

Oracle® Applications

Concepts

Release 11 for UNIX

April 1998

Part No. A63418-01

Oracle Applications Concepts, Release 11 for UNIX

Part No. A63418-01

Release 11

Copyright © 1998, Oracle Corporation. All rights reserved.

Primary Author: Michael Fiore

Contributors: Janet Buchbinder, Gail D'aloisio, Carole Eubanks, Rick Lotero, Jeff Lunn, Kent Noble, Keith M. Swartz

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

This Program contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright, patent and other intellectual property law. Reverse engineering of the software is prohibited.

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

If this Program is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication and disclosure of the Programs shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-14, Rights in Data -- General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

"Restricted Use Programs" are limited to use solely in conjunction with and in support of the licensed Oracle Applications Programs. Customers may not use the Restricted Use Programs for general data processing purposes or for any purpose other than specified in the previous sentence. The following Programs are included as Restricted User programs with Oracle Applications Programs: Oracle8 Enterprise Edition, SQL*Plus, Programmer/2000, and Oracle Application Object Library.

Oracle is a registered trademark, and Context, Developer/2000, EasiPath Migration Method, EMM Advantage, Net8, Oracle8, Oracle Alert, Oracle Application Object Library, Oracle Context, Oracle Financials, Oracle InterOffice, Oracle Office, Oracle Quality, Oracle Web Customers, Oracle Web Employees, Oracle Work in Process, Oracle Workflow, PL/SQL, SmartClient, SQL, SQL*Forms, SQL*Loader, SQL*Menu, SQL*Net, SQL*Plus, SQL*Report, and Web Agent are trademarks or registered trademarks of Oracle Corporation.

All other company or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

Contents

Preface	vii
Audience	vii
Structure	viii
Related Documents	viii
Getting Help	x
Oracle Consulting Services	x
Oracle Support Services.....	x
Send Us Your Comments	xiii
1 Oracle Applications Release 11	
Release 11 Overview	1-1
Network Computing Architecture	1-1
Desktop Client Tier	1-2
Application Tier	1-4
Release 11 Database Tier	1-7
Network Traffic and Optimization	1-11
Forms Server and Forms Client Applet.....	1-11
Performance Considerations.....	1-12
New Functionality	1-13
Multiple Reporting Currencies.....	1-13
Attachments	1-14
Spreadsheet Export	1-14
Context-sensitive Help.....	1-15

Report Review	1-15
2 System Components	
Release 11 Products and Product Families	2-1
Product Families	2-3
System Issues.....	2-5
Oracle Applications Language Translations.....	2-8
Underlying System Components.....	2-9
UNIX Operating System.....	2-9
Oracle8 Server	2-9
Oracle Tools.....	2-10
3 Data Model	
Basic Data Model	3-1
Product Groups.....	3-1
ORACLE User IDs.....	3-2
APPS Schema.....	3-3
How AutoInstall Creates APPS Schemas.....	3-5
Localizations Objects.....	3-6
Custom Schema Access.....	3-6
Data Access.....	3-6
Multiple Organization (Multi-Org) Architecture.....	3-7
Overview.....	3-7
Partitioned Objects	3-8
Changes from Release 10.6.....	3-9
Multiple Reporting Currencies.....	3-9
Functional Currencies	3-9
Setting Up the Primary Responsibility	3-10
MRC Architecture.....	3-11
Multiple Sets of Books.....	3-16
Multiple Product Installation Groups and Multiple Sets of Books	3-18
Oracle8 Server Organization and Requirements.....	3-19
Database Creation.....	3-20
Oracle8 Server Tablespaces	3-21
Oracle Applications Tablespaces.....	3-22

Sizing Factor	3-23
---------------------	------

4 File System

Directory Structure	4-1
Basic Directory Structure.....	4-1
Distributed Directory Structure.....	4-5
Localization Directories.....	4-5
Language Directory Structure	4-6
Multiple Applications Top Directories.....	4-7
Product Files for Multiple Product Groups.....	4-8
Log and Output Files	4-9
Log and Output Files for a Single Product Group.....	4-10
Log and Output Files for Multiple Product Groups	4-12
Selecting a Log and Output File Storage Method.....	4-15
Temporary Files	4-15

Index

Preface

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.

Audience

Oracle Applications Concepts explains the technology, architecture, and terminology used in *Oracle Applications Installation*. You should read *Concepts* along with *Installation* if you are responsible for installing or upgrading Oracle Applications. *Installation* provides instructions for installing Oracle Applications products and explains the utilities you use for installation and maintenance. These people typically use *Concepts* and *Installation*:

- Technical Specialist
Creates the accounts and directories used by Oracle Applications and ensures that the system has adequate memory and disk space resources.
- Database Administrator
Prepares the Oracle Server and Oracle tools for an installation or upgrade of Oracle Applications. Has access to the ORACLE SYSTEM and SYS accounts.
- System Administrator
After the installation, this person assumes responsibility for tasks such as registering new users and defining system printers.

Structure

Oracle Applications Concepts contains these chapters:

Chapter 1	Explains the new features of Oracle Applications Release 11, including Network Computing Architecture and Multiple Reporting Currencies.
Chapter 2	Discusses the components of Release 11, including the Release 11 product family, the UNIX operating system, and the Oracle Server database.
Chapter 3	Covers the Oracle Applications data model, including the APPS schema and Multiple Organization architecture.
Chapter 4	Describes the Release 11 file system, including multiple product groups, the product directory structure, localizations, and setting up directories for log and output files.
Index	An alphabetical listing of references to topics in this manual.

Related Documents

All the documentation related to Release 11 is included on the *Oracle Applications Documentation Library* CD, which is supplied with Release 11. You can purchase additional sets of printed and online documentation for Oracle software products from the Oracle Documentation Sales department. Type the following URL: (<http://www.oracle.com/documentation/sales/index.html>).

Attention: With Release 11 Network Computing Architecture, you may choose to install Oracle Applications software on a combination of operating systems. For example, the forms server may be on a Windows NT machine, but the administration server or database may be on a UNIX machine. In this case, you would follow the instructions for installing and implementing the forms server from the Windows NT server edition of the *Installation* manual, and the instructions for installing and implementing the Release 11 database server from the UNIX server edition of the *Installation* manual.

Chapter 1 in this manual explains the different types of servers and their roles in Release 11, and possible ways to deploy this release in a mixed-platform, multi-tier topology.

In addition to *Concepts and Installation*, we suggest the following references:

- *Oracle Applications Installation Release Notes*

The *Release Notes* provide up-to-date information on this release and its components. Material in the *Release Notes* may complement or supersede the information in this manual.
- *Oracle Applications Installation Update* for your platform

Each update document, such as the *Oracle Applications Installation Update for Sun SPARC Solaris*, provides installation information specific to your platform. It describes installation procedures on your platform that differ from *Oracle Applications Installation*.
- *Oracle Applications Upgrade Manual*

A successful upgrade requires you to prepare your existing Oracle Applications products as described in the *Upgrade Manual*.
- *Oracle Applications Product Update Notes*

Provides a history of the changes to Oracle Applications since Release 10 SmartClient Production 1. When upgrading, use the *Product Update Notes* to take advantage of new features in Release 11.
- Oracle Applications Technical Reference manuals

List information about database structures and help determine what changes you need to make to customizations after an upgrade.
- *Oracle Applications System Administrator's Guide*

Provides additional information you need to administer the Oracle Applications database server.
- *Oracle Applications User's Guide*

The *User's Guide* provides an overview of Oracle Applications, including basic concepts, terminology, and navigation.
- *Oracle Financials Country-Specific Installation Supplement*

Contains country-specific information about topics such as responsibilities and report security groups and provides any post-install steps that may be required by a country or by the Global Accounting Engine.

Getting Help

Oracle Consulting Services and Oracle Support Services are the main sources of help for installing Oracle Applications.

Oracle Consulting Services

Oracle Consulting Services can help:

- determine machine size and database size required by Oracle Applications
- install or upgrade Oracle Applications
- implement Oracle Applications products
- customize Oracle Applications products
- develop custom applications for use with Oracle Applications
- train users of Oracle Applications

Oracle Support Services

Have this information ready when you contact Oracle Support Services:

- your CSI number
- the operating system and versions of all Oracle Applications servers
- the release of Oracle Applications you are installing and the versions of the Oracle Server and Oracle tools you are using
- the release of Oracle Applications you are upgrading from
- a description of the problem as well as specific information about any error messages you received
- whether you have dial-in capability
- the number and status of the AutoInstall parallel workers
 - Additional Information:** *Monitoring AutoInstall, Oracle Applications Installation*
- the output of the AD Configuration Utility, contained in the adutconf.lst file
 - Additional Information:** *The AD Configuration Utility, Oracle Applications Installation*

Conventions

We use the following typographical conventions in this manual.

Convention	Meaning
Monospace text	Command line text. Type this text exactly as shown.
< >	Text enclosed in angle brackets represents a variable. Remove the angle brackets and substitute an appropriate value for the variable.
[]	Brackets enclose optional items. Do not type the brackets.
	A vertical bar represents an “or” option among several options. You must enter only one of the options. Do not type the vertical bar.
/directory	A slash before a directory name indicates that it is a subdirectory. The path name is case-sensitive.
\$	Represents the platform command prompt (your prompt may differ). Also represents the value of operating system variables.
\	In command examples, a backslash at the end of a line means you must type the entire command on one line. Do not type the backslash.

The following special notes alert you about particular information:

Additional Information:	Refers you to portions of this manual, another manual, or the online documentation.
Attention:	Alerts you to important information that will help you use the system.
Note:	Highlights helpful hints and practical tips that can save time and make installation or other procedures easier.
Warning:	Warns you about actions which, if not carried out properly, could be damaging or destructive to your operations.

Send Us Your Comments

Oracle Applications Concepts, Release 11 for UNIX

Part No. A63418-01

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 other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- FAX - 650.506.7369 Attn: Oracle Applications Release Group
- postal service:
Oracle Corporation
Oracle Applications Release Group
500 Oracle Parkway, M/S 3op4
Redwood Shores, CA 94065 U.S.A.

If you would like a reply, please give your name, address, and telephone number.

Oracle Applications Release 11

This chapter describes the new architectural features in Oracle Applications. Refer to the *Product Update Notes* for information about each product's new features. This chapter includes these sections:

- Release 11 Overview
- Network Computing Architecture
- Network Traffic and Optimization
- New Functionality

Release 11 Overview

Oracle Applications Release 11 is Oracle's complete enterprise suite of Financial, Human Resources, Manufacturing, Supply Chain, and Sales Force Automation software, comprising more than 50 different applications. Because Release 11 uses a Java-based interface, corporations can implement enterprise applications that have all of the benefits of graphical client/server software without installing Oracle Applications client software on each user's desktop. Based on Network Computing Architecture (NCA) designed by Oracle, Oracle Applications can be deployed on any PC, network computer, or other Java-enabled desktop.

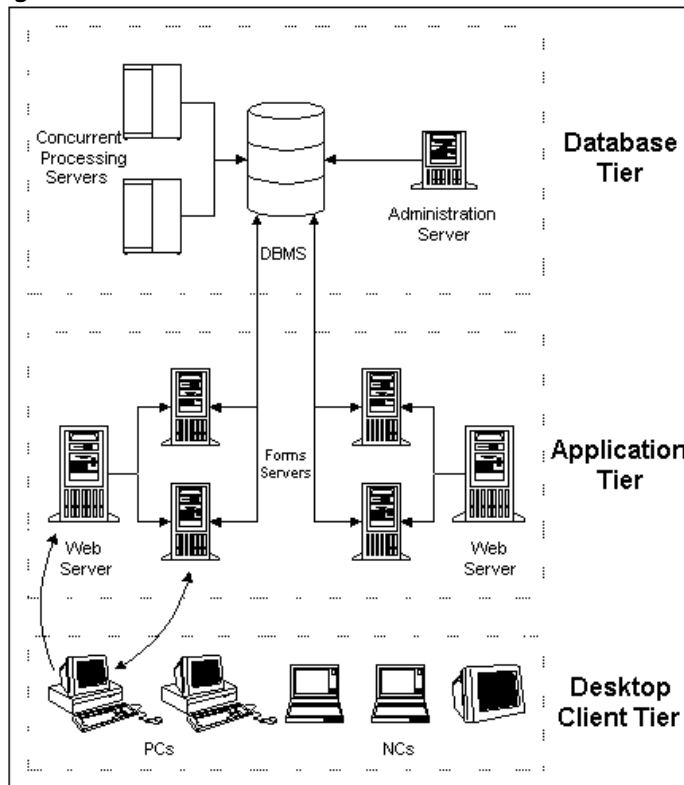
Network Computing Architecture

Network Computing Architecture updates the Release 10 SmartClient model, where application software was installed and maintained on the desktop as well as on servers. In Release 11, the client interface is provided through a Java applet. The client can download the applet on demand, or cache the applet locally and download only when the applet is updated. All updates are installed on the server

and are downloaded to the client automatically from the server. Shifting software administration from the desktop to the network in Release 11 provides you with both a graphical user interface and lower administration costs.

Network Computing Architecture provides Oracle Applications with a framework for *multi-tier*, distributed computing. In the multi-tier computing model, various functions of the Oracle Applications architecture are distributed among multiple levels, or *tiers*, of machines. Although many physical machines may be used in a configuration, scalability derives from processing capabilities on three separate tiers: the desktop client tier, the application tier, and the Release 11 database tier.

Figure 1-1 Overview of Mutli-tier Architecture



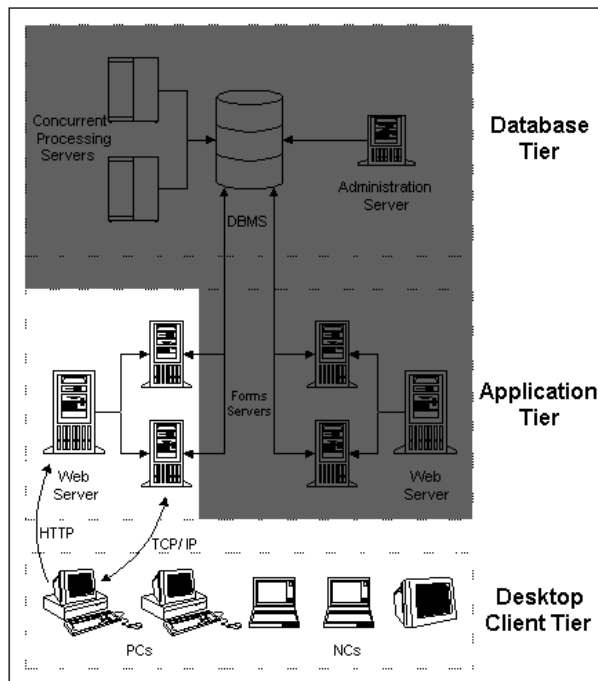
Desktop Client Tier

The *desktop client* runs a Java applet using a Java-enabled web browser or appletviewer. The applet sends user requests to the forms server and handles such

responses as screen updates, pop-up lists, graphical widgets, and cursor movement. The desktop client's Java applet is another component of Oracle's Developer/2000 application development tool called the *Forms client applet*. It can display any Oracle Applications screen and supports field-level validation, multiple coordinated windows, and data entry aids like lists of values.

A *web browser* or appletviewer manages the downloading and storage of the Forms client applet on each user's desktop. They also supply the Java Virtual Machine (JVM) that runs the Forms client applet. The following figure illustrates the communication between the desktop client tier and the other tiers.

Figure 1-2 How the Desktop Client Tier Communicates



JAR File and Security

The Forms client applet is a general-purpose applet that works with all Oracle Applications, including those with customizations and extensions. To achieve rapid startup and keep network traffic light, the Forms client is packaged as a single Java Archive (JAR) file that contains all the Java classes typically required. The JAR file is downloaded once, at the beginning of the client's first session. Afterwards, it

remains in the browser's local disk cache, ready for future sessions, until an updated version is released.

The JAR file for the Forms client applet uses a *digital signature* to guarantee its authenticity to the desktop client. The digital signature is an encrypted message containing the identity of the JAR file's author—in this case, Oracle Applications.

To validate the digital signature, each desktop client holds a *certificate file*. The certificate contains the identity of a "trusted source", and a key that can be used to decrypt messages sent by that source, such as its digital signature. When a web browser or appletviewer submits a request to download the JAR file containing the Forms client applet, the JAR file presents its credentials (the digital signature). The web browser or appletviewer decrypts the digital signature and compares the result with what was stored in the certificate. If they match, the signature is considered to be authentic, and the JAR file can be downloaded.

This form of security ensures that the forms server only accepts connections from "certified" Forms clients bearing the Oracle Applications signature. For additional security, all communication between the Forms client applet and forms server is encrypted using the RSA RC4 40-bit standard form of encryption.

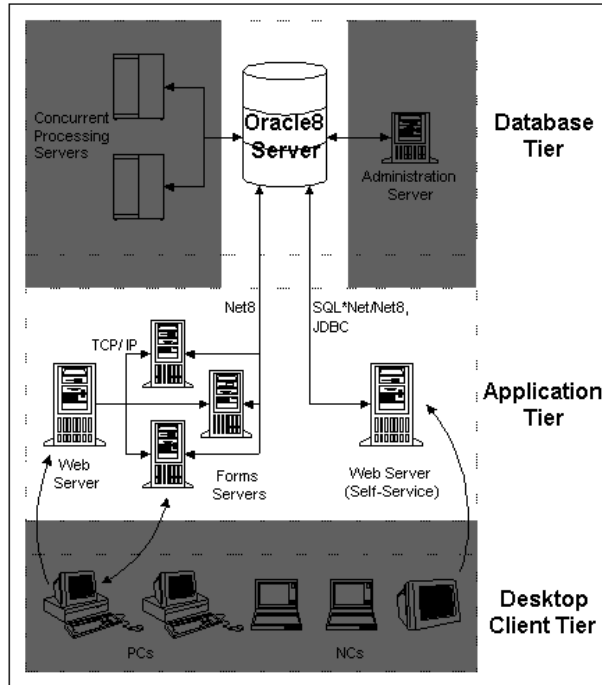
Application Tier

The *application servers* form the middle tier between the desktop clients and Release 11 database servers. They provide load balancing, business logic, and other functionality. In installations that use multiple application servers, only one needs to run the Oracle Web Application Server software.

Note: The Release 10 database server was often called the Applications server. In Release 11, the database server is always called the Release 11 database server. *Application server* now refers to the tier that runs the Oracle Web Application Server and Oracle Forms Server software. Please note the change in terminology.

The following figure illustrates the communication between the application tier and the other tiers.

Figure 1-3 How the Application Tier Communicates



Forms Server

The *forms server* is a specific type of application server that hosts the Oracle Forms Server engine. The Oracle Forms Server is a Developer/2000 component that mediates between the desktop client and the Oracle8 Server, displaying client screens and causing changes in the database records based on user actions. Data is cached on the forms server and provided to the client as needed, such as when scrolling through multiple order lines. The forms server exchanges messages with the desktop client across a standard TCP/IP network connection.

Load Balancing Among Forms Servers Release 11 provides automatic load balancing among multiple application servers. In a load-balancing configuration, a single coordinator called the *Metrics Server* is on one application server. Metrics Clients located on the other application servers periodically send load information to the Metrics Server so it can determine which has the lightest load. When a client issues a request to download the Forms client applet, the Metrics Server provides the name of the *least-loaded host* for the applet to connect to. This process is made possible through mediation by the web server and Oracle Web Application Server.

Automatic failover capabilities are inherent in this load-balancing system. If an application server becomes unavailable for any reason, the Metrics Server ceases to route requests to the server until it comes back online. While the application server is offline, requests are routed to one of the other application servers.

Additional Information: Setting up Load Balancing, *Oracle Developer/2000 Release 1.6: Deploying Applications on the Web*; Chapter 4, *Oracle Applications Installation*

Web Server

The *web server* is another type of application server, which runs an *HTTP listener*. The HTTP listener (also called a web listener) is a component of an HTTP server, such as Microsoft Internet Information Server, or Netscape Enterprise Server. This listener accepts incoming HTTP requests (or URLs) from desktop clients, via the web browser or appletviewer. These requests are either immediately processed—for example, by returning an HTML document—or are passed on to the Oracle Web Application Server, which also resides on this machine.

The Oracle Web Application Server handles these requests by redirecting them to a *cartridge*. You configure the Web Application Server to indicate what types of requests should go to various cartridges that are installed. For example, if you set up the Oracle Forms Load Balancing Cartridge, this cartridge examines the HTML file requested by the desktop client and provides information like the name of the least-loaded host and the username and password for connecting to the Applications database. The results are then delivered to the desktop client via the HTTP listener.

To access Oracle Applications, you start up a web browser or appletviewer and submit a request to the web server for the initial signon HTML page by entering the URL for the page. The page may be processed by the Load Balancing cartridge, if it has been configured, then is returned to you, along with the Java applet used for accessing Oracle Applications forms. The Forms client applet is invoked with parameters that include the name of the forms server to which the client applet connects. The applet connects to this forms server using a standard TCP/IP connection, and from this point on, you are connected directly to the forms server until you choose to exit Applications.

The applet parameters also contain information about the Applications database to which the forms server should connect. This makes it easy to set up access to multiple Oracle Applications environments, and to connect to these environments simultaneously. For example, you can create two instances of the Load Balancing

Cartridge, and configure one to connect you to a production database, while another connects you to a test database. The Web Application Server examines the URL to determine which cartridge instance should process the request. The cartridge then provides the information on which database instance to connect to, and this data is relayed by the HTTP listener back to the desktop client.

Web Server and Oracle Self-Service Web Applications

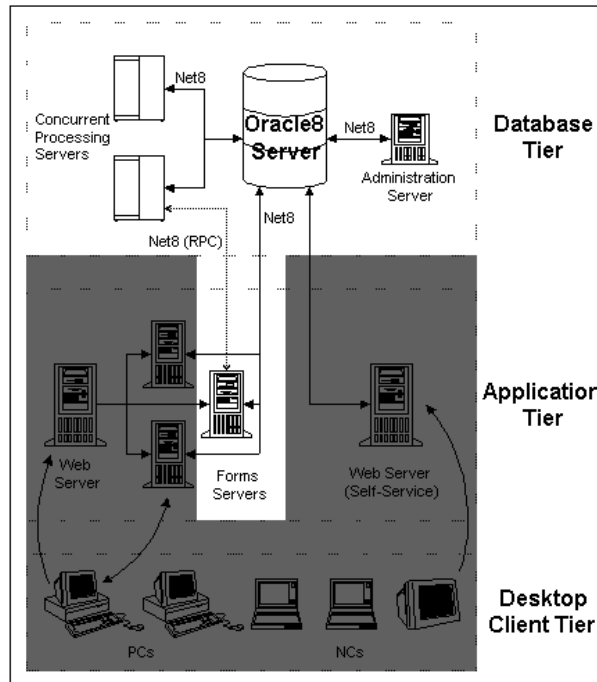
The Oracle Self-Service Web Applications are unique in that they do not use Oracle Forms for the interface. The Self-Service Applications are designed in pure HTML and JavaScript, and operate by direct connection to an HTTP listener running Oracle Web Application Server. Logic is controlled through stored procedures that execute through the Web Application Server's PL/SQL Cartridge, and Java *servlets*—Java programs executed by the Web Application Server—that communicate with the database using JDBC.

The web server used for Self-Service Applications can be the same as the one used for accessing other Oracle Applications products through Oracle Forms. However, Self-Service Applications perform load balancing using features in Oracle Web Application Server, and do not use the Oracle Forms Load Balancing Cartridge.

Release 11 Database Tier

The database tier holds all data and data-intensive programs, and processes all SQL requests for data. The database tier includes the Oracle8 Server, the administration server, and the concurrent processing server. By definition, machines in this tier do not communicate directly with Applications users, but rather with machines on the application tier that mediate these communications, or with other servers on the Release 11 database tier.

The following figure illustrates the communication between the Release 11 database tier and the other tiers.

Figure 1–4 How the Database Tier Communicates

Database Server

The Release 11 database server contains the data associated with Oracle Applications. This machine does not contain any Oracle Applications *files*, only the Oracle8 Server files, and an Applications database instance that physically stores the tables, indexes, and other database objects for your installation.

The installation process for Oracle Applications begins here, by installing and creating a database and configuring Net8 to accept connections from other servers. Once you have created your database, you administer the Applications data from the administration server.

Administration Server

The administration server is the machine from which you maintain the data in your Applications database. There are three types of operations you will carry out here, each using a different program:

- installing and upgrading the database
This process is only done when you are installing a new release, or are upgrading to a new minor or major release. You use the AutoInstall program to do this.
- applying Applications database updates
Most bug fixes will consist of new files and scripts that update database objects. You use the AutoPatch program to apply the updates to the database objects from the administration server. (AutoPatch is also used on other servers to update the file systems on those machines.)
- maintaining the Applications data
Some features, such as MultiLingual Support and Multiple Reporting Currencies, require regular maintenance to ensure updates are propagated to the additional schemas used by these features. The AD Administration program (adadmin) allows you to do this. (You can also administer files on other tiers using this program.)

These administration tasks can only be performed when users are not accessing the system, and none of the operations listed above provide interaction with Applications users. This means that the administration server is the most infrequently used, compared to other servers in the Applications multi-tier environment, and has the smallest computing requirements. You therefore should not need to have more than one administration server for your installation.

Additional Information: Chapter 5, *Oracle Applications Installation*

Concurrent Processing Server

Most interaction with Applications data is done through Applications forms. There are also reporting programs, however, that periodically need to be run. These programs may contain a very large number of computations, so to ensure that they do not interfere with interactive operations, they can be configured to run on a separate machine called the *concurrent processing server*.

Concurrent Processing

Processes that run on the concurrent processing server are called *concurrent programs*, and operate in the background while you and other users continue to work on other tasks. These programs are typically executables written in C or reports written using Oracle Reports. You submit a request to run concurrent programs through Applications forms, which insert the request into a database table. When the table is read by a monitoring process, the request is assigned to one

of several *concurrent managers* (also referred to as *workers*) running on the concurrent processing server. The concurrent manager processes the request (which may involve calling another program, for example, or running Oracle Reports), and generates log and output files, which are stored on the concurrent processing server. The *Internal concurrent manager* process 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.

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

Parallel Concurrent Processing

Parallel concurrent processing allows you to run concurrent managers on multiple servers in your database tier. This distributes the processing load and provides fault tolerance in case one or more servers fail.

The processing load for concurrent processing is distributed across all servers in a parallel concurrent processing system. This system consists of multiple concurrent processing servers in a network, cluster, or massively parallel environment. Each server performs the processing for one or more concurrent managers. In addition, some operating systems have native mechanisms for queuing batch processes or distributing process loads. Parallel concurrent processing is designed to integrate with these mechanisms.

Additional Information: *Oracle Applications Installation Update* for your platform

Fault tolerance is achieved by assigning every concurrent manager a *primary server*, which determines where the process will run under normal conditions, and a *secondary server*, which determines where it will run if the primary server fails. If the primary server for a concurrent manager fails, the Internal concurrent manager restarts the manager on its secondary server. Additionally, *Internal monitors* ensure the Internal concurrent manager is always available. If the primary server for the Internal concurrent manager fails, an Internal monitor can restart it on the local server, until the primary server is available again. This provides multiple fault tolerance across your entire parallel concurrent processing system.

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

Interaction with Forms Servers

Most of the time, the concurrent processing servers only communicate directly with the Release 11 database server. For example, all concurrent requests are entered and controlled through Applications forms, which update information in database tables. These tables are then polled on a regular basis by the Internal concurrent manager process.

Occasionally, however, the forms servers may need to communicate directly with the concurrent processing servers. This may be necessary to retrieve reports for on-line viewing, or to run a program that returns its results to the Applications user interactively, rather than in the background. The forms server achieves this by using Net8 to issue a Remote Procedure Call (RPC) to the concurrent processing server. The concurrent processing server runs a Net8 Listener, which responds to the request by invoking an Oracle Applications executable, then returning the results via Net8.

Although these connections with the forms server are much less frequent than communications with the Release 11 database server, network latency between the concurrent processing server and forms server may result in Applications data being returned more slowly to Applications users. This may, therefore, have an impact on how you decide to distribute and configure the servers on your database tier with respect to those on the application tier.

Additional Information: Setting Up Report Review Agent, *Oracle Applications Installation*

Network Traffic and Optimization

Oracle Applications Release 11 operates efficiently over a Wide Area Network (WAN) or dial-up connection. Only one network round-trip is needed for field navigation, and only a small number of round-trips are needed to change forms.

Forms Server and Forms Client Applet

The Forms client applet handles all screen and local data management directly, without contacting the forms server. Such operations as scrolling through records, moving between windows, or typing characters in a field produce no network traffic at all. In contrast to block-mode clients, the Forms client applet also knows what user interface operations can benefit from immediate contact with the forms server—for example, entering data in a validated field.

When an end user tabs out of a validated field or performs some other action that requires an immediate response, the Forms client applet sends a single message to

the forms server that describes the event in shorthand form—for example, "John Proctor entered in Name field" or "list of values button clicked." The forms server replies with a single message that conveys all of the changes that need to be made in the user interface to convey a full and complete response to this event, again in shorthand form. Depending on the situation, the response might consist of changing values in other fields displayed on screen, popping up a dialog box that presents a choice of valid entries for the field, making new UI elements appear and old ones disappear, or relaying an error message (or other advice) that requires immediate attention. This approach minimizes the number of network round trips and reduces bandwidth consumption.

Performance Considerations

There are several ways you can improve the performance of the Oracle Applications, depending on your networking environment and client/server distribution.

- Minimize network latency between the forms servers and Release 11 database server.

The majority of network exchange occurs between these two tiers. We strongly recommend that these components of your network topology have the lowest-possible latency and highest-possible bandwidth between them. For example, all forms servers and the Release 11 database server should be installed on the same subnet, and should communicate with one another using a high-speed network connection.

- Minimize network latency between concurrent processing servers and Release 11 database server.

Concurrent programs are typically very data-intensive, and can generate heavy network traffic to and from the Release 11 database server. Because these jobs are intended to run as background tasks, this traffic is not optimized for operation over wide-area networks. However, depending on the volume and frequency of requests you handle in your environment, you may be able to copy with minor amounts of network latency here, as it will not be perceived by Applications users as an interruption or delay in their interactive operation. This does not apply to traffic between the forms server and Release 11 database server, as indicated previously.

- Use load balancing to evenly distribute connections among multiple forms servers.

Developer/2000 provides a load balancing feature that allows you to dynamically balance load among multiple systems running Oracle Forms Servers.

Additional Information: Setting Up Load Balancing, *Oracle Developer/2000 Release 1.6: Deploying Applications on the Web*.

- Download Java archive files from a local file server when connecting over slow or wide-area networks.

Network traffic requirements between desktop clients and the forms servers are considerably smaller, since only changes in the graphical user interface are transmitted once the Forms Java applet is downloaded at startup. This initial startup time can be improved when connecting over slow or wide-area networks by having the desktop clients download the Java archive from a local file server rather than the application server. This avoids having each user download the applet over the WAN when a new version is released. See Edit the Oracle Applications Initial HTML File in Chapter 4 of *Oracle Applications Installation* for details on how to configure this.

Additional Information: Chapter 4, *Oracle Applications Installation*

- Pin stored packages in your Oracle System Global Area.

This prevents your SGA from becoming fragmented and affecting performance.

Additional Information: Chapter 4, *Oracle Applications Installation*

New Functionality

Although the underlying architecture differs, Release 11 includes all the functionality of Release 10 SmartClient. This section highlights some features that are new in Release 11 or that operate differently than in Release 10 and Release 10SC.

Multiple Reporting Currencies

The Multiple Reporting Currencies (MRC) feature allows you to create, maintain, and report on 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. You can set up as many as eight reporting sets of books for each primary set of books.

Typically, you should use MRC if:

- you operate in a country with an unstable currency and you need to concurrently report your business in a hard currency.
- your company is multinational, and you need to report financial information in a common functional currency other than that of the transaction or your primary functional currency.
- you operate in a country that is part of the European Monetary Union (EMU), and you want to concurrently report in Euro in preparation for the single European currency.

Additional Information: Multiple Reporting Currencies in Chapter 3 of this manual

Attachments

In Release 11, you can attach files of nearly any kind—word processing, spreadsheet, image, audio, and video—to records for subsequent viewing or retrieval. You can attach a scanned image to illustrate a product record, for example, or a Microsoft Word file with an embedded Microsoft Excel spreadsheet to justify a purchase order. When opened, attached files are sent to the desktop's web browser, which displays them using the helper application for that MIME type. Attachments can either be files accessible from the web, which are attached using their URL, or files on the local file system, which Oracle Applications uploads and stores on the Oracle8 Server.

Additional Information: Chapter 4, *Oracle Applications Installation*

Spreadsheet Export

You can export the contents of customizable, auto-querying folders and other multi-record regions directly to a spreadsheet program. When opening the spreadsheet, the web browser uses the spreadsheet program set up as the helper application for this file type.

Additional Information: Document Update section, *Oracle Applications User's Guide* (HTML version only)

Context-sensitive Help

Release 11 help text is in HTML format. The help text displays in the client web browser as context-sensitive help when you click the help button (represented as a question mark) on any Oracle Applications screen. Or, you can use the Oracle Information Navigator (OIN) where indicated to perform Full-Text Search on the documentation library.

Additional Information: *Oracle Applications User's Guide*

Report Review

Oracle Applications uses a tool called the *Report Review Agent* to view concurrent processing files online. This utility is on the concurrent processing server and uses Net8 to transfer files from the concurrent processing server to the forms server. It then transmits the file contents—one page at a time or in its entirety—to the desktop client. The forms server sends reports to the client web browser for online viewing in either unformatted text, native HTML or Adobe Acrobat format.

Additional Information: Chapter 4, *Oracle Applications Installation*

System Components

This chapter explains the basic Oracle Applications software components. You can find information on the following topics:

- Release 11 Products and Product Families
- Underlying System Components

Release 11 Products and Product Families

The following table lists the names, abbreviations, and TOP environment variables of products included in Release 11:

Table 2-1 Release 11 Products

Oracle Applications Product	Abbreviation	Environment Variable
DateTrack	dt	DT_TOP
FastFormula	ff	FF_TOP
Global Accounting Engine	ax	AX_TOP
Oracle Alert	alr	ALR_TOP
Oracle Applications Implementation Wizard	az	AZ_TOP
Oracle Application Object Library	fnd	FND_TOP
Oracle Applications DBA	ad	AD_TOP
Oracle Applications Utilities	au	AU_TOP
Oracle Assets	fa	FA_TOP
Oracle Automotive	veh	VEH_TOP
Oracle Bills of Material	bom	BOM_TOP

Table 2–1 Release 11 Products

Oracle Applications Product	Abbreviation	Environment Variable
Oracle Capacity	crp	CRP_TOP
Oracle Cash Management	ce	CE_TOP
Oracle Common Modules	ak	AK_TOP
Oracle Cost Management	cst	BOM_TOP
Oracle EDI Gateway	ec	EC_TOP
Oracle Engineering	eng	ENG_TOP
Oracle Financials for Asia/Pacific	ja	JA_TOP
Oracle Financials for Europe	je	JE_TOP
Oracle Financials for Latin America	jl	JL_TOP
Oracle Flow Manufacturing	flm	FLM_TOP
Oracle General Ledger	gl	GL_TOP
Oracle Human Resources	per	PER_TOP
Oracle Inventory	inv	INV_TOP
Oracle Master Scheduling/MRP	mrp	MRP_TOP
Oracle Order Entry/Shipping	oe	OE_TOP
Oracle Payables	ap	AP_TOP
Oracle Payroll	pay	PAY_TOP
Oracle Product Configurator	cz	BOM_TOP
Oracle Projects	pa	PA_TOP
Oracle Project Manufacturing	pjm	PJM_TOP
Oracle Public Sector General Ledger	gl	GL_TOP
Oracle Public Sector Payables	ap	AP_TOP
Oracle Public Sector Purchasing	po	PO_TOP
Oracle Public Sector Receivables	ar	AR_TOP
Oracle Purchasing	po	PO_TOP
Oracle Quality	qa	QA_TOP
Oracle Receivables	ar	AR_TOP

Table 2–1 Release 11 Products

Oracle Applications Product	Abbreviation	Environment Variable
Oracle Release Management	rla	RLA_TOP
Oracle Application Report Generator	rg	RG_TOP
Oracle Sales and Marketing	as	AS_TOP
Oracle Sales Compensation	cn	CN_TOP
Oracle Self-Service Web Applications	icx	ICX_TOP
Oracle Service	cs	CS_TOP
Oracle SSP/SMP	ssp	SSP_TOP
Oracle Supplier Scheduling	chv	PO_TOP
Oracle Supply Chain Planning	msc	MSC_TOP
Oracle Time Management	hxt	HXT_TOP
Oracle Training	ota	OTA_TOP
Oracle U.S. Federal Human Resources	ghr	GHR_TOP
Oracle Work in Process	wip	WIP_TOP
Regional Localizations	jg	JG_TOP

Product Families

Oracle Applications are grouped into several families of products. These families work together, rely on common underlying components, or have similar industry uses.

Figure 2–1 Oracle Applications Product Families and Products



Oracle Alert

Oracle Alert is installed automatically under the Oracle Application Object Library ORACLE database schema. You can use Oracle Alert with Oracle InterOffice or another mail system that meets the criteria listed in the *Oracle Alert Reference Manual*. You must use Oracle InterOffice to use Oracle Alert response processing, however. When you install Oracle Alert, AutoInstall prompts you for the mail system you use.

Additional Information: *Oracle Alert User's Guide*

Oracle Human Resource Management Systems

FastFormula and DateTrack are available as dependent products only. These products must be installed in the same ORACLE schema as Oracle Human Resources and Oracle Payroll. The default user ID for this product family is HR/HR. You can, however, install other HRMS products in their own ORACLE database schemas.

Oracle Manufacturing

Oracle Manufacturing refers to the product family listed in Figure 2-1. Oracle Inventory provides the basic Oracle Cost Management functionality. Oracle Bills of Material and Oracle Work in Process provide the remaining functionality. Oracle Cost Management, therefore, does not show up on the AutoInstall screens as a separate product that you can install.

Oracle Projects

Oracle Projects consists of either Oracle Project Billing or Oracle Project Costing. When you choose to install Oracle Projects, AutoInstall prompts you to choose the Billing or the Costing product. When you install Oracle Project Billing, you automatically get Oracle Project Costing as well.

System Issues

This section discusses installation issues and restrictions for Oracle Applications systems in general.

Required Products

Each Oracle Applications installation requires the following products:

- **Oracle Application Object Library**
Provides the common functions for Oracle Applications products.
- **Oracle Alert**
Provides shared messaging and alert functionality. It is installed automatically as a shared product. To use the functionality of Oracle Alert, you must select it to be fully installed.
- **Oracle Applications DBA**
Provides programs that let you install, upgrade, administer, and maintain an Oracle Applications system. It is installed automatically as a shared product only.
- **Oracle Applications Utilities**
Includes shared PL/SQL resources (including PL/SQL libraries, graphics, and forms) used by all Oracle Applications products. AutoInstall copies these shared resources automatically from other product directories.

When you estimate required system resources, be sure that you have enough disk space for these products as well as for all other Oracle Applications products you have licensed.

Full Product Installations Required

In addition, some Oracle Applications products require full installations of other products. AutoInstall automatically sets the required products to install as necessary. For example, Table 2–2 shows that you must install Oracle Inventory fully to install Oracle Bills of Material. When you choose to install Oracle Bills of Material, AutoInstall sets Oracle Inventory to install fully.

Table 2–2 Full Product Installation Requirements

A full installation of this product...	Requires full installations of these products...
Oracle Bills of Material	Oracle Inventory
Oracle Capacity	Oracle Bills of Material, Oracle Inventory, Oracle Master Scheduling/MRP
Oracle Engineering	Oracle Bills of Material, Oracle Inventory
Oracle Flow Manufacturing	Oracle Bills of Material, Oracle Inventory
Oracle General Ledger	Oracle Report Generator
Oracle Master Scheduling/MRP	Oracle Bills of Material, Oracle Inventory

Table 2–2 Full Product Installation Requirements

A full installation of this product...	Requires full installations of these products...
Oracle Order Entry/Shipping	Oracle Inventory, Oracle Receivables
Oracle Payroll	Oracle Human Resources
Oracle Product Configurator	Oracle Bills of Material, Oracle Inventory, Oracle Order Entry/Shipping, Oracle Receivables
Oracle Report Generator	Oracle General Ledger
Oracle Quality	Oracle Inventory
Oracle Release Management	Oracle EDI Gateway, Oracle Inventory, Oracle Master Scheduling/MRP, Oracle Order Entry/Shipping, Oracle Receivables
Oracle Sales Compensation	Oracle General Ledger
Oracle Service	Oracle Order Entry/Shipping
Oracle SSP/SMP	Oracle Human Resources
Oracle Supplier Scheduling	Oracle Purchasing
Oracle Supply Chain Planning	Oracle Bills of Material, Oracle Inventory, Oracle Master Scheduling/MRP
Oracle Time Management	Oracle Human Resources, Oracle Payroll
Oracle U.S. Federal Human Resources	Oracle Human Resources
Oracle Work in Process	Oracle Bills of Material, Oracle Inventory

Dependent Products

AutoInstall installs all objects and seed data for all Oracle Applications products (except JA, JE, and JL) in your database, regardless of whether you have licensed the product.

In addition, the Oracle Applications products you license for full installation may require product file components from *dependent* (also called *shared*) products. For example, Oracle Purchasing uses product file components of Oracle Payables. This makes Oracle Payables a dependent product of Oracle Purchasing. If you fully install Oracle Purchasing but not Oracle Payables, Oracle Payables is installed automatically as a dependent product.

A dependent product installation includes only the product files needed by your combination of fully installed products. In the example in the previous paragraph,

the Oracle Payables dependent product installation includes only the files needed by Oracle Purchasing.

Before you install an Oracle Applications system, you need to verify that you have enough database space for all database objects for all Oracle Applications. Also verify that you have enough product file disk space for dependent products as well as for fully installed products. *Oracle Applications Installation* lists the disk space required for dependent products.

Localizations

Localizations are modules that extend Oracle Applications to meet the unique business requirements of one or more geographic regions. Localizations are different from customizations, which are created to meet the unique requirements of a specific site.

Release 11 has four localization module sets:

- Asia/Pacific
- Europe
- Latin America
- Regional

Each localization module contains forms, reports, seed data, and database objects for use with Oracle Applications in that region. By default AutoInstall does not select localizations for installation. You select the localization modules to install.

Additional Information: Localization Directories in Chapter 4 of this manual

Oracle Applications Language Translations

Oracle Applications is available in several languages. When you license Oracle Applications in a language other than American English, you receive an additional CD that contains translated forms, reports, seed data, help text, and messages. *Oracle Applications Installation* includes information on how to install Oracle Applications in any of the supported languages.

Additional Information: Language Directory Structure in Chapter 4 of this manual

Underlying System Components

This section explains what you need to know about the operating system, the Oracle8 Server, and the Oracle Tools to install or upgrade Oracle Applications.

UNIX Operating System

Running the AutoInstall program and finishing an installation or upgrade requires knowledge of the UNIX operating system. At a minimum, you should know how to do the following:

- start a UNIX Bourne or Korn shell
- create UNIX user logins
- create mount points and directories
- set file privileges
- edit a file with a text editor
- determine the amount of available real memory, virtual memory, and file system disk space

The commands listed in this manual are standard across most UNIX platforms. Check the *Oracle Applications Installation Update* for your platform for additional information that is specific to your platform.

Oracle8 Server

Oracle Applications uses Oracle8 Enterprise Edition Release 8.0.4 (or later versions that are certified for compatibility) to store and update Oracle Applications data. Your site needs a database administrator who has access to the ORACLE SYS and SYSTEM accounts. The database administrator must be able to do the following:

- create a database
- create a control file
- set database initialization parameters
- shut down and restart a database
- create and size tablespaces and rollback segments
- export and import a database

The Oracle8 Server documentation explains how to set up and administer a database.

Additional Information: *Oracle8 Concepts and Oracle8 Administrator's Guide*

Oracle Tools

A typical Oracle Applications user does not use the Oracle tools directly. Oracle Applications reference manuals explain how to use forms and print reports from within the applications. On a system level, however, Oracle Applications requires the following Oracle tools in addition to the Oracle8 Server:

Oracle Product	Certified Version
The following tools are installed with Oracle8 Enterprise Edition:	
Oracle8 Server Enterprise Edition	8.0.4.0.0
SQL*Plus	8.0.4.0.0
Net8	8.0.4.0.0
JDBC	8.0.4.0.0
Oracle8 ConText Cartridge	2.3.6.0.0
PL/SQL	8.0.4.0.0
SQL*Loader	8.0.4.0.0
The following tools are installed with Developer/2000:	
Developer/2000	1.6.1
Oracle Reports	2.5.7.5.0
Oracle Forms	4.5.10.3.3
Java Developer's Kit	1.1.5.12 o
Pro*C is installed with Oracle8, and is used to write C language extensions to Oracle Applications:	
Pro*C	8.0.4.0.0
Oracle Web Application Server	
Oracle Web Application Server	3.0.1

WARNING: Specific versions of these tools are certified for use with this release of Oracle Applications. The tools must be installed and running before you install Oracle Applications. Contact Oracle Support Services to find out which later versions of these tools may be certified for compatibility with Oracle Applications.

Data Model

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

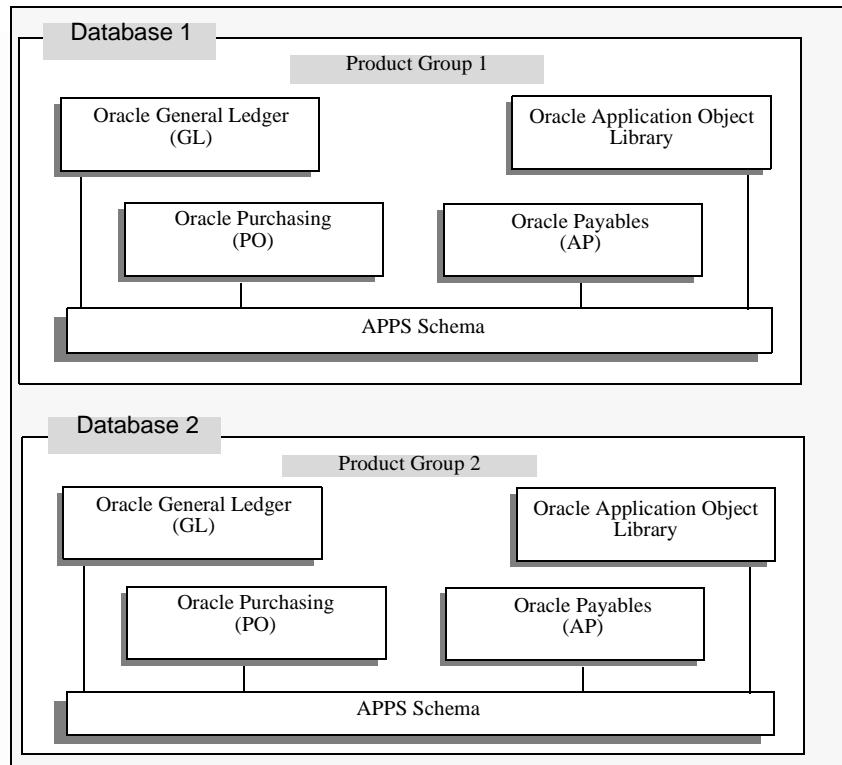
- Basic Data Model
- Multiple Organization (Multi-Org) Architecture
- Multiple Reporting Currencies
- Multiple Sets of Books
- Oracle8 Server Organization and Requirements

Basic Data Model

Each installation of Oracle Application Object Library tables in an ORACLE database defines a product group. Each product group can contain any number of Oracle Applications products in addition to Oracle Application Object Library.

Product Groups

In Release 11, you can have only one product group in each database. Figure 3-1 illustrates two identical product groups installed in separate databases. Because the products reside in separate databases, they can use the same ORACLE schema names. The schema names are shown here as the default names GL, PO, AP, and APPLSYS. The products installed in each database do not have access to the data of products installed in another database. For example, PO installed in Database 1 cannot access the data from AP installed in Database 2.

Figure 3–1 Multiple Product Groups in Separate Databases

ORACLE User IDs

Each Oracle Applications product has a default ORACLE user ID that uses 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 during the installation, 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) and the user accesses all data objects through the APPS schema, AutoInstall creates the necessary grants and synonyms between the schemas.

A product's data objects can be installed in any tablespace regardless of whether it shares a schema with another product.

APPS Schema

The APPS schema is an ORACLE schema that has access to the complete Oracle Applications data model. It is analogous to the SYSTEM schema, which has access to the entire database. The APPS schema dramatically simplifies and reduces the time of the installation process and the stability of upgrades.

Oracle Applications responsibilities connect to an APPS schema. AutoInstall automatically sets the environment variable FNDNAM to the name of the APPS schema.

APPS and Base Product Schemas

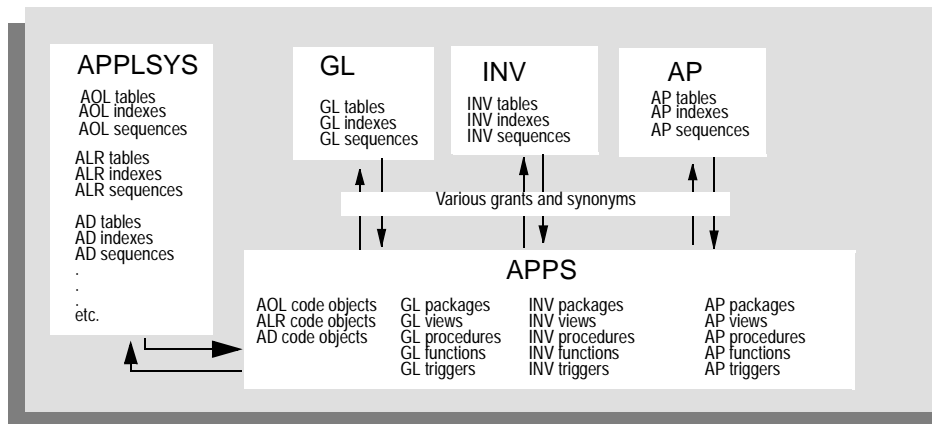
AutoInstall creates one schema for each product—a base product schema—which owns the data objects, and one schema—APPS—which owns the code objects and has access to all data objects. There is one APPS schema for every product installation group. The following code objects are installed in each APPS schema:

- 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 3–2 APSS Schema and Base Product Schemas

APSS Schema Benefits

The APSS 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.

In Release 10.5 (and earlier), you used an Oracle Applications product through the base product schema. All Oracle Inventory responsibilities, for example, connected to the Oracle Inventory schema (like INV). In order to use functionality in another product, such as purchase orders in Oracle Purchasing, the current schema needed access to the shared product's objects. This access was accomplished by grants and synonyms.

For example, the PO schema granted access on PO_HEADERS to the INV schema and the INV schema created a synonym called PO_HEADERS to PO.PO_HEADERS. The synonym allowed object reference without the necessity of qualifying it by schema. So, by using the INV schema, you could execute all functions necessary to use Oracle Inventory, including those that used components outside the INV schema. Similarly, you would connect to the Oracle Payables or Oracle General Ledger product schemas when using those products' responsibilities.

This architecture, however, required a complex and extensive set of grants from and synonyms to objects in other products' schemas. Utilizing one schema that has access to all objects eliminates these cross-product dependencies and creates a hub-and-spokes model rather than a spider web.

How AutoInstall Creates APPS Schemas

AutoInstall performs these steps to create and maintain the APPS schemas:

1. Creates the APPS schema or schemas.

AutoInstall creates one APPS schema per product installation group using the following naming conventions:

- If the AOL schema is named APPLSYS, the first APPS schema will be named APPS. Subsequent APPS schemas will be named APPS<install group number>, like APPS2, APPS3, and so on.
- If the AOL schema is not named APPLSYS, the first APPS schema will be named APPS_<AOL Schema Name>. Subsequent APPS schemas will be named APPS<Install Group Number>_<AOL Schema Name>. For example, if the AOL schema is named APPLMGR, then the first, second and third APPS schemas are named APPS_APPLMGR, APPS2_APPLMGR and APPS3_APPLMGR.

2. Maintains the APPS schema or schemas automatically.

AutoInstall grants privileges to and creates synonyms in the APPS schema(s) for the base product objects, and creates code objects in the APPS schema(s).

3. Registers the APPS schemas.

Each Oracle Applications schema is listed in the FND_ORACLE_USERID table. This table describes the schemas that currently exist in the database for Oracle Applications use. AutoInstall registers each APPS schema by creating entries in this table.

4. Associates each data group unit with the appropriate APPS schema.

Data group units map applications to ORACLE schemas for a data group (which is used by a responsibility). AutoInstall sets the ORACLE schema for each Oracle Application to the appropriate APPS schema. This change makes Oracle Applications run from an APPS schema instead of from a base product schema.

5. During an upgrade, AutoInstall changes the APPS schema password.

When upgrading, AutoInstall automatically sets each APPS schema password to match the AOL schema password.

Warning: If you change the AOL schema password after the installation or upgrade, you must also change each APPS schema password to match it.

Localizations Objects

A localization module may include new database objects. If a module is for one Oracle Applications product, the database objects may be installed in the product's base schema or in the appropriate region's schema, such as in the JE schema or JA schema. Additional space should be made in the table and index tablespaces for these database objects.

By default, AutoInstall creates an ORACLE schema for JG (Regional Localizations). It creates JE, JL, or JA schemas only if you install European, Latin America, or Asia/Pacific localizations, respectively.

Custom Schema Access

In some circumstances you may wish to create a schema that has limited or read-only access to Oracle Applications data. You may 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 APPS schema, not from a base product schema.

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

Data Access

Some views access packages or functions, 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 than if you are running within the environment of Oracle Applications. 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

New sites use the Multiple Organization architecture rather than the old Multiple Sets of Books architecture. This means that you use a single installation of any Oracle Applications product 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.

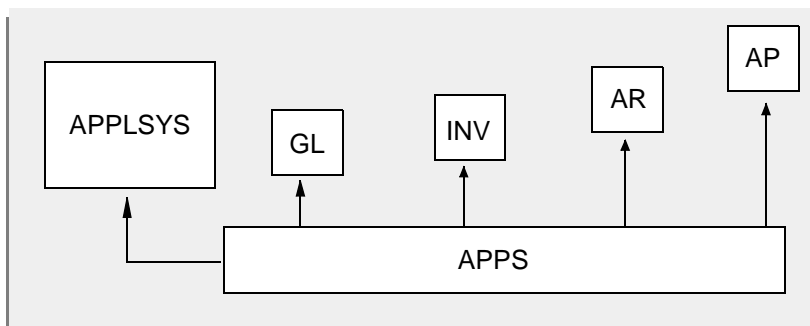
It is important that you understand these terms as you work with Multi-Org architecture:

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 Order Entry, Oracle Payables, Oracle Purchasing, or 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.

A single product installation group architecture is used by Multi-Org, or if you do not use the old Multiple Sets of Books architecture. There is one APPS schema and a single schema for each product.

Figure 3–3 Single Product Installation Group



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.

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*

Changes from Release 10.6

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 upgrade, 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. Previously, the text of the views was created dynamically based upon the site's configuration.

In Release 10.6, operating unit schemas were necessary to partition data by operating unit. Release 10.6 Multi-Org views incorporated a DECODE statement based upon the name of the schema. Beginning with Release 10.7, Multi-Org views use a DECODE based upon CLIENT_INFO. Therefore the Release 10.6 operating unit schemas are now obsolete.

In Release 11, you use adadmin to initially convert seed data and transaction data to Multi-Org. The adadmin utility does not change the data model. When you create a new operating unit, a concurrent program adds the appropriate seed data. You no longer need to run adadmin (as you did in Release 10.6) after applying any database patches to a Multi-Org implementation, since the Multi-Org architecture is built into the underlying applications.

Additional Information: AD Administration Utility in Chapter 5 of *Oracle Applications Installation*

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 adadmin program to validate and compile your APPS schema, and install the MRC schema.

Additional Information: Multiple Reporting Currencies, *Oracle Applications Installation*

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:

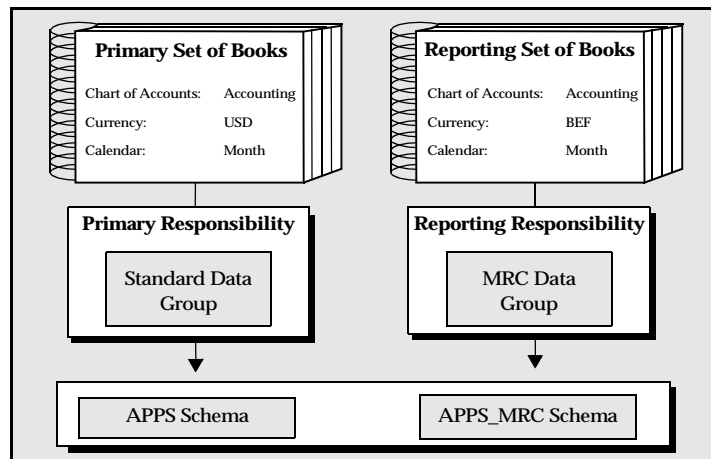
- To identify the ORACLE schema used by a responsibility
- To tell the concurrent manager which ORACLE schema to use when running a report or concurrent program

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 than the default name of APPS, your MRC schema will be named according to the following format:

<name of your APPS schema>_MRC

Figure 3–4 MRC Set of Books, Responsibility, and Database Schema Setup



MRC Architecture

MRC is supported by the following Oracle Applications:

- General Ledger
- Assets
- Payables
- Receivables
- Purchasing

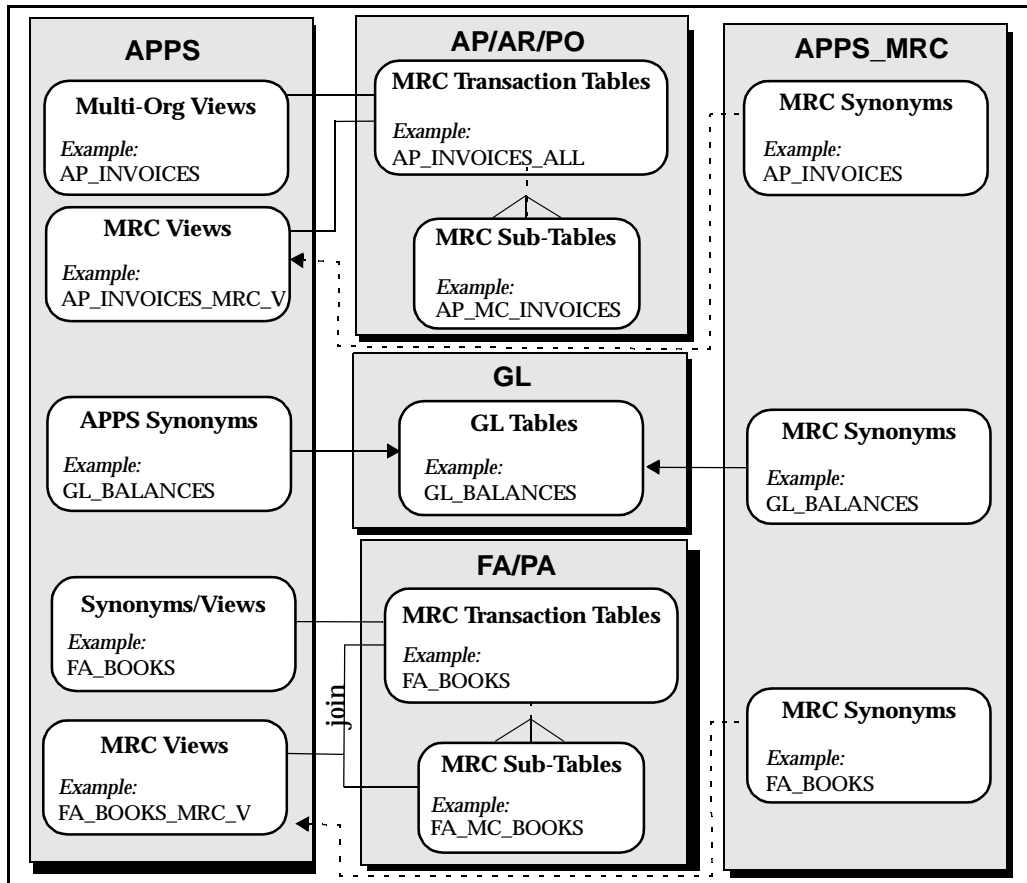
- Projects

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 sub-table. Each record in the base transaction table may have one or more MRC sub-table records, which contain currency-sensitive information on the base transaction in each associated reporting currency. In the APPS schema, there are MRC views, which 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 for FA) 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 or Multiple Sets of Books Architecture is not implemented.

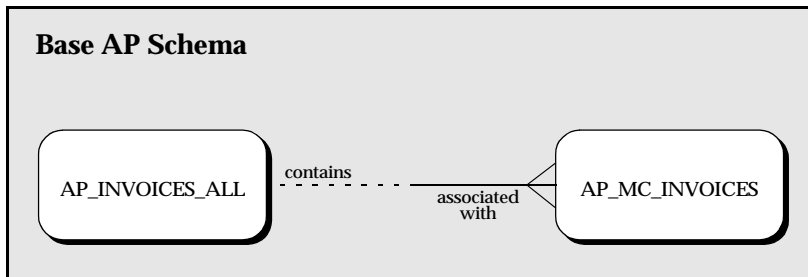
Figure 3-5 MRC Architecture



MRC Sub-Tables

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 sub-table.

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

Figure 3–6 Transaction Table and MRC Sub-Table

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 sub-tables.

For example, when you enter an invoice in Oracle Payables, 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 sub-tables AP_MC_INVOICES and AP_MC_INVOICE_DISTs. See *Multiple Reporting Currencies in Oracle Applications* for information about the MRC sub-tables 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 Views in Oracle Assets and Projects In Oracle Assets and Projects, the MRC views are joined table views, which consist of columns from the base transaction tables and the corresponding MRC sub-tables. These views retrieve currency-sensitive information from the MRC sub-tables for your reporting set of books.

MRC Views in Oracle Payables, Receivables, and Purchasing For performance reasons, MRC views in Payables, Receivables, and Purchasing are based on additional columns in the base transaction tables, instead of the MRC sub-tables. These columns consist of pairs of concatenated strings of the reporting set of books ID and its corresponding selected data. These columns are also maintained by the MRC database triggers.

AP_INVOICES_ALL	
invoice_id	NUMBER(15)
set_of_books_id	NUMBER(15)
invoice_date	DATE
amount	NUMBER
org_id	NUMBER(15)
base_amount	NUMBER
mrc_base_amount	VARCHAR2(2000)
posted_flag	VARCHAR2(1)
mrc_posted_flag	VARCHAR2(2000)

In the example, the column `mrc_base_amount` may contain information in the following format:

```
<reporting set of books id1>accounted amount1<reporting set of books id2>
accounted amount2.....
```

For example:

```
<104>35.49<105>389.47....
```

The MRC views in Oracle Payables, Receivables, and Purchasing are single-table views built on top of base transaction tables. Based on the reporting responsibility to which you are logged on, these views extract the currency-sensitive information from the MRC VARCHAR2 columns for the reporting set of books. Using the example above, if you are currently in a reporting responsibility associated with the reporting set of books ID 105, the MRC view on top of AP_INVOICES_ALL (that is, AP_INVOICES_MRC_V) will extract from the `mrc_base_amount` column a base amount of 389.47. See *Multiple Reporting Currencies in Oracle Applications* for information about base transaction tables that use additional columns to store reporting currency information.

Oracle Payables, Receivables, Purchasing, and Projects have Multi-Org views to perform data partitioning. These views, which reside in the APPS schema, use the information contained in the CLIENT_INFO environment variable to determine the operating unit. The MRC views give organization-specific and currency-specific views of the transaction tables.

MRC Synonyms

From the APPS_MRC schema, the MRC synonyms, which have names identical to the corresponding APPS Multi-Org views (or synonyms for FA), 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.

General Ledger

Oracle General Ledger does not use MRC sub-tables to store, or MRC triggers to create, reporting currency information. Primary and reporting journal entries are created separately in the primary set of books and in each reporting set of books in the base journal tables. Synonyms are created identically in both the APPS and the APPS_MRC schemas.

Additional Information: *Multiple Reporting Currencies in Oracle Applications*

Oracle General Ledger and Oracle Applications modules that do not implement MRC are also maintained under the APPS_MRC schema. This is required to facilitate the integration and drilldown from these modules to the Oracle subledgers that support MRC in the reporting responsibilities.

Multiple Sets of Books

Multiple Sets of Books architecture from earlier releases is not automatically consolidated or converted to the Multiple Organization architecture. You can still upgrade an existing Multiple Sets of Books implementation to Release 11 without using Multiple Organization architecture.

In earlier releases, a set of books defined a company or group of companies within Oracle Applications that shared a common Accounting Flexfield structure, calendar, and functional currency. Oracle Applications let you set up multiple sets of books that each had its own flexfield structures, calendars, and currencies.

You set up multiple sets of books by installing one product group for each set of books. We implemented multiple sets of books within a single product group differently for different products. Some products, such as Oracle Ledger, maintained multiple sets of books within one installation of their product tables. Others, such as Oracle Application Object Library, did not maintain multiple sets of books at all. We called these two classes of products *SOA products* (Single Oracle Account), because a single copy of the database objects is installed.

Other products, such as Oracle Payables, maintained multiple sets of books using multiple installations of their product tables. We called these *MOA products* (Multiple Oracle Account), because we installed multiple copies of the database objects for these products—one copy for each set of books.

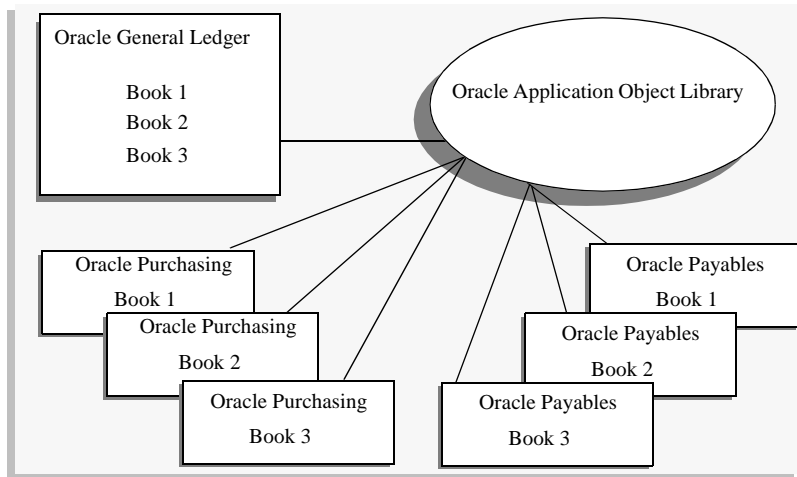
The \$APPL_TOP/admin/applprod.txt file is a text file listing information for each Oracle Applications product. The applprod.txt file contains, among other things, a notation on whether a product requires multiple copies of the database objects. If the line immediately after "Multiple product installations for MSOB" begins with Yes, the product is an MOA product and requires multiple installations of the product tables. If the line begins with No, the product is SOA.

If you upgrade a Multiple Set of Books implementation, each installation of product tables must be in a unique ORACLE schema. For example, the Oracle Payables tables for the first set of books may be installed in the AP schema, while the tables for the second set are installed in the AP2 schema.

Note: Doing so requires a considerable amount of free SYSTEM tablespace.

Figure 3–7 illustrates a product group consisting of Oracle Application Object Library, Oracle General Ledger, Oracle Purchasing, and Oracle Payables. The products in this group, except for Oracle Application Object Library, maintain three sets of books.

Suggestion: Multiple Sets of Books architecture will not be supported in future releases. Contact Oracle Consulting Services before upgrading Multiple Sets of Books architecture or converting to Multi-Org architecture.

Figure 3–7 Multiple Sets of Books Architecture

Multiple Product Installation Groups and Multiple Sets of Books

The multiple product installation group architecture was used with Multiple Sets of Books Architecture in earlier releases to implement multiple sets of books accounting functionality.

Previously you would install multiple copies of the same product (say Oracle Receivables), each in a different ORACLE schema (for instance AR1 and AR2), in order to partition the product's data by set of books.

Some products, such as GL and APPLSYS, only needed one product install no matter how many sets of books you maintained. But most products required multiple installs. There was one APPS schema as well for each set of books.

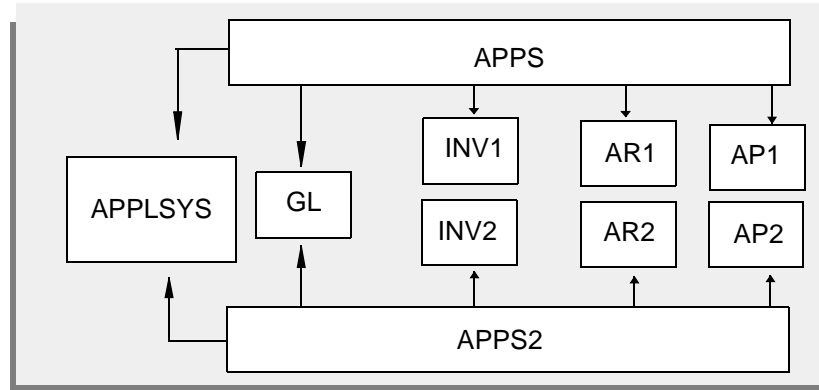
Figure 3–8 Multiple Product Installation Groups

Figure 3–8 illustrates a multiple product installation group architecture for two sets of books. There were two APPS schemas, two INV schemas, two AR schemas and two AP schemas. The APPLSYS schema, like the GL schema, was installed only once.

With MSOB multiple product installation groups, it was difficult to have cross-product or cross-set of books functionality. With the installation shown in Figure 3–8, it would be difficult to view all inventory items in all sets of books. Although a global view incorporating data from multiple sets of books was possible, it tended to be inefficient and complex. Beginning with Release 10.6, Multiple Organization architecture resolved this problem.

Oracle8 Server Organization and Requirements

This section discusses how to set up an Oracle8 Enterprise Edition database for use with Oracle Applications. It provides recommendations on tablespaces to use and explains how the AutoInstall sizing factor controls the amount of database space reserved for Oracle Applications products.

You should plan to distribute tablespaces across different disks. This can reduce disk head contention and improve system response time. Refer to your Oracle8 Server documentation for complete information about planning a database and tablespaces.

Additional Information: PART III: Oracle Server Configuration, *Oracle8 Administrator's Guide*; Chapter 1: Optimal Flexible Architecture on Oracle8, and Chapter 3: Tuning Oracle8, *Oracle8 Administrator's Reference* (for your platform)

Database Creation

Observe the following recommendations and requirements when you create a Oracle8 database for use with Oracle Applications. See your Oracle8 Server documentation for complete information about database creation.

Additional Information: Creating an Oracle8 Database, *Oracle8 Administrator's Guide*

Database Character Set

You must specify a character set when creating the database. It is important to consider carefully which character set to use because you cannot change it after the database is created.

Oracle Applications require you use an eight-bit character set or the appropriate character set for your language. Appendix A in *Oracle Applications Installation* contains a table that lists the default character sets for the languages supported by Oracle Applications. For example, WE8ISO8859P1 is used for American English and ZHT16BIG5 is used for Traditional Chinese.

Additional Information: Appendix A, *Oracle Applications Installation*

We recommend you use the default character set. You may choose an alternate character set if there is a conversion path to the default character set. If you do, AutoInstall automatically converts text files to the new character set when it unloads the files. For example, you may install into a JA16SJIS database and client environment, even though the files are in JA16EUC on the distribution media.

Data Files

Keep data files to a size that is convenient to back up on your system. You may want the maximum file size to be 300 to 600 MB, for example. You can create the database with one or more data files for tablespaces used by the Oracle8 Server, such as SYSTEM. You can then create one or more new data files for each Oracle Applications tablespace you set up.

When you create the database, set the MAXDATAFILES parameter to a value larger than the number of data files you intend to create. A value such as 256 is typically sufficient. The limit is platform-specific. Note that you can alter the MAXDATAFILES parameter by recreating the database control file.

Additional Information: Managing Control Files, *Oracle8 Administrator's Guide*

The `DB_BLOCK_SIZE` parameter in the database initialization parameter file indicates the size of ORACLE database blocks that are created when you `CREATE DATABASE`. For optimum performance, set this to 8192 before you create your database. If you are upgrading from an earlier release, however, do not change the `DB_BLOCK_SIZE` parameter. If you change the parameter when upgrading, you may not be able to generate ORACLE control files.

Additional Information: Creating an Oracle Database, *Oracle8 Administrator's Guide*

Initialization Parameters

Set the database initialization parameters to the "large" settings in the initialization parameter file (`init.ora`). Set the `DB_FILES` parameter to a value less than or equal to the value specified for `MAXDATAFILES` during database creation. Your Oracle8 Server database administrator will have to modify other parameters in this file before you install Oracle Applications.

Oracle8 Server Tablespaces

The following are recommendations for setting up tablespaces for the Oracle8 Server and for Oracle tools. Your database administrator may want to set up the tablespaces differently, however. See the section Oracle Applications Tablespaces in this manual for sizing guidelines for the tablespaces listed in this section.

SYSTEM Tablespace

This tablespace, which holds data dictionary tables owned by the `SYS` account, is created when you install the database. You may have to increase its initial size to accommodate the growth of data dictionary tables that occurs when you upgrade Oracle Applications.

Tools Tablespace

Tools tables, such as forms and report tables, are created in the default tablespace of the ORACLE schema used to install the tools. This default tablespace is typically `SYSTEM`. However, you may want to install these tables in a tablespace such as `TOOLS` if you plan to develop applications. This can help reduce resource contention between developers and Oracle Applications users.

Users Tablespace

Create a tablespace, such as USERS, for ORACLE schemas that hold demonstration tables for Oracle Tools. These are IDs such as SCOTT/TIGER and EXAMPLE/EXAMPLE. You can also use this tablespace as the default for database administrator accounts.

Rollback Tablespace

The Oracle8 Server uses rollback segments to track database changes until they are committed. It uses them during an installation or upgrade of Oracle Applications, as well as during normal, daily use. Create a tablespace such as RBS for rollback segments.

Oracle Applications Tablespaces

You must create product tablespaces before you install your Oracle Applications products. The following sections give recommendations on the tablespaces to set up. See the section on Sizing Factor in this manual for sizing guidelines.

You designate the table, index, and temporary tablespaces used by each Oracle Applications product when you install the products.

Suggestion: Your tablespace design affects how your system operates and can affect system expansion and upgrades. An Oracle Applications consultant can help you set up tablespaces that meet your present and projected future system requirements.

Product Tablespaces

For each Oracle Applications product, we recommend that you create a tablespace for tables and another one for indexes. Creating separate table and index tablespaces for each product makes it easier to license and install products later and may increase database performance.

Some products maintain multiple sets of books through multiple installations of product tables. If you already use this architecture, you can install each set of these product tables in separate tablespaces. For example, you could set up a new table tablespace and a new index tablespace for a new set of Oracle Payables books. The tables and indexes for some dependent products will be installed multiple times as well. You can install all of these in the dependent product's main table and index tablespaces.

Designate tablespaces with a standard naming convention, such as the product's short name or ORACLE schema name along with D for data or X for indexes. For example, create the tablespaces APD and APX for Oracle Payables tables and indexes, respectively. The tablespaces for a second product installation group could be AP2D and AP2X.

AutoInstall automatically sets the MAXEXTENTS for all product tablespaces to UNLIMITED.

Temporary Tablespaces

The Oracle8 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 AOL) is the tablespace used by each product.

Interim Tablespace

Oracle General Ledger and Oracle Government General Ledger use large interim tables during certain processes. Create an interim tablespace such as INTERIM if you plan to install one of these products. You designate the interim tablespace after installing the product.

Additional Information: Storage Parameters for Interim Tables, *Oracle General Ledger User's Guide* or *Oracle Public Sector General Ledger User's Guide*

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 AutoInstall. The sizing factor affects only the size of next extents, determined by the Oracle8 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.

Additional Information: Chapter 3: Data Blocks, Extents, and Segments, *Oracle8 Concepts*

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 actually 16 KB.

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

Table 3–1 Sample Extent Sizes for a 100% Sizing Factor

Tables	Next Extent
AP_AGING_PERIODS	32 KB
AR_STATEMENT_LINE_CLUSTERS	1 MB
Indexes	Next Extent
AP_AGING_PERIODS_U1	32 KB
AR_STATEMENT_LINE_CLUSTERS_N1	256 KB

Oracle Applications Installation lists the database space used by product tables and indexes installed with a 100% sizing factor. The tables in *Installation* tell you the sizes of the tablespaces required to hold products when all database objects are allocated two extents. 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.

Sizing Factor Greater or Less than 100%

A sizing factor greater or less than 100% (except for 0%) creates database objects at that percentage of their 100% size. For example, a 200% sizing factor doubles each object's 100% next extent sizes. The 200% sizing factor creates AP_AGING_PERIODS with next extents of 64 KB.

If you do not use the default sizing factor, use a sizing factor that is 100% multiplied or divided by a factor of 2, such as 50%, 200%, or 400%. There is no upper limit on sizing factors. They do not affect static objects, such as product lookup tables, that generally do not increase in size beyond the initial seed data. AutoInstall creates static objects at one size regardless of the product's sizing factor.

Suggestion: As the default initial extent size is relatively small, usually 4 KB, we do not recommend a sizing factor less than 100%. If you had sizing factors less than 100% in earlier releases, we suggest you change them to 100%. When installing systems for low-transaction testing or training purposes, however, use 0% sizing to conserve space.

File System

This chapter discusses file management for Oracle Applications. It includes the following topics:

- Directory Structure
- Log and Output Files

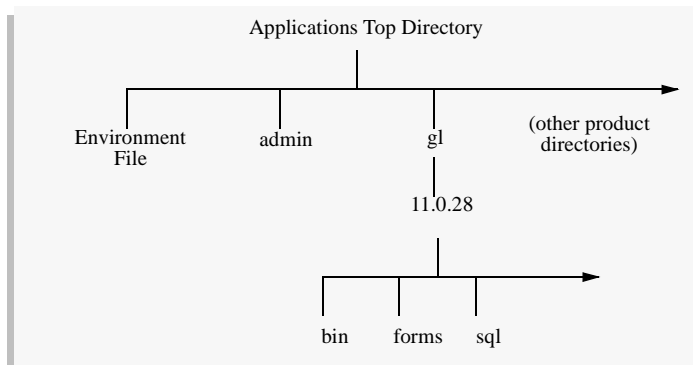
Directory Structure

This section explains the directory structure of Oracle Applications product files. These include the forms, reports, files to install or upgrade the database, and other components shipped with the software release. Tables and other database objects are stored separately in an Oracle8 Server database.

Basic Directory Structure

In the basic directory structure, a top Applications directory holds one set of product files. You create this directory and set the environment variable `APPL_TOP` to point to it. `AutoInstall` sets up a directory tree for each fully installed or dependent product in this directory. `AutoInstall` also sets up a minimal product tree for all other products so it can install their database components.

A product directory tree starts with a directory that uses the product's abbreviation, such as `gl` for Oracle General Ledger. Below that is a subdirectory that uses the product version, such as `11.0.28`. Below that are various subdirectories for product files.

Figure 4–1 Basic Directory Structure

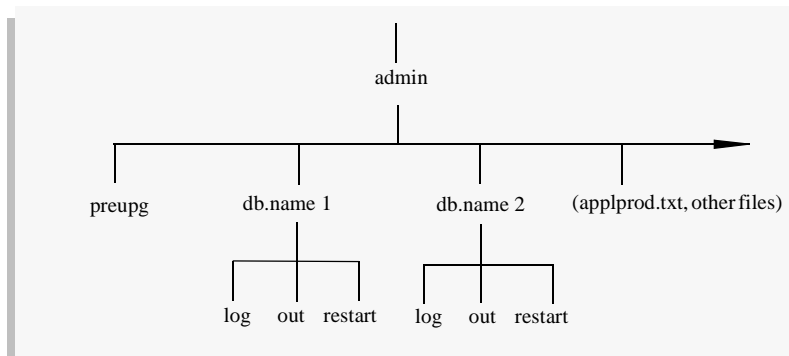
Product TOP Variables

AutoInstall sets <prod>_TOP environment variables that point to each product's directory tree. In the installation depicted in Figure 4–1, the variable GL_TOP points to \$APPL_TOP/gl/11.0.28. To display data entry forms for Oracle General Ledger, for example, the Oracle Applications system accesses files in SGL_TOP/forms.

Additional Information: Table 2–1 in this manual lists product top directory names.

Admin Directory and Environment File

The admin directory (which replaces the install directory from previous releases) holds files used to install or upgrade any Oracle Applications products. This admin directory is used by AutoInstall. Subdirectories in this directory hold the log and restart files that record the actions AutoInstall performs.

Figure 4–2 *admin Directory Structure*

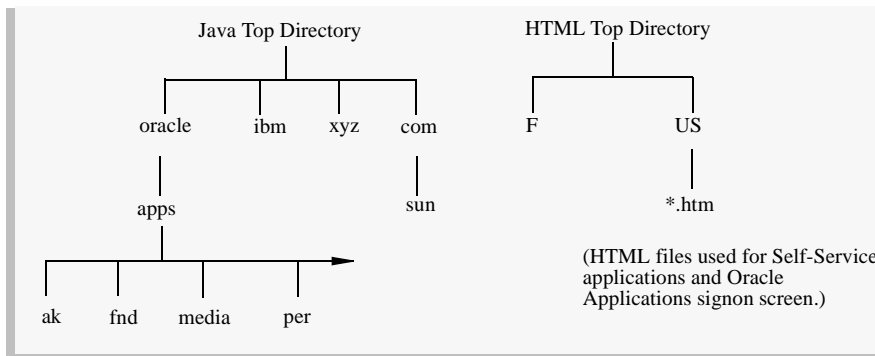
An environment file stored in the APPL_TOP directory defines the <prod>_TOP variables and other environment variables. For example, your environment file might be called APPLTEST.env. The default name for environment files is derived from the name of the current database. For example, if you are installing to database PRODUS, AutoInstall and the adadmin utility will name the environment file PRODUS.env, unless you choose a different name.

Oracle Applications users must run an environment file before logging on to Oracle Applications in order to set up the appropriate environment.

Additional Information: Appendix A, *Oracle Applications Installation*

Java and HTML Files

Java file naming conventions indicate that a machine should have a single directory under which all Java class files are stored. This directory contains multiple namespaces, which provide a way of keeping Java programs written by one company from interfering with those written by another. A namespace maps to a subdirectory tree underneath the top-level Java directory. This way, for example, XYZ Corporation can have a file called "uiDisplay.class," and Oracle Corporation can have a file with the same name. Oracle Applications will always know which one to use because the Oracle version of this file is always installed in the Oracle namespace—the oracle subdirectory.

Figure 4–3 Java and HTML Directory Structure

Oracle Applications requires that you define the location of Java files on your system using the `JAVA_TOP` environment variable before you begin your installation or upgrade. The Java and HTML directories can be anywhere in your file system—they do not have to be under `APPL_TOP`. AutoInstall will then install all Applications class files in the Oracle namespace under your `$JAVA_TOP` directory—`$JAVA_TOP/oracle/apps`. Each product will install its Java class files into a subdirectory beneath this directory, as shown in the example figure.

Additional Information: Edit your `adovars.env` File, *Oracle Applications Installation*

Note: Oracle Applications installs all media files, such as GIF images, under the `$JAVA_TOP` directory. You must not change this setting. If you do, Oracle Applications will not work properly.

In order to make it easier to maintain the virtual directory settings for your web server, Oracle Applications also consolidates all HTML files into a single directory tree during installation. You determine the location of this top-level directory by setting the environment variable `OAH_TOP` before you begin your installation or upgrade. The HTML file used to launch the initial Applications signon screen is typically installed here, as are all HTML files used by the Self-Service Web Applications product. Note that there are subdirectories named for the language code. See the Language Directory Structure section in this chapter for more information.

Additional Information: Edit your `adovars.env` File and Setting Up the Web Server, *Oracle Applications Installation*

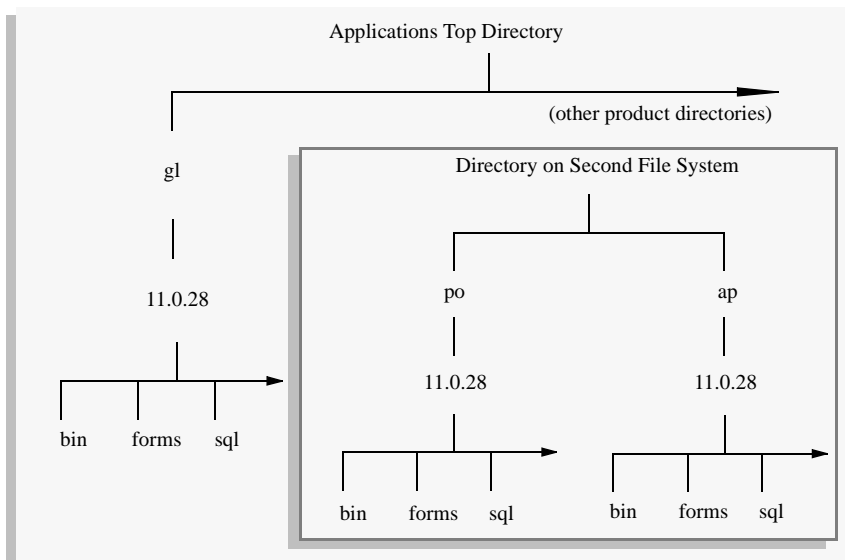
Distributed Directory Structure

A *distributed directory structure* places product files on different file systems. You must use a distributed structure if you do not have enough disk space available on a single file system for all Oracle Applications product files.

Figure 4–4 illustrates a distributed directory structure. Oracle General Ledger product files are stored in \$APPL_TOP, with Oracle Purchasing (po) and Oracle Payables (ap) files stored on a different file system. AutoInstall lets you define which file system holds each product's directory tree.

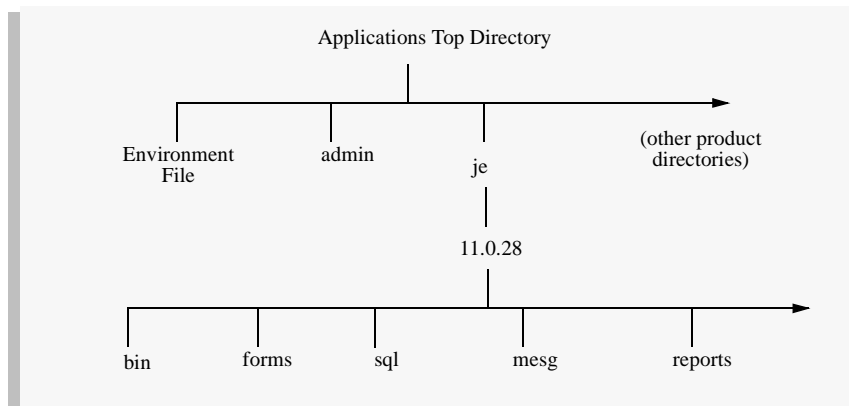
Product files for Oracle Application Object Library, Oracle Applications DBA, and Applications Utilities are unloaded automatically in \$APPL_TOP when you prepare your system to run AutoInstall. You cannot unload these files to a different file system.

Figure 4–4 Distributed Directory Structure



Localization Directories

For product files, each localization region is treated like a separate product. Module files are stored in the region directory trees. Figure 4–5 illustrates the directory tree for European localizations (je). For this directory tree, AutoInstall sets the variable JE_TOP to \$APPL_TOP/je/11.0.28

Figure 4-5 Localization Directory Structure

Like all other Oracle Applications products, localizations are also translated into various languages. These translated files follow the standard language directory structure.

Language Directory Structure

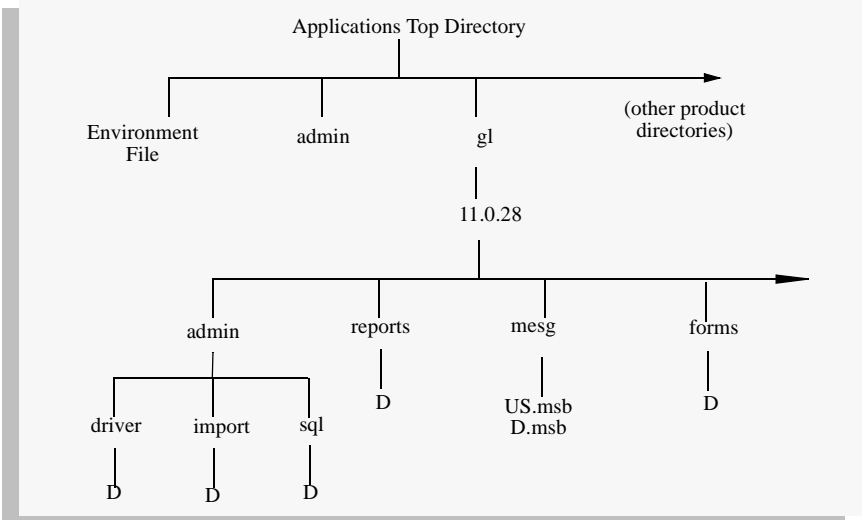
When you install Oracle Applications in a language other than American English, each product tree includes directories that use the NLS language code. These directories hold translated form, report, message, SQL scripts, and installation files. The language directory in Figure 4-6 is D, which designates German.

D subdirectories in the driver, import, and sql subdirectories of the admin subdirectory hold files to convert the database from American English to German. The reports subdirectory holds Oracle Reports report files. Its D subdirectory contains these same files translated into German.

The forms directory shown in Figure 4-6 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.

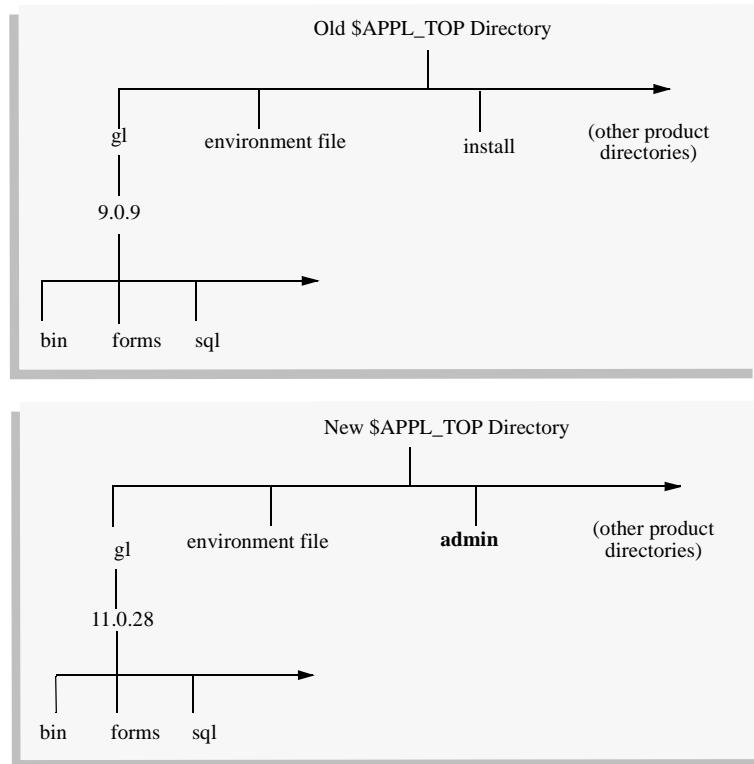
Additional Information: Appendix A, *Oracle Applications Installation*

Figure 4-6 Language Directory Structure



Multiple Applications Top Directories

You must create a new APPL_TOP directory when you upgrade. You cannot reuse an existing APPL_TOP because the existing installation directory and environment file are overwritten. AutoInstall does not delete the existing product files. Instead, it unloads new product files in a new version directory in each product's directory tree. You cannot have multiple releases and product versions in one APPL_TOP directory. Figure 4-7 illustrates the directory structure after an upgrade that uses a new APPL_TOP directory. New components are shown in **boldface** text.

Figure 4–7 Multiple Product Versions Installed on One File System

Attention: Because upgrading changes the Oracle Applications database structure, you cannot use old product files with an upgraded database. You can delete them once you are certain the upgrade is complete and successful.

Product Files for Multiple Product Groups

For many product groups, such as demonstration and test product groups, you can use the same set of Oracle Applications product files. Follow these guidelines if you use the same product files for multiple product groups:

- Your production database must use a separate set of Oracle Applications product files from all other product groups.

This is required to ensure that patches applied to one product group, such as a demonstration or test product group, are kept separate from your production system.

- Apply database patches to every product group.

If you apply a database patch to one group, be sure to apply it to every other product group.

- Use the same Oracle8 Server and tools versions for all databases.

When you install the first product group, you link Oracle Applications product files with the Oracle8 Server libraries. These product files may not function with a database that uses a different version of the Oracle8 Server libraries.

- Use the same ORACLE_HOME for both Oracle8 Server and tools.

Because Oracle Applications product files link with the Oracle8 Server libraries, the server and tools software should be installed in the same ORACLE_HOME.

- Maintain a separate environment file for each group.

AutoInstall and adadmin use the database name when creating environment files. Therefore, if you keep the default environment file names, you will have separate environment files for each product group.

- Keep log and output files for each product group separated.

Each product group needs a separate directory area for its log and output files. There are several methods for setting this up, as explained in this chapter.

Log and Output Files

When the concurrent managers run Oracle Applications reports, they write the output files, as well as diagnostic log files and temporary files, to directories that you define during the installation process. This section explains your options for setting up these directories. Note the following:

- Concurrent managers inherit directory privileges from the login that starts them. Since you start the managers from the main Oracle Applications login, applmgr, they can write to any directory accessible to this login.

- Log files for the concurrent managers themselves are stored in \$APPLCSF/\$APPLLOG if APPLSCF is set, and in \$FND_TOP/\$APPLLOG if APPLCSF is not set.

Additional Information: Appendix A, *Oracle Applications Installation*

You should be familiar with the basics of concurrent processing before you install Oracle Applications.

Additional Information: Overview of Concurrent Processing, *Oracle Applications System Administration Reference Manual*

Log and Output Files for a Single Product Group

When you install a single Oracle Applications product group, you can choose one of two methods for storing the group's log and output files:

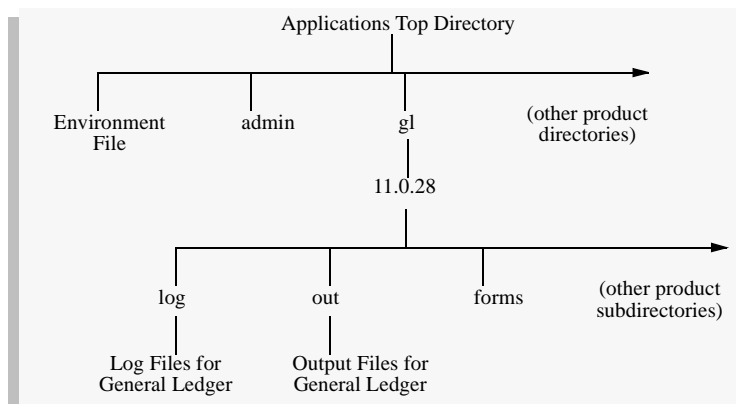
- Log and output subdirectories in each product's directory tree (default).
- One log and one output subdirectory in a common directory.

The following sections explain these methods in detail. *Oracle Applications Installation* guides you through setting up a common directory.

Default Storage in Product Directory Tree

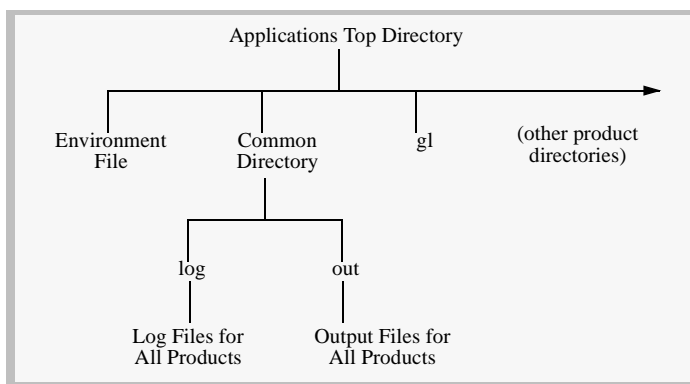
Figure 4-8 illustrates the default storage method that uses log and out subdirectories in each product's directory tree. AutoInstall sets up this storage method automatically unless you specify otherwise.

By default, the product group's environment file sets the variable APPLLOG to *log* and the variable APPLOUT to *out*. Concurrent managers write to these subdirectories in each product tree. For Oracle General Ledger, for example, the concurrent managers write log files to \$GL_TOP/\$APPLLOG and output files to \$GL_TOP/\$APPLOUT.

Figure 4–8 Default Storage for Log and Output Files**Common Directory**

Instead of using the default storage method, you can set up a common directory that has a subdirectory for all log files and subdirectory for all output files. The common directory can reside on any file system accessible to the concurrent managers. Figure 4–9 illustrates a common directory in the APPL_TOP directory.

You designate the common directory and the subdirectories when you run AutoInstall or by re-creating the environment file using adadmin. The environment file sets the variable APPLCSF to the common directory and the variables APPLLOG and APPOUT to the log and output subdirectories, respectively. The concurrent managers write all log files to \$APPLCSF/\$APPLLOG and all output files to \$APPLCSF/\$APPOUT.

Figure 4–9 Common Directory for Log and Output Files

The following sections explain these methods in detail. Later sections in this manual guide you through setting up the directories for the storage method you choose.

Additional Information: Selecting a Log and Output File Storage Method in this chapter, and Directory Requirements in Chapter 3.

Log and Output Files for Multiple Product Groups

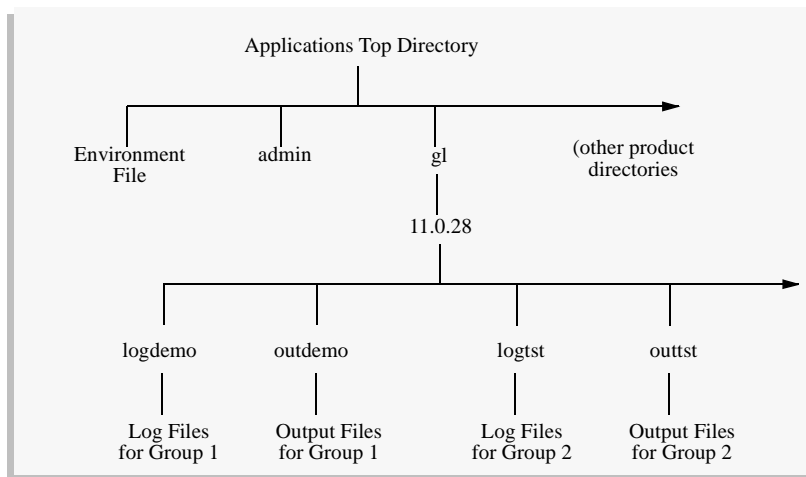
If you install multiple product groups in the same APPL_TOP directory, each one needs its own log and output directories to keep its log and output files separate from those of other groups. There are three methods for creating separate log and output areas:

- Multiple log and output subdirectories in each product's directory tree.
- Multiple log and output subdirectories in one common directory.
- Multiple common directories.

Log and Output Directories in Each Product's Directory Tree

The default storage method depicted in Figure 4-10 uses multiple log and output subdirectories in each product tree. The logdemo and outdemo subdirectories are for log and output files from a demonstration product group. The logtst and outtst subdirectories are for a test product group. You select the subdirectory names when you run AutoInstall or adadmin, but you must manually create the subdirectories if you name them something other than *log* or *out*.

Each product group's environment file defines the group's log and output subdirectories as the values of the variables \$APPLLOG and \$APPLOUT, respectively. A product group's concurrent managers write log and output files to \$APPLLOG and \$APPLOUT in each product tree. For Oracle General Ledger, for example, the managers write log files to \$GL_TOP/\$APPLLOG and output files to \$GL_TOP/\$APPLOUT.

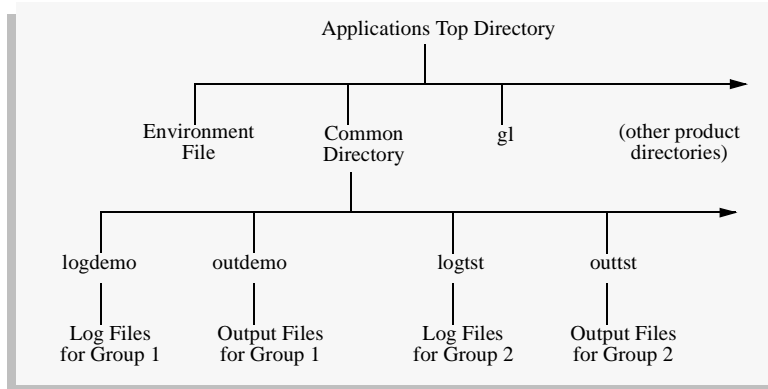
Figure 4–10 Default Log and Output File Storage for Multiple Product Groups

Single Common Directory

As Figure 4–11 shows, a single common directory should use separate log and output subdirectories for each product group. The common directory can reside on any file system accessible to the concurrent managers. You designate a group's common directory and its log and output subdirectories when you run AutoInstall or adadmin for each product group.

All product group environment files define the common directory as the value of the variable APPLCSF. Each file defines different log and output subdirectories for as the values of the variables APPLLOG and APPOUT, however. The concurrent managers for each product group write the group's log and output files to \$APPLCSF/\$APPLLOG and \$APPLCSF/\$APPOUT, respectively.

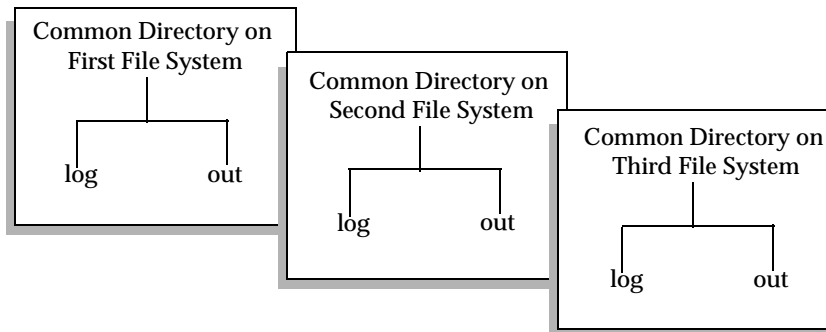
Figure 4–11 Common Directory for Multiple Product Groups



Multiple Common Directories

Figure 4–12 illustrates three common directories set up for three product groups on three separate file systems. You designate a product group’s common directory and the log and output subdirectories when you run AutoInstall or adadmin.

Figure 4–12 Multiple Common Directories for Multiple Product Groups



Each product group’s environment file defines the log and output subdirectories (shown below as log and out in each case) as the values of \$APPLLOG and \$APPLOUT, respectively. Each file defines a different common directory as the value of \$APPLCSF. A group’s concurrent managers write log files to \$APPLCSF/\$APPLLOG and output files to \$APPLCSF/\$APPLOUT.

Selecting a Log and Output File Storage Method

Consider the following when selecting a method for storing log and output files:

- Oracle Applications configuration

Parallel concurrent processing systems should store log and output files on each server rather than have one log and output area for all servers.

- anticipated size of log and output files

Oracle Applications Installation contains information that helps you estimate how much disk space you need for log and output files. You may want to store files across file systems if you plan to keep a large volume of log and output data online.

- security needs

With the default storage method, you can restrict operating system access to log and output directories on a product-by-product basis. For example, you can grant access to the Oracle General Ledger log and output directories only to a certain group of users.

Temporary Files

You can store a product group's temporary files in one or two directories that you create. This lets you avoid space limitations on the default temporary directory (/usr/tmp or /tmp) and restrict access to temporary files. Your temporary directory or directories can reside on any file system accessible to the concurrent managers.

Applications uses temporary directories for many different functions. You may specify up to three separate locations for these files, according to how they are produced. You designate all three destinations when you run AutoInstall or adadmin to create a new environment file.

Most Applications temporary files are written to the location specified by the environment variable APPLTMP. Oracle Reports temporary files can be directed to a separate location according to the variable REPORTS25_TMP. Both of these variables can be set to any directory on your file system. The defaults are /tmp and /usr/tmp, respectively.

Applications also produces temporary PL/SQL output files used in concurrent processing. These files are written to a location specified by the variable APPLPTMP. APPLPTMP can also be located anywhere on your filesystem, but is more restricted because it uses PL/SQL I/O functionality. For security reasons, in order to direct output to a directory using this functionality, you must list the

directory in your database initialization file, in the value for the `utl_file_dir` parameter. Without this setting, Applications will fail to install or function properly. For this reason, the `APPLPTMP` variable has no default setting.

Attention: Some Applications utilities will use the default temporary directory for your operating system (typically `/tmp` or `/usr/tmp`), even if you set the environment variables listed in the previous paragraph to different values. Be sure to have available disk space for these default directories as well as those denoted by `APPLTMP`, `REPORTS25_TMP`, and `APPLPTMP`.

Index

Symbols

\$APPLLOG
 overview, 4-9
\$APPLOUT
 overview, 4-9
_ALL, 3-8

A

AD Configuration Utility, x
admin directory
 overview, 4-2
administration server
 in Release 11, 1-8
APPL_TOP, 4-1
application server
 defined, 1-4
 in Release 11, 1-4
application tier
 for Release 11, 1-4
Applications data
 maintaining, 1-9
Applications database updates
 applying, 1-9
Applications server
 in Release 10, 1-4
APPS schema
 and base product schemas, 3-3
 benefits of using, 3-4
 custom schema access, 3-6
 data access, 3-6
 how AutoInstall creates, 3-5
 overview, 3-3

attachments
 using, 1-14

C

cartridge, 1-6
certificate file
 definition of, 1-4
character set
 for Oracle8 Server, 3-20
Client/Server
 common log and output directory, 4-11
CLIENT_INFO, 3-8
common directories
 for multiple product groups, 4-14
 illustration, 4-14
 multiple, 4-14
common directory
 single, 4-13
concurrent parallel processing
 overview, 1-9
concurrent processing
 how it works, 1-9
concurrent processing server
 in Release 11, 1-9
 interaction with forms servers, 1-11
customizations
 custom schema access, 3-6
 data access, 3-6

D

data files
 for Oracle8 Server, 3-20

- database
 - installing and upgrading, 1-9
- database server
 - in Release 11, 1-8
- database tier
 - for Release 11, 1-7
- DateTrack, 2-5
- DB_BLOCK_SIZE parameter, 3-21
- default storage
 - in product directory tree, 4-10
- dependent products
 - about, 2-7
- desktop client
 - defined, 1-2
- desktop client tier
 - for Release 11, 1-2
- Developer/2000
 - version, 2-10
- digital signature
 - definition of, 1-4
- directories
 - across file systems, 4-5
 - common log and output, 4-11
 - languages, 4-6
 - localizations, 4-5
 - multiple version, 4-7
 - temporary, 4-15
- directory structure
 - basic, 4-1
 - illustration of basic, 4-2
- distributed directory structure
 - illustration of, 4-5
 - overview, 4-5

E

- environment file
 - overview, 4-2

F

- FastFormula, 2-5
- forms client applet
 - how it works, 1-11
- forms server

- defined, 1-5
- optimizing, 1-11

full product installation requirements for, 2-6

G

- grants
 - between user IDs, 3-2

H

- help
 - context-sensitive, 1-15
- HRMS product family
 - about, 2-5
- HTML directory
 - illustration of, 4-4
 - overview, 4-3
- HTTP listener, 1-6

I

- initialization parameters
 - for Oracle8 Server, 3-21
- interim tablespace, 3-23
- Internal monitors
 - how they work, 1-10

J

- JAR file
 - definition of, 1-3
- Java archive (JAR) files
 - downloading, 1-13
- Java Developer's Kit
 - version, 2-10
- Java directory
 - illustration of, 4-4
 - overview, 4-3
- JDBC
 - version, 2-10

L

- language directory structure

- illustration of, 4-7
- language installation
 - about, 2-8
- load balancing
 - among forms servers, 1-5
 - using, 1-13
- localization directory structure
 - illustration of, 4-6
- localizations
 - about, 2-8
 - database objects included, 3-6
 - directory structure, 4-5
- log and output directories
 - in a product directory tree, 4-12
 - single common directory, 4-13
- log and output files
 - default storage
 - for multiple product groups, 4-13
 - illustration of, 4-11
 - for single product group, 4-10
 - selecting storage for, 4-15
- log files
 - managing, 4-9

M

- MAXDATAFILES parameter, 3-20
- MAXEXTENTS, 3-23
- Metrics Server
 - definition of, 1-5
- MOA (Multiple Oracle Account), 3-16
- MRC
 - illustration of architecture, 3-13
 - illustration of setup, 3-11
 - overview of architecture, 3-11
 - See Multiple Reporting Currencies, 3-9
 - setting up primary responsibility for, 3-10
 - synonyms, 3-16
 - transaction table/sub-table illustration, 3-14
 - triggers, 3-14
 - using in GL, 3-16
 - views, 3-14
 - in Oracle Assets and Projects, 3-14
 - in Oracle Payables, Receivables, and Purchasing, 3-15

- Multiple Oracle Account (MOA), 3-16
- Multiple Organization Architecture
 - _ALL, 3-8
 - changes from Release 10.6, 3-9
 - CLIENT_INFO, 3-8
 - ORG_ID, 3-8
 - overview, 3-7
 - partitioned objects, 3-8
 - patching, 3-9
 - SO_HEADERS_ALL, 3-8
- multiple product groups
 - log and output file management, 4-12
- multiple product installation groups
 - illustration of, 3-19
- multiple product versions
 - installed on one file system, 4-8
- Multiple Reporting Currencies
 - overview, 1-13, 3-9
- multiple sets of books
 - illustration of architecture, 3-18
 - overview, 3-16
- multi-tier architecture
 - how it works, 1-2

N

- Network latency
 - minimizing, 1-12

O

- operating unit
 - definition of, 3-8
- Oracle Alert
 - description of, 2-6
 - mail products used with, 2-5
- Oracle Application Object Library
 - description of, 2-6
- Oracle Applications DBA
 - description of, 2-6
- Oracle Applications Utilities
 - description of, 2-6
- Oracle Consulting Services, x
- Oracle Forms
 - version, 2-10

- Oracle Forms server engine, 1-5
- Oracle General Ledger
 - interim tablespace, 3-23
- Oracle InterOffice, 2-5
- Oracle Manufacturing product family
 - about, 2-5
- Oracle Net8
 - version, 2-10
- Oracle Projects product family
 - about, 2-5
- Oracle Reports
 - version, 2-10
- Oracle Self-Service Web Applications and Web Server, 1-7
- Oracle Support Services, x
- ORACLE user ID
 - localizations, 3-6
 - overview, 3-2
- Oracle Web Application Server
 - version, 2-10
- Oracle8 ConText Cartridge
 - version, 2-10
- Oracle8 Server
 - data files for, 3-20
 - database character set, 3-20
 - database creation, 3-20
 - initialization parameters for, 3-21
 - overview, 3-19
 - requirements for, 2-9
 - tablespaces, 3-21
 - version, 2-10
- ORG_ID, 3-8
- output files
 - managing, 4-9

P

- parallel concurrent processing
 - how it works, 1-10
- partitioned objects
 - in Multi-Org, 3-8
- pinning stored packages, 1-13
- PL/SQL
 - version, 2-10
- primary functional currency

- definition of, 3-9
- primary responsibility
 - setting up for MRC, 3-10
- primary server
 - definition of, 1-10
- primary set of books
 - definition of, 3-10
- Pro*C
 - version, 2-10
- product families
 - groupings, 2-3
 - list of, 2-4
- product files
 - for multiple product groups, 4-8
- product groups
 - in databases, 3-1
 - multiple in separate databases, 3-2
- product installation groups and Multiple Sets of Books, 3-18
 - illustration of multiple, 3-19
 - multiple, 3-18
- product version numbers
 - list of, 2-1
- products
 - required for installation, 2-5
 - required full installations, 2-6

R

- Report Review Agent, 1-15
- reporting functional currency
 - definition of, 3-10
- reporting set of books
 - definition of, 3-10
- rollback tablespace, 3-22

S

- secondary server
 - definition of, 1-10
- security
 - how it works, 1-3
- set of books
 - definition of, 3-7
- short names

- products, 2-1
- Single Oracle Account (SOA), 3-16
- single product installation group
 - illustration of, 3-8
- sizing factor
 - 100%, 3-24
 - greater than 100%, 3-24
 - less than 100%, 3-24
 - overview, 3-23
 - recommended, 3-25
 - sample extent sizes, 3-24
- SO_HEADERS_ALL, 3-8
- spreadsheets
 - exporting, 1-14
- SQL*Loader
 - version, 2-10
- SQL*Plus
 - version, 2-10
- SYSTEM tablespace, 3-21

T

- tablespaces
 - setting up for products, 3-22
- temporary files
 - storage methods, 4-15
- temporary tablespaces, 3-23
- tools tablespace, 3-21
- TOP directories
 - multiple, 4-7
- TOP variables
 - product
 - overview, 4-2
 - products
 - list of, 2-1

U

- UNIX operating system
 - using for AutoInstall, 2-9
- users tablespace, 3-22

W

- web server, 1-6

