multiple Organization architecture determines how transactions flow through different organizations in Oracle Applications and how those organizations interact with each other.

Overview

The Multiple Organization architecture uses a single installation of Oracle Applications to support any number of organizations, even if those organizations use different sets of books.

You define multiple organizations and their relationships. These organizations can be sets of books, business groups, legal entities, operating units, or inventory organizations.

When you run any Oracle Applications product, you first choose an organization—either implicitly by choosing a responsibility, or explicitly in a Choose Organization window. Each window and report displays information only for your organization.

It is important that you understand these terms as you work with Multi-Org architecture:

Set of Books	A financial reporting entity that uses a particular chart of accounts, functional currency, and accounting calendar. Oracle General Ledger secures transaction information (journal entries, balances) by set of books. When you use Oracle General Ledger, you choose a responsibility that specifies a set of books. You then see information only for that set of books.
Operating Unit	An organization that uses Oracle Cash Management, Oracle Order Management and Shipping Execution, Oracle Payables, Oracle Purchasing, and Oracle Receivables. It may be a sales office, a division, or a department. An operating unit is associated with a legal entity. Information is secured by operating unit for these applications; each user sees information only for their operating unit. To run any of these applications, you choose a responsibility associated with an organization classified as an operating unit.

Additional Information: *Multiple Organizations in Oracle Applications*

Multi-Org Partitioned Objects

Tables that contain Multi-Org data have the suffix "_ALL" in the table name. These tables include a column called ORG_ID that partitions Multi-Org data by organization.

Every Multi-Org table has a corresponding view that partitions the table's data by operating unit. Multi-Org views partition data by including a DECODE on the internal variable CLIENT_INFO. CLIENT_INFO is similar in operation to the variable LANGUAGE, which returns the language of the current session, and is set by the security system to the operating unit designated for the responsibility.

Note: If accessing data from a Multi-Org partitioned object when CLIENT.INFO has not been set (for example, from SQL*Plus), use the _ALL table, not the view.

SO_HEADERS_ALL, with its corresponding view SO_HEADERS, is an example of a Multi-Org partitioned object.

Additional Information: Technical Overview, *Multiple Organizations in Oracle Applications*

Convert to Multi-Org

For a fresh install, the data model is identical whether you implement Multi-Org or not. Tables that contain Multi-Org partitioned data use the suffix "_ALL." When you convert, tables are renamed to use this suffix.

Multi-Org views partition data using CLIENT_INFO. These views are incorporated into the normal install and use predefined text.

In Release 11*i*, you use AD Administration to initially convert seed data and transaction data to Multi-Org. The AD Administration utility does not change the data model. When you create a new operating unit, a concurrent program adds the appropriate seed data.

Additional Information: AD Administration, *Oracle Applications Maintenance Utilities*

Multiple Reporting Currencies

The Multiple Reporting Currencies (MRC) feature allows you to report and maintain accounting records at the transaction level in more than one functional currency. You do this by defining one or more reporting sets of books, in addition to your primary set of books.

Functional Currencies

In your reporting sets of books, you maintain records in a functional currency other than your primary functional currency.

Primary Functional Currency	The currency you use to record transactions and maintain your accounting data within Oracle Applications. The primary functional currency is generally the currency in which you perform most of your business transactions and the one you use for legal reporting.
Reporting Functional Currency	A currency, other than your primary functional currency, for which you need to report.

Primary Set of Books	A financial reporting entity in which you conduct business. The primary set of books uses a particular chart of accounts, accounting calendar, and functional currency (in the primary set of books, the functional currency is always the primary functional currency).
Reporting Set of Books	A financial reporting entity that is associated with a primary set of books. The reporting set of books has the same chart of accounts and accounting calendar as the primary set of books, but usually has a different functional currency. The reporting set of books allows you to report in a different functional currency than that of your primary set of books.

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

Additional Information: Setup, *Multiple Reporting Currencies in Oracle Applications*

Setting Up the Primary Responsibility

Before you set up MRC, you need to prepare your database, run the AD Administration utility to validate and compile your APPS schema, and install the MRC schema.

Additional Information: AD Administration, *Oracle Applications Maintenance Utilities*

The primary responsibility is linked to the primary set of books, and each reporting responsibility is linked to a corresponding reporting set of books. To create an association between a reporting responsibility and a reporting set of books, you set the profile options GL:Set of Books Name and MRC:Reporting Set of Books to the reporting set of books name at the responsibility level.

You assign the primary responsibility to the Standard data group, and the reporting responsibilities to the MRC data group. The data group determines the tables and privileges accessible by the associated responsibility. Data groups are used for the following purposes:

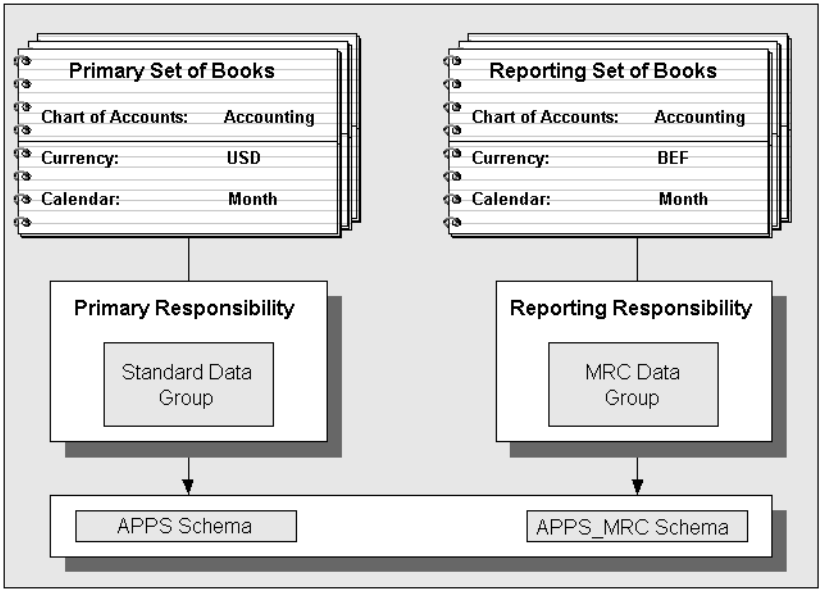
- To identify the Oracle schema used by a responsibility
- To tell the concurrent manager which Oracle schema to use when running a report or concurrent program

In your primary responsibility, the Standard data group connects the application associated with the responsibility to the APPS database schema. In a reporting responsibility, the MRC data group connects the application associated with the responsibility to the MRC database schema.

The default name of the MRC schema is APPS_MRC. If your APPS schema is named something different from the default name of APPS, your MRC schema is named according to the following format:

<name of your APPS schema>_MRC

Figure 5–2 MRC Set of Books, Responsibility, and Database Schema Setup



MRC Architecture

MRC is supported by the following Oracle Applications:

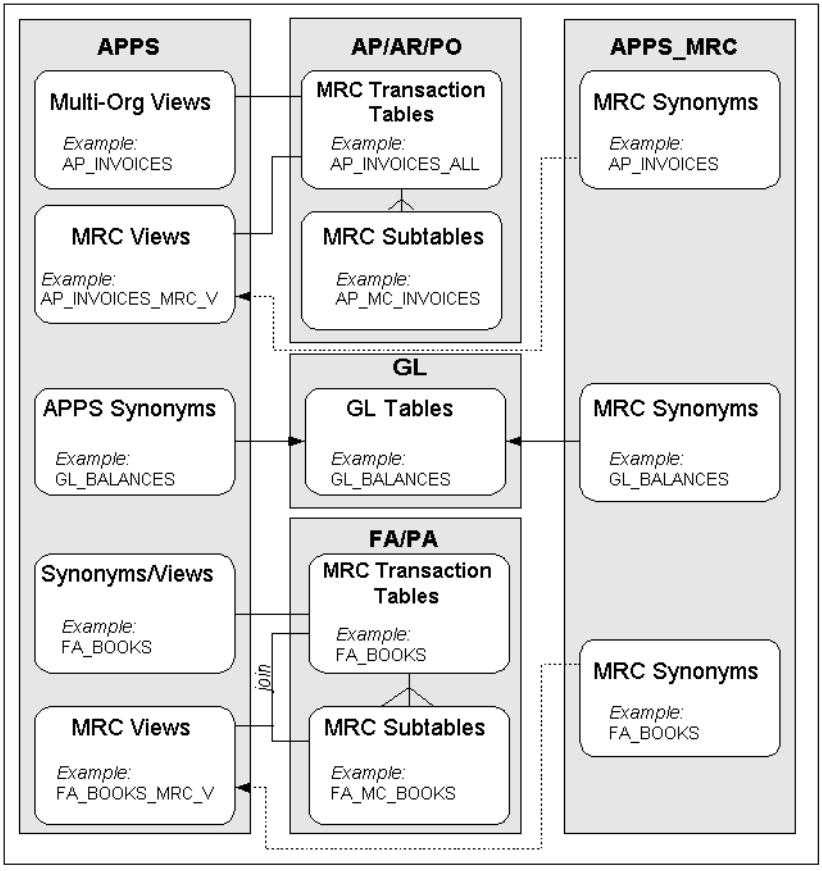
- Oracle Assets
- Oracle Cash Management
- Oracle Cost Management
- Oracle General Ledger
- Oracle Global Accounting Engine
- Oracle Payables
- Oracle Receivables
- Oracle Purchasing
- Oracle Projects

For the subledgers that support MRC, each transaction table in the base schema that has currency-sensitive information, such as conversion rate and transaction amount, now has an associated MRC subtable. Each record in the base transaction table may have one or more MRC subtable records, which contain currency-sensitive information on the base transaction in each associated reporting currency. In the APPS schema, there are MRC views that provide reporting currency representations of each base transaction table.

From the APPS_MRC schema, the MRC synonyms that have names identical to the corresponding APPS Multi-Org views or synonyms point to these MRC views in the APPS schema. When you view a transaction in a reporting responsibility through these MRC synonyms and the MRC views they point to, you see a transaction representation in the reporting currency associated with the corresponding reporting set of books.

Attention: MRC can be used in a database even if Multi-Org Architecture is not implemented.

Figure 5–3 MRC Architecture



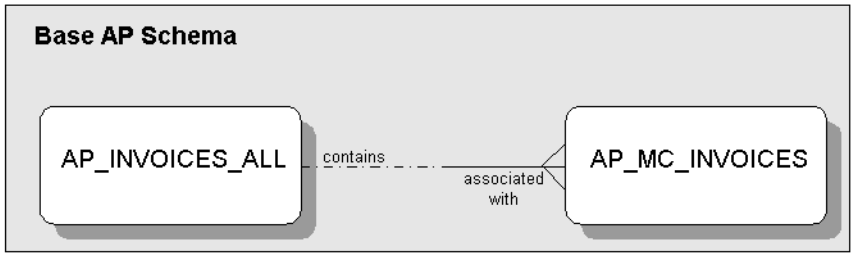
MRC Subtables

For the subledgers that support MRC, each transaction table in the base schema that has currency-sensitive information, such as conversion rate and transaction amount, now has an associated MRC subtable.

There is an optional one-to-many relationship between records in a base transaction table and records in the corresponding MRC subtable. For each transaction record in the base transaction table, there may be one or more corresponding records in the MRC subtable. Each record is associated with a reporting currency. The primary key

for an MRC subtable is the combination of the primary key of the base transaction table and the reporting set of books ID.

Figure 5–4 Transaction Table and MRC Subtable



MRC Triggers

There are MRC database triggers associated with each base transaction table. Whenever you insert, update, or delete records from the base transaction tables, the MRC triggers perform the corresponding database action for the associated records in the MRC subtables.

For example, when you enter an invoice, Oracle Payables inserts records into AP_INVOICES_ALL and AP_INVOICE_DISTRIBUTIONS_ALL. When Payables inserts these records, the database triggers in each of the two base transaction tables execute and create reporting currency records in the MRC subtables AP_MC_INVOICES and AP_MC_INVOICE_DISTS. See *Multiple Reporting Currencies in Oracle Applications* for information about the MRC subtables and their corresponding products.

MRC Views

MRC views provide representations of transactions in each reporting currency. They are created in the APPS schema. MRC views show information for only one set of books at a given time. That is, these views retrieve information based on the reporting set of books associated with the responsibility to which the user is currently logged on. When you sign on to a responsibility, Oracle Applications sets the userenv variable based on the profile option MRC: Reporting Set of Books. This variable value is used by the MRC views to show only the information associated with your reporting set of books.

MRC Synonyms

From the APPS_MRC schema, the MRC synonyms, which have names identical to the corresponding APPS Multi-Org views or synonyms, point to these MRC views in the APPS schema. When you view a transaction in a reporting responsibility through these MRC synonyms and the MRC views they point to, you see the transaction represented in the reporting currency associated with the corresponding reporting set of books.

Release 11*i* uses the Invoker Rights functionality of Oracle9*i* to ensure that most packages are only installed in the APPS schema. The MRC schema, has synonyms to the packages in the APPS schema, and the corresponding packages in the APPS schema have grants to the MRC schema.

Oracle9*i* Server Organization and Requirements

This section discusses how the Oracle9*i* Enterprise Edition database is set up for use with Oracle Applications. It provides information on tablespaces and explains how the AutoUpgrade sizing factor controls the amount of database space reserved for Oracle Applications products.

During the installation, Rapid Install provides the option of distributing tablespaces across different disks. This can reduce disk head contention and improve system response time. However, you normally use disk striping on a production system to increase performance. Refer to your Oracle9*i* Server documentation for complete information about planning a database and tablespaces.

> **Additional Information:** Considerations Before Creating a Database, *Oracle9i Administrator's Guide*

Oracle9*i* Server Tablespaces

The following are recommendations for setting up tablespaces for the Oracle9*i* Server when upgrading to Release 11*i*.

SYSTEM Tablespace

This tablespace holds data dictionary tables owned by the SYS account and is created when you install the database. You may have to increase its size to accommodate the growth of data dictionary tables that occurs when you upgrade Oracle Applications to Release 11*i*.

Rollback Tablespace

The Oracle9*i* Server uses rollback segments to track database changes until they are committed. It uses them during an upgrade of Oracle Applications, as well as during normal, daily use. Verify that rollback segments are large enough to handle upgrade data volume before running AutoUpgrade.

> **Additional Information:** Verify rollback segment sizing, *Upgrading Oracle Applications*

Temporary Tablespaces

The Oracle9*i* Server uses temporary tablespaces to sort data. You can use one tablespace, such as TEMP, as the temporary tablespace for all Oracle Applications products. You can also create up to one temporary tablespace for each product, although this is not necessary. Users access Applications objects through the APPS schema, so the temporary tablespace for that schema (initially the same as that for the Application Object Library) is the tablespace used by each product.

Oracle Applications Tablespaces

For a fresh install of Release 11*i*, Oracle Applications product tablespaces are already created for your system. The following sections provide information on the tablespaces that are set up. There are table, index, and temporary tablespaces used by each Oracle Applications product. These tablespaces use local extent management.

Product Tablespaces

For each Oracle Applications product, there is a tablespace for tables and another one for indexes. Having separate table and index tablespaces for each product makes it easier to license and maintain products later and may increase database performance under certain circumstances.

Oracle Applications tablespaces have a standard naming convention: D for data or X for indexes is added to the product's short name or Oracle schema name. For example, the tablespaces APD and APX are the tablespaces for Oracle Payables tables and indexes, respectively. For performance reasons, these are created as locally managed tablespaces using the temporary file option with a uniform allocation size.

Database Block Size

Release 11*i* requires an Oracle database blocksize of 8K or larger. In addition to providing significant performance improvement, Oracle Applications has some indexes that require a minimum blocksize of 8K.

If you are upgrading to 11*i* and your current system uses a 2K or 4K Oracle blocksize, you must migrate the database to a blocksize of 8K (or larger) before you run AutoUpgrade.

Sizing Factor

A sizing factor sets the extent sizes for dynamic objects such as tables and indexes for an Oracle Applications product. Its value is a percentage of the default estimated growth rate for Applications database objects. You choose a sizing factor for each product when you run AutoUpgrade. The sizing factor affects only the size of next extents, determined by the Oracle9*i* Server with the NEXT object creation parameter. The NEXT extent size is not applied to existing objects, only to newly created objects.

Suggestion: An Oracle Applications consultant can help you determine what sizing factors to choose for your products.

Warning: Do not use a sizing factor of 0%.

100% Sizing Factor (default)

The default sizing factor of 100% creates database objects based on an estimate of how the objects grow at a typical site. Most objects are defined with small first extents and larger additional extents. The NEXT extent sizes vary by object, however.

Objects are created with an initial extent and add extents as necessary. By default, the initial extent is defined as 4 KB for tables and indexes. Oracle uses at least two blocks when creating objects, so if you created your database using a blocksize of 8 KB, for example, the initial storage allocated is at least 16 KB.

The following table shows NEXT extent sizes for two sample tables and two sample indexes created at 100% and 400% sizing factors. These tables and indexes each have an initial extent size of 4 KB.

Table 5–1 Sample Extent Sizes for Sizing Factors

Table/ Index	Table or Index Name	Next Extent (100%)	Next Extent (400%)
Table	AP_AGING_PERIODS	32 KB	128 KB
Index	AP_AGING_PERIODS_U1	32 KB	128 KB
Table	AR_STATEMENT_LINE_CLUSTERS	1 MB	4 MB
Index	AR_STATEMENT_LINE_CLUSTERS_N1	256 KB	1 MB

Your space usage will vary according to how you use your products. Some tables may never grow past their initial extent. Some may grow to several extents.

Glossary

admin server

The server from which the system administrator runs programs that maintain and update an Oracle Applications database. For example, AutoUpgrade and AutoPatch are both run on this machine to install, upgrade, or update the database which resides on the database server.

applet

A Java program that is downloaded to a desktop client from an HTTP server, and run within a Java-enabled web browser.

application servers

Servers that reside in a middle tier, between the desktop clients and database tier. Desktop clients send their requests to application servers, which process the request or send it to another server, such as the database server. The desktop clients never connect directly to the database server. The Forms server and HTTP server are types of application servers. See also *tier.*

applmgr

The login used to install and upgrade Oracle Applications. This account owns the Oracle Applications product files.

APPS schema

An ORACLE schema that has access to the complete Oracle Applications data model.

AutoUpgrade

The Oracle Applications upgrade program.

background process

A noninteractive process that runs in an operating system environment and performs a task.

bandwidth

The amount of data that can be sent through a network connection, 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.*

certificate file

Contains the identity of a "trusted source" that the desktop client uses to guarantee the authenticity of a JAR file. Information contained within the certificate file allows the desktop client to decrypt the digital signature of a JAR file. If the identity can be confirmed, the desktop client assumes the JAR file is safe to download and execute. See also *digital signature*.

character set

A set of encoded binary values that represents the letters, numerals, and punctuation marks of a language, or of a group of languages that use similar written symbols. For example, the WE8ISO8859P1 character set can be used by English and many other languages that use a Latin-based alphabet and Arabic numerals. Terminals and printers handle text data by converting these encoded values to characters. A character set may also be called a *codeset*.

client/server architecture

A configuration in which one or several servers perform database processing or other functions for applications that are run on clients. Software must be installed on each client before the client can interact with the servers. Client / server architecture is not used by the Oracle E-Business Suite in Release 11*i*.

codeset

See *character set*.

command

An instruction or request for the system to perform a particular action. An entire command can consist of the command name, parameters, and qualifiers.

command file

A file containing a predefined sequence of commands to be executed by the operating system.

concurrency

The simultaneous access of the same data by multiple users.

concurrent manager

A process manager that coordinates the processes generated by users' requests to run various data-intensive programs. An Oracle Applications product group can have several concurrent managers.

concurrent process

A task run by a concurrent manager. A concurrent process runs simultaneously with interactive functions and other concurrent processes.

concurrent processing server

An Oracle Applications server that runs time-consuming, noninteractive tasks in the background.

concurrent queue

A list of concurrent requests awaiting completion. Each concurrent manager has its own queue of pending requests.

concurrent request

A request issued to the concurrent processing server when you submit a noninteractive task, such as running a report.

customization

Enhancements to an Oracle Applications system made to fit the needs of a specific user community.

data dictionary

A set of 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—for example, the Oracle9*i* server.

database administrator (DBA)

The person who prepares the Oracle9*i* server and Oracle tools for an installation or upgrade of Oracle Applications, and performs maintenance on them after the installation. The DBA has access to the ORACLE SYSTEM and SYS accounts.

database instance

A running ORACLE system. There is always a one-to-one correspondence between an ORACLE instance and a system global area (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

A machine on which the database holding Oracle Applications data resides. The database server processes SQL and Java requests from other machines, such as Forms servers, HTTP servers, and concurrent processing servers.

database space

The amount of disk space used by a set of database objects.

demonstration product group

A product group that includes predefined transaction data for Oracle Applications products. It is used primarily for system testing and user training. See also *product group*.

dependent product

An Applications product that is not licensed, but whose files are shared in part by a fully installed Applications product. A dependent product is also known as a *shared product*.

desktop client

A computer that sends user requests to the Web server and handles responses such as screen updates, pop-up lists, graphical widgets, and cursor movements.

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.*

distributed concurrent processing

See *parallel concurrent processing*.

distributed directory structure

Applications product files installed in more than one file system, as when there is insufficient disk space in a single file system for all Applications product files.

environment file

A command file that sets environment variables. Only servers running UNIX use environment files—Windows NT servers use the Windows registry instead.

environment setting

A parameter that controls the behavior of Applications and Applications programs for your installation. Environment settings are stored as environment variables on UNIX servers and as registry keys or environment variables on Windows servers.

environment variable

A variable maintained by the UNIX shell that can be referenced by any program running within the shell. Environment variables hold values used by many Oracle programs and utilities.

On Windows, a string consisting of environment information, such as a drive, path, or filename, associated with a symbolic name. You use the System option in the Control Panel or the set command from the Windows command prompt to define environment variables.

Export utility

An Oracle9*i* server utility used to write database data to operating system files external to the database. These files can then be used with the Import utility to read the data back into the same database or into a different one.

extension

The second part, or suffix, of a filename, used to indicate the type or purpose of the file. For example, the extension .sql indicates a SQL*Plus script. See also *filename*.

form

A logical collection of fields, regions, and graphical components that appears on a single screen. Oracle Applications Forms resemble paper forms used to run a business. You enter data by typing information into the form.

Forms client

A Java applet that runs on a desktop client and provides the user interface and interaction with a Forms server.

Forms server

A type of application server that hosts the Forms server engine. It mediates between the desktop client and the database, providing input screens for the Forms-based products on the desktop client and creating or changing database records based on user actions.

functional currency

In Multiple Reporting Currencies, a currency other than your primary currency.

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.

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.

HTML (Hypertext Markup Language)

A simple language used to format documents, predominantly for viewing with a Web browser. Portions of text or images, called hypertext, can be associated with other documents.

HTTP (Hypertext Transfer Protocol)

The TCP/IP-based network protocol used to transmit requests and documents between an HTTP server and a Web browser.

HTTP listener

A program on an HTTP server that accepts and processes incoming HTTP requests from Web browsers.

HTTP server

An application server that runs an HTTP listener, and sends out Web pages in response to HTTP requests from remote browsers. See *HTTP listener, Web browser.*

Import utility

An Oracle9*i* server utility used to read operating system files written by the Export utility. You use it to restore data into a database.

index

A database object associated with a table and used by the Oracle9*i* server to locate rows of that table quickly.

initialization parameters

Parameters defined in an initialization file that configure an Oracle9*i* server database. The parameters affect how the database performs.

Internal concurrent manager

A concurrent manager process that monitors, controls, and dispenses requests to all other concurrent manager processes.

intranet

A network of computers that are internal to a company. A company's intranet is basically a local extension to the globally-distributed Internet, which is a global interconnected network of computers and smaller computer networks.

Java

A computing 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* and *servlet.*

JAR (Java Archive) file

A collection of Java classes compressed into a file for faster download to a desktop client.

Java class

Components of a Java program that define objects and operations performed on objects. Java class also identifies an operating system file that contains a program or part of a program written in Java.

JDBC (Java Database Connectivity)

A Java programming interface that enables Java programs to access the Oracle9*i* server.

JVM (Java Virtual Machine)

An interpreter that translates a compiled Java program, called bytecode, to machine code. JVM makes Java portable, because each operating system's JVM translates bytecode to instructions that the microprocessor can execute. 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 WAN. See also *WAN*.

latency

In networking, the amount of time it takes a packet of data to travel from a source to its destination. The speed and capacity of a network depend on both bandwidth and latency. See also *bandwidth*.

load balancing

Distributing tasks to the server that is least busy when several servers are handling the same workload. Using load balancing, the HTTP server connects the Forms client to the Forms server that has the lightest load. This server is called the "least loaded host."

LOCAL

Under Windows, an environment setting that identifies the network alias of an ORACLE instance running on the local machine or on another networked machine. This variable overrides any setting for ORACLE_SID and causes the Net8 software to manage the connection request. See also *ORACLE_SID* and *TWO_TASK*.

log in

To perform a sequence of actions that establishes communication with the operating system or a secured program, such as the Oracle9*i* Server or Oracle Applications, and sets up default characteristics for the session.

Megabyte (MB)

A unit of memory or disk space equal to 1,048,576 bytes (1024 x 1024). Often rounded to one million bytes.

Multiple Organization Architecture (Multi-Org)

A single installation of any Oracle Applications product to support any number of organizations or different sets of books. The data contained in product schemas is for all organizations, and is partitioned by the ORG_ID column in tables.

Multiple Reporting Currency (MRC)

An Oracle Applications feature that allows you to create, maintain, and report on accounting records at the transaction level in more than one functional currency.

Net8

The Oracle product that enables network connectivity between a client machine and the Oracle9*i* server. Net8 manages communication sessions between these machines by opening and closing sessions and by packaging and sending SQL statements and data responses.

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. NLS involves operation in only one language per installation.

node

A stand-alone machine or a machine connected to the network.

operating system

The computer software that performs basic tasks such as allocating memory and allowing computer components to communicate.

ORACLE

An Oracle9*i* server database. This generally refers to a database and the objects it contains, not to the Oracle9*i* server product files.

ORACLE_HOME

An environment setting that specifies the top directory for Oracle9*i* server program files.

ORACLE schema

See *schema*.

Oracle9*i* server

The database management system used by Release 11*i*. The term refers in general to the product files or the ORACLE instances created using these files.

ORACLE_SID

An environment setting that identifies an ORACLE instance running on the current machine. See also *TWO_TASK* and *LOCAL*.

ORACLE user ID

A username and password used to access an ORACLE instance.

parallel concurrent processing

In a UNIX environment, distribution of concurrent processes among multiple concurrent processing servers. Also called *distributed concurrent processing*.

password

An identification word, associated with your username, that you must supply to access an ORACLE instance or an Oracle Applications system.

platform

The underlying structure of a computer system, including hardware and software, on which application programs run. The hardware component includes the microprocessor, which is the microchip that performs logic operations and data management. The software component includes the operating system, which is the computer system's coordinating program. Sun SPARC Solaris is an example of a platform. Some Oracle Applications functionality is platform-specific, meaning its behavior may differ on other platforms.

PL/SQL

A procedural extension of SQL that provides programming constructs such as blocks, conditionals, and functions.

primary functional currency

The currency you use to record transactions and maintain your accounting data within Oracle Applications.

primary set of books

A financial reporting entity in which you conduct business.

Pro*C/C++

An Oracle precompiler product that allows developers to embed standard database calls to an ORACLE database in C and C++ programs.

product group

A set of Oracle Applications product schemas linked together by a single Oracle Application Object Library schema.

registry

A Windows database that holds configuration information. During installation, Oracle Applications writes data to the registry. You can also edit the registry directly with the Registry Editor (regedt32.exe).

registry key

A folder that appears in the left pane of the Registry Editor window. A key can contain subkeys and value entries. For example: Environment is a key of HKEY_ CURRENT_USER. See also *subkey*.

registry subkey

A key within a key. Subkeys are analogous to subdirectories in the registry hierarchy. Keys and subkeys are similar to the section heading in .ini files; however subkeys can carry out functions. Oracle Applications stores important information about a product group in a registry subkey. See also *registry key* and *registry*.

report

An organized display of Oracle Applications information. A report can be viewed online or sent to a printer. The content of a report can range from a summary to a complete listing of values.

Report Review Agent

A tool used by Oracle Applications to view concurrent processing files online.

reporting functional currency

A currency, other than your primary functional currency, for which you need to generate reports.

reporting set of books

A financial reporting entity that is associated with a primary set of books. It has the same chart of accounts and accounting calendar, but usually a different functional currency.

reserved word

A word that has a special meaning to any computer program. Custom-built programs that integrate with Oracle Applications must not use reserved words.

responsibility

A collection of functions within an Oracle Application. Each Applications user is assigned one or more responsibilities to allow them access to Applications Forms.

rollback segment

A set of entries used to undo changes in the database in the event of transaction rollback, crash, or media recovery.

rollback tablespace

A tablespace created for rollback segments.

RPC (Remote Procedure Call)

A protocol that allows 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.

schema

An ORACLE account or ORACLE ID.

servlet

A Java program executed on an HTTP server, rather than downloaded to a desktop client. See also *applet*.

setup data

Company-specific configuration data, such as locations, freight terms, and payment terms. You create this data when initially configuring an Oracle Applications product.

SGA (System Global Area)

A reserved section of main memory that provides communication between all database users and the ORACLE background processes.

shared product

See *dependent product*.

short name

An abbreviation for an Oracle Applications product (such as *gl* for Oracle General Ledger).

sizing factor

An integer that determines the growth rate, as a percentage of their defaults, for the database objects of an Oracle Applications product.

SQL (Structured Query Language)

An internationally standard language used to access data in a relational database. The acronym is pronounced “sequel.”

SQL script

A file containing SQL statements that you run with a tool such as SQL*Plus to query or update ORACLE data.

subdirectory

A directory that is contained within another directory.

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 SQL statements for database users.

syntax

The orderly system by which commands, qualifiers, and parameters are arranged together to form valid command strings.

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

A necessary table in any relational database. SYS.DUAL contains exactly one row, and is used as a "dummy" table in a SQL statement to return values that are not stored in tables, such as constant values, evaluations of arithmetic expressions, or system values like the current date.

system administrator

The person who manages administrative tasks in Oracle Applications, such as registering new users and defining system printers, using the system administrator responsibility.

SYSTEM 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 kinds of values (columns).

tablespace

A logical portion of an ORACLE database used to allocate storage for data and to group related logical structures. For example, one tablespace may contain all of one Oracle Applications product's database tables and indexes.

TCP/IP (Transmission Control Protocol / Internet Protocol)

A widely-used industry-standard networking protocol used for communication among computers.

temporary tablespace

A tablespace used when a SQL statement requires the creation of temporary segments (for example, the creation of an index).

TWO_TASK

Under UNIX, an environment setting that identifies the network alias of an ORACLE instance running on the local machine or on another networked machine. This variable overrides any setting for ORACLE_SID and causes the Net8 software to manage the connection request. See also *ORACLE_SID* and *LOCAL*.

user ID

The combination of a username and its password.

username

A name that grants access to a secure environment or program, such as an ORACLE database or Oracle Applications. A username is customarily associated with a collection of privileges and data available to a particular user (*responsibilities* in Oracle Applications). Every username is associated with a password.

view

A custom-tailored presentation of the 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 running on a desktop client that views documents formatted in HTML and runs Java applets. A Web browser sends requests to a HTTP server using a special protocol (HTTP) to retrieve documents and Java applets. See *HTTP listener, HTTP server.*

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

P

R

S

T

U

W