

Oracle® Applications

Oracle Applications Installation

Release 11 for UNIX

July 1998

Part No. A57983-02

Oracle Applications Installation, Release 11 for UNIX

Part No. A57983-02

Copyright © 1998, Oracle Corporation. All rights reserved.

Authors: Michael Fiore, Keith M. Swartz

Contributors: Steve Carter, Gail D'Aloisio, Carole Eubanks, Ashok Handigol, Rick Lotero, Jeff Lunn, Michael Mast, K. Narayanan, Kent Noble, Vesa Peltola, Iru Puri, Greg Seiden, Robin Varnes, Sara Woodhull

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. Customer 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 Use 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 Schmoracle, 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, and SQL*Report, and Web Agent are trademarks or registered trademarks of Oracle Corporation.

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

Contents

Preface	ix
Audience.....	ix
Structure	x
Related Documents	xi
Getting Help	xii
Oracle Consulting Services	xii
Oracle Support Services.....	xiii
 Send Us Your Comments.....	 xv
 1 Requirements	
Your Place in the Installation or Upgrade Process	1-1
CPU Requirements	1-1
Memory Requirements	1-3
Disk Space Requirements	1-5
Dependent Products.....	1-6
Product File Sizes.....	1-10
Additional Disk Space Requirements.....	1-14
Verifying Disk Space.....	1-17
Oracle8 Server and Oracle Tools Requirements	1-18
Oracle Product Versions	1-18
Initialization Parameters.....	1-21
SQL*Plus Options.....	1-23
Tablespace Requirements.....	1-24

SYSTEM Tablespace	1-24
Rollback Segments	1-25
Temporary Tablespace	1-25
Product Table and Index Tablespaces	1-26

2 Preparing to Install or Upgrade

Your Place in the Installation or Upgrade Process	2-2
Checklist for This Chapter	2-3
Set Up the Technology Stack	2-5
Create the Applications Login Account	2-9
Create Directories	2-12
Prepare Your Database	2-14
Set Environment Variables	2-17
Unload AutoInstall Files	2-19
Prepare for Upgrade	2-24
Set Up AutoInstall Environment	2-30

3 Using AutoInstall

Your Place in the Installation or Upgrade Process	3-2
Running AutoInstall (adaimg)	3-2
Stopping AutoInstall	3-3
Restarting AutoInstall	3-3
Running AutoInstall in a Multi-Tier Environment	3-5
AutoInstall Questions	3-5
Applications Top Directory	3-6
AutoInstall Log File	3-6
Software Compatibility	3-6
Email Notification	3-6
Batch Size	3-6
Installing Files in APPL_TOP	3-7
ORACLE Database	3-7
AutoInstall Action	3-8
Temporary Tablespace	3-8
Organization Type	3-9
Territory and Language	3-10

Using AutoInstall Menus and Screens	3-10
Main Menu	3-11
About the AutoInstall Installation Screens	3-12
Select Products Screen	3-13
Product Detail Screens	3-17
Select Modules Screen.....	3-22
Database Parameters Screen	3-23
Tasks Screen	3-27
Running the Installation or Upgrade.....	3-34
Creating an Environment File	3-37
Monitoring AutoInstall	3-40
Manager and Worker Messages.....	3-40
Log and Restart Files.....	3-41
Correcting AutoInstall Errors	3-43
Acceptable Errors	3-45
After Running AutoInstall.....	3-49

4 Finishing Your Installation or Upgrade

Your Place in the Installation or Upgrade Process	4-2
Checklist for This Chapter.....	4-3
Review AutoInstall Log Files	4-5
Set Oracle8 Server Parameters.....	4-6
Configure the Web Server	4-9
Configure the Forms Server.....	4-14
Set Up Concurrent Managers	4-20
Install Parallel Concurrent Processing	4-26
Set Up the Report Review Agent.....	4-29
Start Oracle Applications	4-31
Complete Post-Upgrade Steps.....	4-33
Re-Enable and Integrate Customizations	4-34
Tasks To Do Any Time After Installing or Upgrading	4-36

5 Installation Utilities

AD Controller Utility (adctrl).....	5-1
Running the AD Controller Utility.....	5-2

Log File	5-3
Reviewing Worker Status	5-3
Restarting a Failed Worker	5-4
Restarting a Terminated Worker	5-5
Reactivating the Manager	5-5
Shutting Down AutoInstall	5-6
Controlling the Number of Workers.....	5-6
AD Administration Utility (adadmin)	5-7
Running the Administration Utility.....	5-7
Log File	5-8
AD Administration Questions	5-8
Maintaining Applications Database Objects.....	5-11
Maintaining Applications Files.....	5-16
AutoPatch Utility (adpatch)	5-20
How AutoPatch Works.....	5-20
Patch Format.....	5-21
Running AutoPatch	5-23
AutoPatch Questions.....	5-25
AutoPatch Messages.....	5-28
After Running AutoPatch.....	5-30
Running AutoPatch Again	5-33
Controlling AutoPatch Behavior	5-35
Patch Merge Utility.....	5-38
Patch Driver Files.....	5-39
Patch History File.....	5-42
Unload Utility (adunload)	5-44
Log File	5-44
Running the Unload Utility.....	5-44
Relink Utility (adrelink)	5-48
Log File	5-48
Running adrelink	5-48
Relinking with Debug	5-51
Target File	5-51
Backup Options for adrelink.....	5-52
Character Set Conversion Utility (adncnv)	5-53

ODF Comparison Utility (adodfcmp)	5-54
Running the ODF Comparison Utility	5-55
AD File Identification Utility (adident)	5-60
AD Configuration Utility (adutconf.sql)	5-61

A System Reference Material

Install Multiple Reporting Currencies	A-2
Integrating Custom Objects and Schemas	A-7
Concurrent Managers	A-9
Starting the Concurrent Managers	A-10
Restarting the Concurrent Managers	A-12
File Conventions	A-13
Directory Privileges	A-14
Printing	A-15
Standard Print Subroutine	A-15
Executable Printing Programs	A-16
Analyzing the Database	A-20
Pinning Packages and Sequences	A-21
Changing Oracle Applications Passwords	A-22
Migrating the Oracle Applications Database	A-24
Preparing to Migrate the Database	A-25
Export the Oracle Database	A-26
Import the Oracle Database	A-27
Finishing the Migration	A-28
Migrating the Oracle Applications Filesystem	A-32
Moving Files On the Same Machine	A-32
Preparing to Migrate the Filesystem to Another Machine	A-33
Unload Oracle Applications Product Files	A-34
Completing the Migration	A-35
Recovering Disk Space	A-36
Product Directory Structure and File Types	A-37
Environment Variables	A-39
Territory Names	A-42
Language Codes and Character Sets	A-45
Reserved Words	A-46

B Oracle Application Object Library Reference Material

Development Environment	B-1
Create Directory Structure.....	B-1
TOP Environment Variable	B-3
Message Dictionary	B-5
Concurrent Processing	B-6
Log and Output Filenames.....	B-6
Oracle Tool Concurrent Programs	B-7
Pro*C Concurrent Programs	B-8
Host Language Concurrent Programs	B-11
Submitting Concurrent Requests (CONCSUB)	B-12
Syntax	B-13
Example.....	B-13
Parameters	B-14

Index

Preface

Oracle Applications Installation provides instructions for installing Oracle Applications products and explains the utilities you use for installation and maintenance.

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. These people typically use *Concepts* and *Installation*:

- Database Administrator
Installs and configures the Oracle database and maintains database access controls. the person provides consultation on performance, and is responsible for monitoring growth and fragmentation of the production database and ensuring database backup and recovery.
- System Administrator
Responsible for administering the development system. This person's responsibilities include:
 - Ensuring that hardware is correctly configured
 - Installing, configuring, and maintaining operating and development software
 - Ensuring that the system is backed up daily
 - Designing and maintaining system security — for example, establishing system accounts.

The system administrator provides first-line support for problems with the development system and ensures that faults are quickly rectified. This person may perform the setup and initial maintenance of the production system or advise the client's operational staff on these tasks. the system administrator works with the project team to optimize system performance. They also install packaged applications environments and convert data.

- Technical Specialist

Responsible for designing, developing, unit testing, implementing, and maintaining the custom extensions for Oracle Applications. These extensions include, but are not limited to, modules such as interfaces, automated data conversions, reports, forms, and enhancements.

Structure

Oracle Applications Installation contains these chapters and appendixes:

Chapter 1	Lists the requirements, such as memory and disk space, needed for an installation or upgrade to Release 11.
Chapter 2	Explains how to prepare your system before running AutoInstall
Chapter 3	Tells how to run AutoInstall, the program that installs or upgrades an Oracle Applications database.
Chapter 4	Explains the steps to perform after you run AutoInstall.
Chapter 5	Documents Oracle Applications utilities you may use before, during, or after your installation or upgrade.
Appendix A	Contains reference material that helps you maintain an Oracle Applications system.
Appendix B	Supplements the <i>Oracle Application Object Library Reference Manual</i> . It provides information that is specific to the operating system.

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, located on the World Wide Web at <http://www.oracle.com/documentation/sales/index.html>.

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

Oracle Applications Concepts explains the tiers in Release 11 and possible ways to deploy this release in a mixed-platform topology.

In addition to *Concepts* and *Installation*, you should refer to the following documents to plan and run your installation or upgrade:

- Oracle Applications Release 11 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 this installation manual.
- Oracle Applications Upgrade Manual for UNIX
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 16. When upgrading, use the *Product Update Notes* to learn about new features in Release 11.
- Oracle Applications Technical Reference manuals
Lists information about database structures and helps determine what changes you need to make to customizations after an upgrade.

Attention: Although this installation manual refers to database changes that may affect customizations, it does not provide a complete account of all changes. Refer to the Oracle Applications technical reference manuals for full information about database structures.

- 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, 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 upgrading Oracle Applications

- implement Oracle Applications products
- customize Oracle Applications products
- install and configure multiple language support
- develop custom applications for use with Oracle Applications
- train users of Oracle Applications

Oracle Support Services

The Oracle Support Services Web site at <http://support.oracle.com/metalink/> offers registered *Oracle MetaLink* customers access to online technical libraries, installation assistance, product reference, certification information, and other support services.

If you contact Oracle Support Services for other assistance, have this information available:

- your CSI number
- the operating system and version
- 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 in Chapter 3

- the output of the AD Configuration Utility, contained in the adutconf.lst file

Additional Information: AD Configuration Utility (adutconf.sql) in Chapter 5

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 for UNIX command prompt. Your prompt may differ.
\	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 Installation, Release 11 for UNIX

Part No. A57983-02

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
Redwood Shores, CA 94065 U.S.A

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

Requirements

This chapter covers the requirements for installing or upgrading Oracle Applications and contains the following sections:

- [CPU Requirements](#)
- [Memory Requirements](#)
- [Disk Space Requirements](#)
- [Oracle8 Server and Oracle Tools Requirements](#)
- [Tablespace Requirements](#)

Your Place in the Installation or Upgrade Process

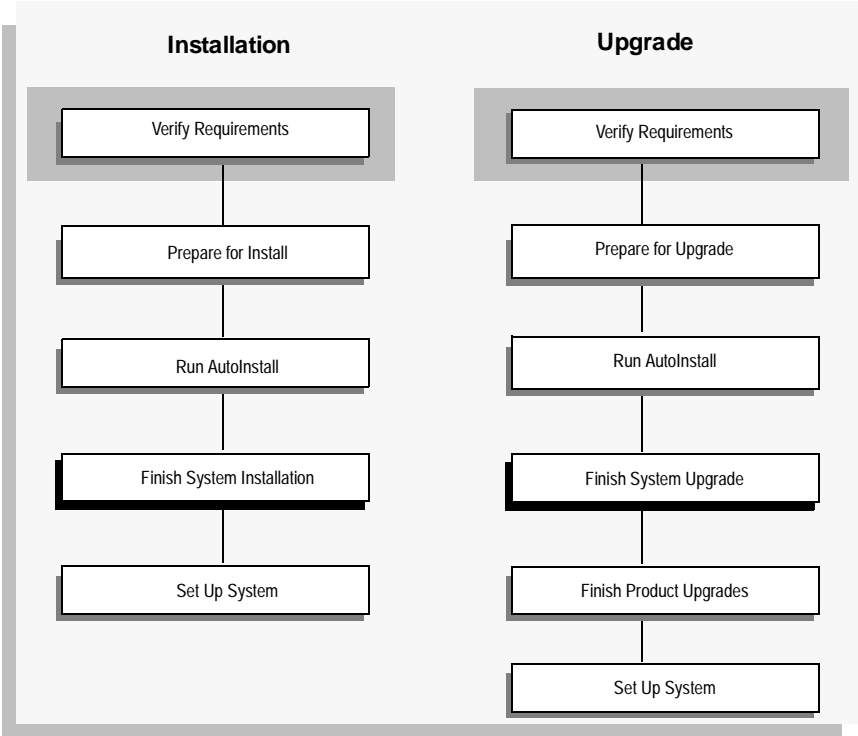
The following figure shows your current place in the Oracle Applications installation or upgrade process. This chapter explains the highlighted steps in this figure.

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

CPU Requirements

The CPU requirements for running Oracle Applications depend on many factors. The number of users and their usage profile is important, as are the number of concurrent manager processes and the types of jobs that they are running. You must also account for the load on the machine for activities besides Oracle Applications.

In general, an Oracle Order Entry/Shipping user will require more CPU than a General Ledger user. However, a General Ledger user running financial statements for a year end close may require more CPU than an Order Entry/Shipping user querying orders. Similarly, running a large batch load during the evening when there are few online users does not create the same peak requirements as running those same jobs during the day.



There are different product combinations, different user profiles, and different machine configurations. There is no one right answer for machine sizing that can be applied to all hardware platforms. The most reliable way to ensure that your hardware is sized appropriately is to run your own benchmark with a machine configuration, product mix and user load that simulates your own. If that is not feasible, find another Oracle Applications installation running a product mix and user profile similar to yours on your target platform. Oracle Consulting Services and your hardware vendor can help you in this search. Some hardware vendors have sizing worksheets that model the CPU and memory requirements of Oracle Applications on their hardware.

Memory Requirements

The following guide for estimating memory needs for Oracle Applications on UNIX machines does not include a precise method for calculating every process requirement on your system. The amount of memory you need for an Oracle Applications system largely depends on your platform. Contact your hardware vendor or Oracle Consulting Services to estimate how much memory is required to run Oracle Applications with good response time.

1. Estimate Oracle8 Server overhead

Your hardware vendor or an Oracle Applications consultant can help determine the amount of memory you need for ORACLE overhead. For forms servers, you do not need to calculate memory requirements for ORACLE processes or concurrent manager processes.

2. Determine size of system global areas

Each Oracle Server database requires memory for the ORACLE processes. The System Global Area (SGA) is a shared memory region of data and control information for an ORACLE database. If you have multiple databases on your server, each database has its own SGA. Because the parameter `SHARED_POOL_SIZE` affects the SGA, determine an SGA size after you set this parameter to the recommended value.

Additional Information: [Initialization Parameters](#) in this chapter

The SGA size displays on the terminal screen when the database administrator starts the database or uses the Oracle Server Manager command `SHOW SGA`.

Additional Information: *Tuning Memory Allocation, Oracle8 Tuning*

3. Estimate the number of users

The number of users running Oracle Applications simultaneously affects memory requirements.

Suggestion: Select a higher number of users if you have frequent periods in which more than the average number of people use Oracle Applications simultaneously. This helps to ensure adequate response time during heavy user demand.

4. Estimate the number of concurrent manager processes

When you set up concurrent managers, you determine a maximum number of jobs that each manager can run (the *target processes*) in a work shift. Estimate how many manager target processes you will define for the busiest work shift. Double this number to account for the spawned processes (the *programs*) that the managers run.

For a parallel concurrent processing server, take into account the number of users and the number of managers defined as primary or secondary to the server. Add the Oracle8 Server overhead and System Global Area only if the server runs an ORACLE instance.

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

For example, [Figure 1-1](#) shows 12 target processes defined for four concurrent managers. In this case, you need to account for 25 processes:

- 1 process for the Internal Manager
- 12 target processes
- 12 spawned processes (programs being run)

You should also allow for additional processes that are not run directly by the concurrent managers, but are run on the concurrent processing servers. This would include, for example, the Conflict Resolution Manager, and the retrieval portion of the Report Review Agent (FNDFS).

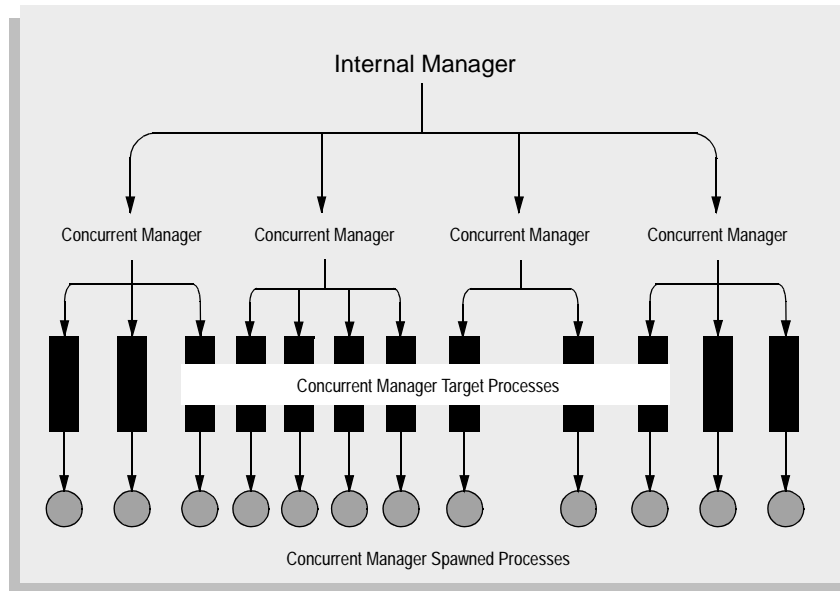
Figure 1-1 *Concurrent Processing Example*

If you plan to install multiple Oracle Applications product groups, you must account for each group's concurrent processing requirements.

Attention: Oracle Applications reserves memory at all times for each active manager target process, regardless of whether it is running a job. Defining more target processes than necessary therefore wastes memory.

Estimate Other Memory Requirements

Your memory estimate must account for processes other than those directly related to Oracle Applications. These processes include the following:



- Operating system overhead
- File system management
- Other applications that you use

Disk Space Requirements

This section explains how to calculate the disk space needed for Oracle Applications products. By using the instructions, you can produce an estimate of disk space required on a concurrent processing server or a forms server.

1. Determine fully-installed and dependent products

Mark the products that you have licensed in the Full column of [Table 1-1](#). In addition, for each of your licensed products, determine the additional products that must be installed fully and mark those in the Full column, if they are not already marked.

Next, use the Additional Dependent Products column to determine the dependent products for each product you have marked in the Full column. Mark the dependent products in the Dep column. Do not mark a product as dependent if you have already marked it as fully installed.

2. Mark installed and dependent-only products

Mark each product you will install fully in the concurrent processing server, forms server, or web server column of [Table 1-2](#). Note that on an administration server, all files needed for administration are installed regardless on license status. The required products are already marked in this table.

[Table 1-2](#) also provides file sizes for products such as DateTrack that are installed only as dependent products. If your fully installed products require dependent-only products, mark the dependent-only products in the appropriate column of [Table 1-2](#). Calculate a subtotal for installed and dependent-only products at the bottom of [Table 1-2](#).

3. List number of remaining dependent products

Determine the number of remaining dependent products. These are the products that can be installed fully, but which you will install as dependent products. Mark this number in the appropriate column at the bottom of [Table 1-2](#).

4. Add product file disk space requirements

At the bottom of [Table 1-2](#), mark the total requirement for installed, dependent-only, and dependent products. This is the space estimate in Megabytes for Oracle Applications product files. See the sections after [Table 1-2](#) for information about variations in product file size and disk space requirements for other files used on your system.

Dependent Products

Use [Table 1-1](#) to determine which products you will install fully or as dependent products. In the Full and Dep columns, ✓ means a required installation, ✗ means the product cannot be installed (as a full or dependent product), and □ means the product can be installed (either as a full or dependent product). For example, ✗ in the Dep column means the product cannot be installed as a dependent product, ✓ in the Full column means the product is always fully installed, and □ in the Dep column means the product may be installed as a dependent product.

Table 1–1 Dependent Products

Full	Dep		Licensed Product	Additional Fully Installed Products	Additional Dependent Products
X	✓	AD	Oracle Applications DBA	FND	ALR
<input type="checkbox"/>	X	AK	Oracle Common Modules	FND	
<input type="checkbox"/>	✓	ALR	Oracle Alert	FND	
<input type="checkbox"/>	<input type="checkbox"/>	AP	Oracle Payables or Oracle Public Sector Payables	AK, AX, AZ, FND	AR, AS, BOM, ENG, FA, GL, INV, OE, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	<input type="checkbox"/>	AR	Oracle Receivables or Oracle Public Sector Receivables	AK, AX, AZ, FND	AP, AS, BOM, CRP, ENG, FA, GL, INV, MRP, OE, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	<input type="checkbox"/>	AS	Oracle Sales and Marketing	AK, AX, AZ, FND	AP, AR, CS, ENG, GL, INV, OE, PA, PAY, PER, PO, WIP
X	✓	AU	Oracle Application Utilities	FND	
<input type="checkbox"/>	X	AX	Global Accounting Engine	FND	
<input type="checkbox"/>	X	AZ	Oracle Applications Implementation	FND	
<input type="checkbox"/>	<input type="checkbox"/>	BOM	Oracle Bills of Material (includes Cost Management)	AK, AX, AZ, FND, INV	AP, AR, AS, CRP, ENG, FA, GL, MRP, OE, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	X	CE	Oracle Cash Management	AK, AX, AZ, FND	AP, AR, AS, ENG, GL, INV, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	X	CHV	Oracle Supplier Scheduling (installed with PO)	AK, AX, AZ, FND, PO	AP, AR, AS, BOM, CRP, ENG, FA, GL, INV, MRP, OE, PA, PAY, PER, WIP
<input type="checkbox"/>	X	CN	Oracle Sales Compensation	AK, AX, AZ, FND, GL	AP, AR, INV, PAY, PER

Table 1–1 Dependent Products

Full	Dep		Licensed Product	Additional Fully Installed Products	Additional Dependent Products
<input type="checkbox"/>	<input type="checkbox"/>	CRP	Oracle Capacity	AK, AX, AZ, BOM, FND, INV, MRP	AP, AR, AS, ENG, FA, GL, OE, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	<input type="checkbox"/>	CS	Oracle Service	AK, AX, AZ, FND, OE	AR, BOM, INV, WIP
<input type="checkbox"/>	<input checked="" type="checkbox"/>	CZ	Oracle Product Configurator (installed with BOM)	AK, AR, AX, AZ, BOM, FND, INV, OE	AP, AR, AS, CRP, ENG, FA, GL, MRP, PA, PAY, PER, PO, WIP
<input checked="" type="checkbox"/>	<input type="checkbox"/>	DT	Oracle DateTrack	FND	
<input type="checkbox"/>	<input type="checkbox"/>	EC	Oracle EDI Gateway	AK, AX, AZ, FND	
<input type="checkbox"/>	<input type="checkbox"/>	ENG	Oracle Engineering	AK, AX, AZ, BOM, FND, INV	AP, AR, AS, CRP, FA, GL, MRP, OE, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	<input type="checkbox"/>	FA	Oracle Assets	AK, AX, AZ, FND	AP, AS, BOM, ENG, GL, INV, PA, PAY, PER, PO, WIP
<input checked="" type="checkbox"/>	<input type="checkbox"/>	FF	Oracle FastFormula	FND	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	FLM	Oracle Flow Manufacturing	AK, AX, AZ, BOM, FND, INV	AP, AR, AS, CRP, ENG, FA, GL, OE, MRP, PA, PAY, PER, PO, WIP
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FND	Oracle Application Object Library		AD, ALR, AU
<input type="checkbox"/>	<input checked="" type="checkbox"/>	GHR	Oracle U.S. Federal Human Resources	AK, AX, AZ, FND, PER	
<input type="checkbox"/>	<input type="checkbox"/>	GL	Oracle General Ledger or Oracle Public Sector General Ledger	AK, AX, AZ, FND, RG	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	HXT	Oracle Time Management	AK, AX, AZ, FND, PAY, PER	

Table 1–1 Dependent Products

Full	Dep	Licensed Product		Additional Fully Installed Products	Additional Dependent Products
<input type="checkbox"/>	X	ICX	Oracle Self-Service Web Applications	AK, AX, AZ, FND	EC, INV, OE
<input type="checkbox"/>	<input type="checkbox"/>	INV	Oracle Inventory	AK, AX, AZ, FND	AP, AR, AS, BOM, CRP, ENG, FA, GL, MRP, OE, PA, PAY, PER, PO, QA, WIP
<input type="checkbox"/>	X	JA	Asia/Pacific Localizations	AK, AX, AZ, FND	AP, AR, FA, GL, INV, JG, OE, PER, PO
<input type="checkbox"/>	X	JE	European Localizations	AK, AX, AZ, FND	AP, AR, FA, GL, INV, JG, OE, PO, PER
<input type="checkbox"/>	<input type="checkbox"/>	JG	Regional Localizations	AK, AX, AZ, FND	AP, AR, BOM, FA, GL, INV, OE, PAY, PER, PO, WIP
<input type="checkbox"/>	X	JL	Latin America Localizations	AK, AX, AZ, FND	AP, AR, BOM, FA, GL, INV, JG, OE, PO, PAY, PER
<input type="checkbox"/>	<input type="checkbox"/>	MRP	Oracle Material Planning/ Master Scheduling	AK, AX, AZ, BOM, FND, INV	AP, AR, CRP, ENG, FA, GL, OE, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	X	MSC	Oracle Supply Chain Planning	AK, AX, AZ, BOM, FND, INV, MRP	AP, AR, CRP, ENG, FA, GL, MRP, OE, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	<input type="checkbox"/>	OE	Oracle Order Entry/Shipping	AK, AR, AZ, AX, FND, INV	AP, AS, BOM, CRP, ENG, FA, GL, MRP, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	X	OTA	Oracle Training Administration	AK, AX, AZ, FND, PER	AR, PER, PO
<input type="checkbox"/>	<input type="checkbox"/>	PA	Oracle Projects	AK, AX, AZ, FND	AP, AR, AS, BOM, CRP, ENG, FA, GL, INV, MRP, OE, PAY, PER, PO, WIP
<input type="checkbox"/>	<input type="checkbox"/>	PAY	Oracle Payroll	AK, AX, AZ, FND, PER	DT, FF

Table 1–1 Dependent Products

Full	Dep		Licensed Product	Additional Fully Installed Products	Additional Dependent Products
<input type="checkbox"/>	<input type="checkbox"/>	PER	Oracle Human Resources	AK, AX, AZ, FND	AP, AS, DT, ENG, FA, FF, GL, INV, PA, PAY, PO, WIP
<input type="checkbox"/>	<input checked="" type="checkbox"/>	PJM	Oracle Project Manufacturing	AK, AX, AZ, FND	BOM, CRP, DT, ENG, GL, INV, OE, PA, PER, PO, WIP
<input type="checkbox"/>	<input type="checkbox"/>	PO	Oracle Purchasing or Oracle Public Sector Purchasing	AK, AX, AZ, FND	AP, AR, AS, BOM, CRP, ENG, FA, GL, INV, MRP, OE, PA, PAY, PER, WIP
<input type="checkbox"/>	<input type="checkbox"/>	QA	Oracle Quality	AK, AX, AZ, FND, INV	AP, AR, AS, BOM, CRP, ENG, FA, GL, MRP, OE, PA, PAY, PER, PO, WIP
<input type="checkbox"/>	<input checked="" type="checkbox"/>	RG	Oracle Report Generator	AK, AX, AZ, FND, GL	
<input type="checkbox"/>	<input type="checkbox"/>	RLA	Oracle Release Management	AK, AR, AX, AZ, EC, FNC, INV, MRP, OE	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	SSP	Oracle SSP/SMP	AK, AX, AZ, FND, PER	PAY
<input type="checkbox"/>	<input checked="" type="checkbox"/>	VEH	Oracle Automotive	AK, AX, AZ, FND	AR, EC, INV, MRP, OE, RLA
<input type="checkbox"/>	<input type="checkbox"/>	WIP	Oracle Work in Process	AK, AX, AZ, BOM, FND, INV	AP, AR, AS, CRP, ENG, FA, GL, MRP, OE, PA, PAY, PER, PO

Product File Sizes

[Table 1–2](#) lists the product file sizes in Megabytes (MB) on the Oracle Applications base platform. Because file sizes vary across platforms, you should check your installation update for any updates to these figures.

Additional Information: Disk Space Requirements, *Oracle Applications Installation Update* for your platform

Note: Installation directories for ALL products are unloaded, including those that are not licensed or dependent. Allow 5 MB for each other unlicensed product.

Table 1–2 Product File Sizes

Product		Concurrent Processing Server	Forms Server	Admin Server	Web Server
	SAPPL_TOP/admin directory	✓ 4.1	✓ 4.1	✓ 4.1	✓ 4.1
	HTML directory tree	✓ 0	✓ 0	✓ 0	✓ 1.7
	Java directory tree (Applications files only)	✓ 0	✓ 0	✓ 7.2	✓ 7.0
AD	Oracle Applications DBA	✓ 19.1	✓ 19.1	✓ 19.1	✓ 19.1
AK	Oracle Common Modules	✓ 0.6	✓ 1.7	✓ 11.2	✓ 0.3
AU	Oracle Applications Utilities	✓ 0.4	✓ 406.2	✓ 0.2	✓ 0.2
AX	Global Accounting Engine	✓ 9.3	✓ 2.1	✓ 19.1	✓ 0.3
AZ	Oracle Applications Implementation	✓ 0.2	✓ 0.4	✓ 6.6	✓ 0.2
FND	Oracle Application Object Library	✓ 49.6	✓ 36.1	✓ 93.8	✓ 14.3
ALR	Oracle Alert	☐ 9.6	☐ 2.3	✓ 0.9	☐ 0.2
AP	Oracle Payables (commercial and public sector)	☐ 40.0	☐ 12.1	✓ 8.5	☐ 0.8
AR	Oracle Receivables (commercial and public sector)	☐ 77.5	☐ 19.4	✓ 24.0	☐ 1.2
AS	Oracle Sales and Marketing	☐ 10.3	☐ 13.0	✓ 6.7	☐ 0.4
BOM	Oracle Bills of Material (including Oracle Cost Management)	☐ 36.4	☐ 10.2	✓ 7.9	☐ 0.6
CE	Oracle Cash Management	☐ 3.6	☐ 2.9	✓ 2.0	☐ 0.3
CHV	Oracle Supplier Scheduling	☐ 0.2	☐ 1.0	✓ 0.2	☐ 0.2
CN	Oracle Sales Compensation	☐ 1.5	☐ 3.2	✓ 4.7	☐ 0.4

Table 1–2 Product File Sizes

Product		Concurrent Processing Server	Forms Server	Admin Server	Web Server
CRP	Oracle Capacity	☐ 7.4	☐ 1.2	✓ 0.4	☐ 0.2
CS	Oracle Service	☐ 5.4	☐ 7.3	✓ 4.1	☐ 0.3
CZ	Oracle Product Configurator	☐ 0.2	☐ 0.2	✓ 0.2	☐ 0.2
DT	DateTrack (dependent only)	☐ 0.3	☐ 0.7	✓ 0.9	☐ 0.2
EC	Oracle EDI Gateway	☐ 0.4	☐ 0.8	✓ 5.9	☐ 0.2
ENG	Oracle Engineering	☐ 3.3	☐ 2.0	✓ 2.6	☐ 0.3
FA	Oracle Assets	☐ 46.2	☐ 9.9	✓ 37.8	☐ 0.6
FF	FastFormula (dependent only)	☐ 2.9	☐ 1.8	✓ 3.1	☐ 0.3
FLM	Oracle Flow Manufacturing	☐ 0.2	☐ 0.2	✓ 0.2	☐ 0.2
GHR	Oracle U.S. Federal Human Resources	☐ 0.3	☐ 0.3	✓ 0.6	☐ 0.3
GL	Oracle General Ledger (commercial and public sector)	☐ 71.0	☐ 17.8	✓ 6.7	☐ 1.3
HXT	Oracle Time Management	☐ 1.7	☐ 2.5	✓ 0.9	☐ 0.2
ICX	Oracle Self-Service Web Applications	☐ 0.2	☐ 0.2	✓ 5.0	☐ 0.3
INV	Oracle Inventory	☐ 58.6	☐ 20.4	✓ 6.8	☐ 0.7
JA	Asia/Pacific Localizations	☐ 10.8	☐ 2.2	✓ 0.9	☐ 0.2
JE	European Localizations	☐ 52.4	☐ 1.7	✓ 1.9	☐ 0.5
JG	Regional Localizations	☐ 5.9	☐ 1.3	✓ 1.0	☐ 0.4
JL	Latin America Localizations	☐ 7.9	☐ 6.7	✓ 3.6	☐ 0.3
MFG	Oracle Manufacturing	☐ 0.2	☐ 0.2	✓ 0.2	☐ 0.2
MRP	Oracle Master Scheduling/MRP	☐ 37.3	☐ 10.5	✓ 3.5	☐ 0.8
MSC	Oracle Supply Chain Planning	☐ 0.2	☐ 0.2	✓ 0.2	☐ 0.2
OE	Oracle Order Entry/Shipping	☐ 71.1	☐ 17.2	✓ 7.5	☐ 0.7
OTA	Oracle Training Administration	☐ 0.7	☐ 5.4	✓ 4.5	☐ 0.3

Table 1–2 Product File Sizes

Product		Concurrent Processing Server	Forms Server	Admin Server	Web Server
PA	Oracle Projects	<input type="checkbox"/> 27.7	<input type="checkbox"/> 15.6	<input checked="" type="checkbox"/> 27.8	<input type="checkbox"/> 0.5
PAY	Oracle Payroll	<input type="checkbox"/> 31.0	<input type="checkbox"/> 17.8	<input checked="" type="checkbox"/> 31.2	<input type="checkbox"/> 1.5
PER	Oracle Human Resources	<input type="checkbox"/> 13.8	<input type="checkbox"/> 23.2	<input checked="" type="checkbox"/> 62.0	<input type="checkbox"/> 1.5
PJM	Oracle Project Manufacturing	<input type="checkbox"/> 0.2	<input type="checkbox"/> 0.8	<input checked="" type="checkbox"/> 0.4	<input type="checkbox"/> 0.2
PO	Oracle Purchasing (commercial and public sector)	<input type="checkbox"/> 39.4	<input type="checkbox"/> 16.7	<input checked="" type="checkbox"/> 15.4	<input type="checkbox"/> 0.9
QA	Oracle Quality	<input type="checkbox"/> 2.7	<input type="checkbox"/> 3.9	<input checked="" type="checkbox"/> 1.3	<input type="checkbox"/> 0.3
RG	Oracle Report Generator	<input type="checkbox"/> 10.0	<input type="checkbox"/> 4.4	<input checked="" type="checkbox"/> 2.0	<input type="checkbox"/> 0.2
RLA	Oracle Release Management	<input type="checkbox"/> 4.2	<input type="checkbox"/> 0.2	<input checked="" type="checkbox"/> 1.2	<input type="checkbox"/> 0.2
SSP	Oracle SSP/SMP	<input type="checkbox"/> 1.6	<input type="checkbox"/> 1.3	<input checked="" type="checkbox"/> 1.1	<input type="checkbox"/> 0.2
VEH	Oracle Automotive	<input type="checkbox"/> 0.8	<input type="checkbox"/> 0.7	<input checked="" type="checkbox"/> 1.0	<input type="checkbox"/> 0.2
WIP	Oracle Work in Process	<input type="checkbox"/> 25.3	<input type="checkbox"/> 9.3	<input checked="" type="checkbox"/> 2.9	<input type="checkbox"/> 0.6
Subtotal for installed and dependent-only products		_____	_____	461.0	_____
Number of other unlicensed products		× 5.0	× 5.0	× 0.0	× 0.5
Total for all products (MB)		_____	_____	_____	_____

Variations in File Sizes for Fully Installed Products

The actual size of Oracle Applications product files on your system may differ according to the block size used on your file system. Typical block sizes range from 2 KB to 8 KB, and vary according to operating system and disk size. All measurements in the table above are made assuming a block size of 8 KB. If you are using a smaller block size, such as 4 KB, your requirements may be slightly lower.

Requirements may also vary in the sizes of executable programs linked with the Oracle8 Server database and tools. These figures vary across platforms and different versions of the database and tools. Check your installation update for disk space figures that include the size of linked executable programs on your platform.

Additional Information: Disk Space Requirements, *Oracle Applications Installation Update* for your platform

There will also be variations in space requirements for common areas based on the products you choose to install. For example, the amount of space required for AU on the forms server may be as low as 24.1 MB, if you are only installing AOL, or as high as 428 MB, if you are installing all products. Requirements for Java and HTML files will also vary accordingly.

Variations in File Sizes for Dependent Products

The dependent product size estimates given at the bottom of [Table 1–2](#) are averages based on installations performed during our testing. Dependent product file sizes vary with the combination of products you install, however. Your disk space calculation may therefore overestimate or underestimate the disk space required for dependent products.

Additional Disk Space Requirements

A total disk space estimate must take into account the space required for files and directories other than the basic Oracle Applications product files. There are several types of files that are needed to use the installed Oracle Applications:

Language Files If you are installing or upgrading Oracle Applications in a language other than American English, you will need additional space for the language files. The breakdown of this space varies among languages, but a general approximation can be made.

Unloading and uncompressing the patch requires about 600 MB per language. This space roughly breaks down as follows:

- 300 MB for forms source files
- 200 MB for reports files
- 100 MB for seed data and other translated files

When applying the patch, the amount of space needed in your APPL_TOP will depend on your configuration. For example, if you are installing on a concurrent processing server, you will need at least 200 MB of space in your APPL_TOP for the reports files. On a forms server, you will need 300 MB of space for the forms sources, *plus* an additional 200–250 MB of space for forms executables.

The actual disk space required per language also varies depending on the number of products translated for that language. The language files are distributed throughout the product directory trees. If you unload product files onto different disks, the disk space required by these files is spread across those disks.

Additional Information: Language Directory Structure, *Oracle Applications Concepts*

Log and Output Files On a concurrent processing server or forms server, the disk space you need for Oracle Applications log and output files varies with your number of users, the types of reports they run, and the amount of time you keep files on disk. Start with at least 50 MB for each fully-installed product in each product group. If you have one product group with five fully installed products, for example, have at least 250 MB of disk space available. If you use parallel concurrent processing, you can spread log and output files across a network file system.

Additional Information: Log and Output Files, *Oracle Applications Concepts*

Suggestion: Log and output files are not automatically purged. After the installation, determine a strategy for archiving and purging these files. Closely monitor the disk space consumed by log and output files to determine how much space you need for these files on an ongoing basis.

Additional Information: Purge Concurrent Request and/or Manager Data, *Oracle Applications System Administrator's Guide*

Temporary Directories and Files Directories for temporary files generated by the concurrent managers (\$APPLTMP and \$REPORTS25_TMP) should each have at least 50 MB of available space. Some large Oracle Applications reports may need up to 100 MB of temporary space, however. You should also have at least 50 MB of space available for the default temporary directory (/tmp or /usr/tmp).

Additional Information: Temporary Files, *Oracle Applications Concepts*

Oracle Forms buffers records to disk and requires sufficient disk space before committing records. "Out of Record Buffer" messages indicate that the amount of disk space in the /tmp filesystem is insufficient, or the file limit is too small. File limits are set at the operating system level.

Additional Information: *Oracle Installation and Configuration Guide* for your platform

Suggestion: Temporary files are not automatically purged. After the installation, determine a strategy for purging these files. Closely monitor the disk space consumed by these files to determine how much space you need on an ongoing basis.

Files Generated by AutoInstall During an installation or upgrade, AutoInstall generates log, output, and restart files in subdirectories under the admin directory (\$APPL_TOP/admin). You can archive and delete these files after the installation or upgrade.

- On an administration server you may need 50–100 MB of disk space for these files.
- On any other type of server, you may need 10–20 MB of additional space.

Tablespace Requirements The Oracle Applications tables and indexes consume disk space in the form of database space.

Additional Information: [Disk Space Requirements](#) and [Tablespace Requirements](#) in this chapter

Release 11 installs all database objects for all Oracle Applications, even for the Applications you did not license. Release 11 generally requires more resources, particularly in database space than releases prior to Release 10.6. Be sure to plan accordingly before running your upgrade or installation.

Virtual Memory Oracle Applications also uses disk space as virtual memory (swap space) for applications processes.

Additional Information: [Memory Requirements](#) in this chapter.

Patches You also need disk space for applying patches, including release updates and individual bug fixes. These patches may create backup copies of files, which also require disk space.

Other Files Your total disk space estimate must also account for the requirements of files other than those directly related to Oracle Applications. For example:

- operating system software
- Oracle8 Server and Oracle tools files
- online backups
- applications development files
- files for other applications that you use
- users' personal files

Additional Information: *Oracle8 Installation Guide* for your platform

Verifying Disk Space

Once you have estimated the total disk space requirement, verify that you have sufficient disk space for Oracle Applications.

You can do the following if you do not have enough space available under the \$APPL_TOP directory:

- Install product files on separate disks.

Additional Information: *Distributed Directory Structure, Oracle Applications Concepts*

- Delete old product files before upgrading.

You can delete old product files before you run AutoInstall. Do not do this until you have prepared your existing products for upgrade, backed up your existing files, and saved any custom files.

Additional Information: [Delete old Oracle Applications files \(optional\)](#) in [Chapter 2](#)

- Delete files used only during an installation or upgrade.

You can back up and remove some files after the installation or upgrade. This frees space for files, such as log and output files, that users generate through Oracle Applications.

Additional Information: [Recovering Disk Space](#) in [Appendix A](#)

- Use another disk for log, output, or temporary files.

Oracle8 Server and Oracle Tools Requirements

This section tells how to prepare the Oracle Server and Oracle tools for an installation or upgrade of Oracle Applications. The database must be running for you to complete the instructions in this section. You will need access to the SYS and SYSTEM ORACLE schemas, which have ORACLE DBA privileges.

Additional Information: Oracle Server Databases and Tablespaces, *Oracle Applications Concepts*

Oracle Product Versions

During the installation process, Oracle Applications relinks executables with the Oracle8 Server and Oracle tools. You must therefore install all necessary Oracle products before you install Oracle Applications. Otherwise the Oracle Applications installation may fail.

Attention: These version numbers are for the base release of Oracle Applications. Your UNIX platform may require different versions for some or all of these products. Check your installation update for any changes to the requirements listed in this section.

Additional Information: Oracle Server and Oracle Tools Requirements, *Oracle Applications Installation Update* for your platform

In addition to the version numbers, the following table contains these three columns:

C	Concurrent processing server
F	Forms server
A	Administration server
W	Web server
D	Release 11 database server

Table 1–3 Certified Versions for Oracle Products

Oracle Product	Certified Version	C	F	A	W	D
✓—required X—not required □—optional						
The following tools are installed with Oracle8 Server Enterprise Edition:						
Oracle8 Server Enterprise Edition	8.0.4	✓	✓	✓	✓	✓
SQL*Plus	8.0.4	✓	✓	✓	✓	✓
Net8	8.0.4	✓	✓	✓	✓	✓
Oracle8 JDBC Drivers	8.0.4	X	X	✓	✓	X
PL/SQL	8.0.4	✓	✓	✓	✓	✓
SQL*Loader	8.0.4	✓	X	✓	X	X
Oracle8 ConText Cartridge (Required for CS and ICX products)	2.3.6	X	X	X	X	✓
The following tools are installed with Developer/2000:						
Developer/2000	1.6.1	✓	✓	X	X	X
Oracle Reports	2.5.7.5	✓	X	X	X	X
Oracle Forms	4.5.10.6	X	✓	X	X	X

Table 1–3 Certified Versions for Oracle Products

Oracle Product	Certified Version	C	F	A	W	D
✓—required ✕—not required □—optional						
Oracle Web Application Server is provided on separate media:						
Oracle Web Application Server	3.0.1	✕	✓	✕	✓	✕
Pro*C is installed with Oracle8, and is used to write C language extensions to Oracle Applications:						
Pro*C	8.0.4	□	□	✕	✕	✕

Note: Required versions of Oracle products may differ for your platform. See your platform-specific installation notes for complete information on version numbers for your platform.

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

Oracle Applications may not support products with version numbers higher than these certified versions. Contact Oracle Support Services to see which versions are supported.

If you have custom files, you must ensure they work correctly with the new releases of the Oracle8 Server and Oracle tools. Your custom files may need to be updated, and you should test them for compatibility with the new releases.

WARNING: Do not attempt to install or upgrade Oracle Applications if the version of a required tool on your system is older than the version certified for your platform.

Installing the Oracle8 Server and Oracle Tools

The Oracle8 Server and the Oracle tools include their own installation and reference manuals. When you install the Oracle8 Server, ensure that it is linked with PL/SQL. Your platform-specific Installation Update notes may list patches you should apply against the Oracle8 server.

Initialization Parameters

If you created a database using the "LARGE" settings in the default initialization parameter file, most of the parameters necessary for running Oracle Applications are already set properly. Table 1-4 lists additional initialization parameter values that your database administrator should set before you install or upgrade Oracle Applications. Restart the database if you change any parameters.

Additional Information: Initialization Parameters, *Oracle8 Reference*

Table 1-4 Initialization Parameter Values

Parameter	Value
DB_BLOCK_SIZE*	8192
DML_LOCKS	500
ENQUEUE_RESOURCES	5000
NLS_DATE_FORMAT	DD-MON-RR
NLS_LANGUAGE	AMERICAN
NLS_NUMERIC_CHARACTERS	“,.”
NLS_SORT	BINARY
NLS_TERRITORY	AMERICA
OPEN_CURSORS	255
OPTIMIZER_MODE	RULE
_OPTIMIZER_UNDO_CHANGES	TRUE
PROCESSES	75
ROW_LOCKING	ALWAYS
SHARED_POOL_SIZE	31457280
SORT_AREA_SIZE	256000
UTL_FILE_DIR	<location of temporary directory>

* DB_BLOCK_SIZE should only be set for database creation before a fresh install, not when upgrading from an earlier release.

UTL_FILE_DIR Parameter

You must identify the directory on the Release 11 database server that you plan to use for temporary PL/SQL files. The UTL_FILE_DIR parameter in the init.ora file points to this temporary directory. For example, if your temporary directory is /d01/applmgr/110/plsqllog, the init.ora file might contain the following line:

```
utl_file_dir = /d01/applmgr/110/plsqllog, /d02/oracle/data/inbound
```

Additional Information: Temporary Files, *Oracle Applications Concepts*

Numeric Parameters

The numeric values listed in [Table 1-4](#) are minimum parameter values for installing or upgrading Oracle Applications. When upgrading, you can set these values higher if necessary.

Language Parameters

The database must run in American English during an installation or upgrade of Oracle Applications.

Additional Information: [Set Oracle8 Server Parameters in Chapter 4](#)

Attention:

- NLS_DATE_FORMAT and NLS_NUMERIC_CHARACTERS must be set to the values given in [Table 1-4](#). Oracle Applications cannot function with any other values for these parameters.
 - If you are using the NLS_DATE_LANGUAGE parameter, be sure it is set to the value AMERICAN.
 - To avoid the potential performance degradation associated with using some of the linguistic sort sequences, we recommend using NLS_SORT=BINARY regardless of the installed language.
-
-

Optimizer Mode

Use rule-based optimization as the default when you install, upgrade, or use Oracle Applications. Oracle Applications SQL statements are tuned for rule-based optimization. Only selected statements are tuned for cost-based optimization.

Additional Information: [Analyzing the Database](#) in [Appendix A](#)

Setting Parameters for Normal Use

After you install or upgrade Oracle Applications, your database administrator may need to modify the initialization parameters to tune the database for optimal performance. Tuning the database requires a thorough understanding of the Oracle8 Server. The value you should set for a certain parameter depends on many factors, including the number of users, the number of applications, and the settings of other parameters.

Suggestion: If you do not have an experienced database administrator at your site, Oracle Consulting Services can help you tune the database.

Additional Information: Part III: Optimizing Database Operations, and Part IV: Optimizing Oracle Instance Performances, *Oracle8 Tuning*

SQL*Plus Options

Verify that SQL*Plus options are set correctly in your SQL*Plus login files. Check the login.sql files for all Oracle Applications users and the glogin.sql file in \$ORACLE_HOME/sqlplus/admin, and ensure that the following commands are set correctly.

The following options and values are required:

```
SET PAUSE OFF
SET NEWPAGE 1
```

The following option and value is recommended:

```
SET PAGESIZE 60
```

Log in as applmgr, connect to SQL*Plus with any ORACLE user ID, and check these options. For example:

```
$ sqlplus <SYSTEM username>/<SYSTEM password>
SQL> show newpage
newpage 1
SQL> show pause
pause off
SQL> show pagesize
pagesize 60
SQL> exit
```

Additional Information: SQL*Plus User’s Guide and Reference

If the SQL*Plus pause formatting option is turned on, SQL*Plus waits for a manual return after each page of output. This causes installation and upgrade tasks, as well as Oracle Applications reports, to fail.

If an option’s value is incorrect, change it in the applmgr login.sql file or the glogin.sql file.

Attention: In general, you should not change the default SQL*Plus settings. Changes to the default NUMFORMAT, CONCAT, ESCAPE, and SCAN settings may also cause SQL*Plus scripts to fail.

Tablespace Requirements

This section summarizes the tablespace requirements for an Oracle Applications database. You must set up your tablespaces before you install or upgrade Oracle Applications. Your ORACLE DBA can determine database space availability.

Additional Information: Oracle8 Server Organization and Requirements, *Oracle Applications Concepts*

SYSTEM Tablespace

The following table lists sizing guidelines for the Oracle Server SYSTEM tablespace.

Product Group	SYSTEM Size
One product installation group	700 MB
Each additional product installation group	+ 450 MB*

* Note that ORACLE data dictionary growth is not always linearly cumulative, so the SYSTEM tablespace used for each additional product installation group is variable.

You may need less SYSTEM space than indicated in the previous table if you install tools and other tables in tablespaces other than SYSTEM.

Rollback Segments

[Table 1–5](#) lists space recommendations for rollback segments. You can create segments with two initial extents. Set the maximum number of extents to allow the segment to grow to at least the minimum size given in the following table.

Table 1–5 Rollback segment requirements

Action	Number of Segments	Extent Size	Minimum Size
Installing	1 segment for each worker	1 MB	40 MB
Upgrading	1 segment for each worker	1 MB	60 MB
Normal Use	1 segment for every 25 users + 1 for each concurrent manager	1 MB	20 MB

To determine the number of segments for normal use, define the number of users as the maximum number of people who will use Oracle Applications at the same time. You may want an additional rollback segment for each concurrent manager. You may need more rollback space depending on how you use Oracle Applications.

If your database is larger than 2 GB, consider at least one segment for each AutoInstall parallel worker, with each segment having an initial extent between 1 MB and 3 MB, next extents between 1 MB and 3 MB, and 30 extents minimum.

Temporary Tablespace

[Table 1–6](#) gives space guidelines for a temporary tablespace:

Table 1–6 Temporary tablespace requirements

Action	Recommended Extent Size	Minimum Tablespace Size
Installing	4 MB	25 MB
Upgrading	4 MB	100 MB
Normal use	4 MB	50 MB

We recommend setting PCTINCREASE to 0 for temporary tablespaces. When creating a temporary tablespace, identify the tablespace as temporary with the following command:

```
CREATE TABLESPACE <tablespace> TEMPORARY;
```

Additional Information: Guidelines for Managing Schema Objects
Oracle8 Administrator's Guide

Product Table and Index Tablespaces

Table 1–7 shows the incremental database space used in Megabytes (MB) when a product is installed with 100% sizing factor and allocates one more extent for all its objects.

Suggestion: Unless your installation requires a different sizing factor, we recommend you use a sizing factor of 100% for all products.

AutoInstall installs all objects and seed data for all Oracle Applications products in your database, regardless of whether or not you licensed the products. All products are installed at the sizing factor shown on the AutoInstall Database Parameters screen. By default, all non-licensed products install at 100% sizing factor, although you can change the sizing factor on the Database Parameters screen before installation.

Additional Information: Sizing Factor, *Oracle Applications Concepts*

Table 1–7 Product Table and Index Sizes

Product		Incremental Growth at 100% Sizing (in MB)	
		Table	Index
AK	Oracle Common Modules	18.6	7.8
ALR	Oracle Alert	8.1	4.4
AP	Oracle Payables or Oracle Public Sector Payables	73.6	90.8
AR	Oracle Receivables or Oracle Public Sector Receivables	368.8	531.9
AS	Oracle Sales and Marketing	23.5	63.3
AX	Global Accounting Engine	43.6	18.2
AZ	Oracle Applications Implementation	0	0

Table 1–7 Product Table and Index Sizes

Product		Incremental Growth at 100% Sizing (in MB)	
		Table	Index
BOM	Oracle Bills of Material (including Oracle Cost Management and Oracle Product Configurator)	61.8	33.0
CE	Oracle Cash Management	54.4	14.2
CHV	Oracle Supplier Scheduling	0	0
CN	Oracle Sales Compensation	130.1	108.5
CRP	Oracle Capacity	5.2	7.1
CS	Oracle Service	6.6	22.0
EC	Oracle EDI Gateway	19.7	10.8
ENG	Oracle Engineering	15.5	5.3
FA	Oracle Assets	37.9	51.2
FLM	Oracle Flow Manufacturing	0	0
FND	Oracle Application Object Library	182.7	115.5
GL	Oracle General Ledger or Public Sector General Ledger	62.5	33.1
HXT	Oracle Time Management	4.2	10.9
ICX	Oracle Applications for the Web	20.0	18.0
INV	Oracle Inventory	118.1	156.2
JA	Asia/Pacific Localizations	81.6	74.3
JE	European Localizations	6.1	8.1
JG	Regional Localizations	0	0
JL	Latin America Localizations	6.2	1.8
MRP	Oracle Material Planning/Master Scheduling (including Oracle Supply Chain Planning)	33.5	49.0
OE	Oracle Order Entry/Shipping	71.1	44.8
OTA	Oracle Training Administration	0.6	1.7
PA	Oracle Projects	461.4	532.9

Table 1–7 Product Table and Index Sizes

		Incremental Growth at 100% Sizing (in MB)	
Product		Table	Index
PER	Oracle Human Resource Management Systems (including DateTrack, FastFormula, Oracle Payroll, and Oracle U.S. Federal Human Resources)	255.3	537.9
PJM	Oracle Project Manufacturing	0.4	0.8
PO	Oracle Purchasing or Oracle Public Sector Purchasing (including Oracle Supplier Scheduling)	78.3	81.8
QA	Oracle Quality	11.8	3.4
RG	Oracle Report Generator	26.7	20.1
RLA	Oracle Release Management	6.4	2.0
SSP	Oracle SSP/SMP	2.4	3.9
VEH	Oracle Automotive	12.0	4.0
WIP	Oracle Work in Process	94.6	75.0

The total minimum tablespace for all product tables at 100% sizing is 2405 MB and for indexes is 2745 MB. For example, if you installed Oracle General Ledger and Oracle Report Generator at 200% sizing, and all other products at 100% sizing, the minimum tablespace you need for tables is 527.5 MB and for indexes is 327.5 MB:

Product	Table	Index
Total of all products at 100% sizing	2405.0	2745.0
Oracle General Ledger, additional 100%	62.5	33.1
Oracle Report Generator, additional 100%	26.7	20.1
Totals	2494.2	2798.2

Multiple Product Installation Groups

The sizing figures in [Table 1–7](#) assume that you are installing one product installation group. Each additional product installation group you set up requires additional table and index space.

Additional Information: [Upgrading Multiple Sets of Books in Chapter 3](#)

Attention: Your 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 architecture implementation to Release 11 without using Multiple Organization architecture.

Interim Tablespace

If you have set up an interim tablespace for use with Oracle General Ledger or Oracle Public Sector General Ledger, have at least 30 MB of space free in this tablespace. You may need more space depending on how you use your Oracle General Ledger application. The interim tablespace is not used during an installation or upgrade.

Languages

Additional space is required for the translated seed data for each installed language. Most extra seed data is stored in the Application Object Library tablespaces, but additional tablespace should be allocated for AK, AS, AX and AZ . Approximately 15 MB of table and 10 MB of index database space per installed language is required for each of these products.

Localizations

Most localizations, such as European Localizations, Asia/Pacific Localizations, and Latin America Localizations, install in a separate schema for the region. Some localization modules require additional database space in the product tablespaces. A localization module for Oracle Payables, for example, may require database space in the Oracle Payables table and index tablespaces for its localized tables and indexes.

The actual database space required varies by localization module. You should have at least 5 MB of space available in each Oracle Financial product's table and index tablespaces for each localization module that you install. Also have at least 5 MB of space per module in both the table and index tablespaces used by the Regional Localization schema.

Tablespace Requirements

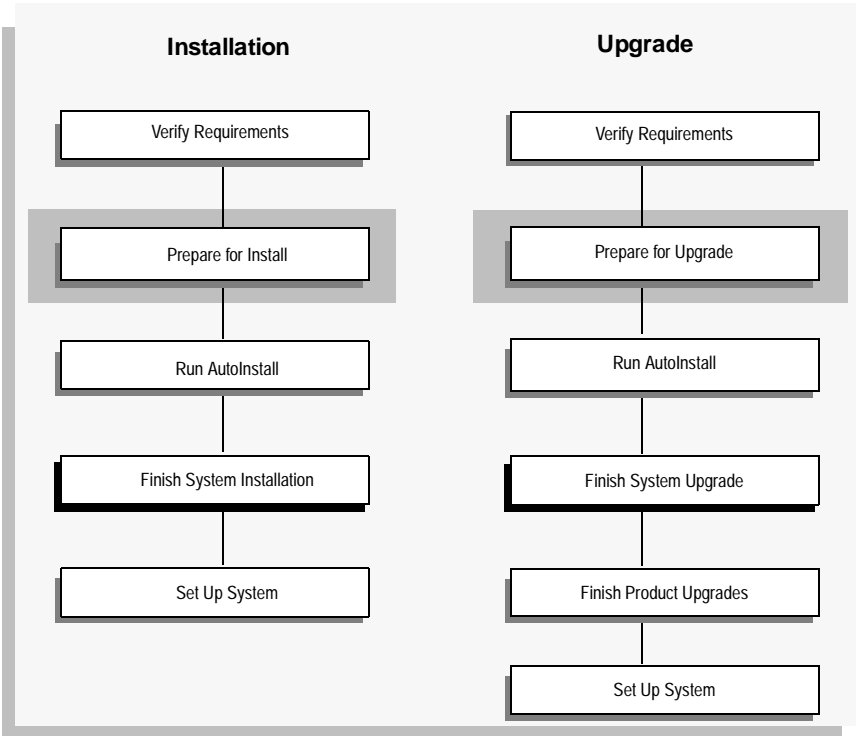
Preparing to Install or Upgrade

This chapter explains how to prepare your system before running AutoInstall. This chapter contains these sections:

- [Your Place in the Installation or Upgrade Process](#)
- [Checklist for This Chapter](#)
- [Set Up the Technology Stack](#)
- [Create the Applications Login Account](#)
- [Create Directories](#)
- [Prepare Your Database](#)
- [Set Environment Variables](#)
- [Unload AutoInstall Files](#)
- [Prepare for Upgrade](#)
- [Set Up AutoInstall Environment](#)

Your Place in the Installation or Upgrade Process

The following figure shows your current place in the Oracle Applications installation or upgrade process. This chapter details the steps highlighted in this figure.



Checklist for This Chapter

Each of the following steps is explained in this chapter.

Set Up the Technology Stack

- ☐ 1: Plan architecture
- ☐ 2: Install Oracle8 Enterprise Edition
- ☐ 3: Install Oracle8 JDBC drivers
- ☐ 4: Install Java Runtime Environment
- ☐ 5: Install Oracle ConText Cartridge
- ☐ 6: Install Oracle Web Application Server
- ☐ 7: Install Oracle Forms
- ☐ 8: Install Oracle Reports
- ☐ 9: Set up tablespaces
- ☐ 10: Configure Net8

Create the Applications Login Account

- ☐ 11: Create applmgr login
- ☐ 12: Verify PATH environment variable
- ☐ 13: Verify \$ORACLE_HOME, \$ORACLE_SID, and \$TWO_TASK
- ☐ 14: Set umask value (recommended)

Create Directories

- ☐ 15: Create top Applications directory
- ☐ 16: Create product directories (optional)
- ☐ 17: Create Java, HTML, and documentation directories
- ☐ 18: Create common directory for log and output files (optional)
- ☐ 19: Create temporary directories (optional)

Prepare Your Database

- ☐ 20: Separate product groups
- ☐ 21: Drop custom schemas that match APPS%
- ☐ 22: Verify SYS.DUAL
- ☐ 23: Verify custom index privileges
- ☐ 24: Drop conflicting custom public synonyms
- ☐ 25: Rename custom database objects with Applications prefixes
- ☐ 26: Perform category 1 upgrade steps

Set Environment Variables

- ☐ 27: Verify \$APPL_TOP
- ☐ 28: Set NLS environment variables

Unload AutoInstall Files

- ☐ 29: Unload installation directory from main CD
- ☐ 30: Edit adovars.env file
- ☐ 31: Run the AutoInstall setup script
- ☐ 32: Unload localization installation directory (optional)
- ☐ 33: Apply pre-installation patches
- ☐ 34: Run preparatory scripts
- ☐ 35: Run adsystx.sql
- ☐ 36: Apply ORACLE patches

Prepare for Upgrade

- ☐ 37: Back up Oracle Applications and customizations
- ☐ 38: Perform category 2 upgrade steps
- ☐ 39: Set up AutoInstall installation groups
- ☐ 40: Verify ORACLE IDs
- ☐ 41: Disable AOL Audit Trail feature

- ☐ 42: Disable custom triggers, constraints, and indexes
- ☐ 43: Drop event alert triggers in custom schemas
- ☐ 44: Perform category 3 upgrade steps
- ☐ 45: Shut down the concurrent managers
- ☐ 46: Back up the Oracle Applications database (recommended)
- ☐ 47: Upgrade Oracle8 Server and Oracle Tools
- ☐ 48: Run preparatory scripts
- ☐ 49: Turn off automatic archive logging
- ☐ 50: Verify rollback segment sizing for upgrades
- ☐ 51: Delete old Oracle Applications files (optional)

Set Up AutoInstall Environment

- ☐ 52: Restrict access to log and restart files
- ☐ 53: Turn off Oracle8 Server Parallel Query Option
- ☐ 54: Check SQL*Plus options
- ☐ 55: Run AutoInstall environment file

Set Up the Technology Stack

1. Plan architecture

Before installing or upgrading, you should plan your Release 11 architecture and distribute server functionality across the machines in your environment. Based on the requirements for each server, determine which machine or machines will be your Oracle8 database server, concurrent processing server, administration server, forms server, and web server. Note that one machine may function as more than one type of server. For example, your Oracle8 database server and concurrent processing server may be the same machine.

Additional Information: Network Computing Architecture, *Oracle Applications Concepts*

Contact Oracle Consulting Services if you need help planning your architecture.

2. Install Oracle8 Enterprise Edition

The Oracle8 Enterprise Edition 8.0.4 and its component tools must be installed on all servers. At minimum, you should install PL/SQL, SQL*Plus, Net8, and the TCP/IP Protocol Adapter in addition to the Oracle8 Enterprise Server. You may also install any other tools you require, such as Pro*C if you plan to develop customizations in C.

Additional Information: [Oracle8 Server and Oracle Tools Requirements](#) in Chapter 1; *Underlying System Components*, *Oracle Applications Concepts*; *Oracle8 Installation Guide*

If you are upgrading from an existing installation, install the new Oracle Home, but do not upgrade your database yet.

3. Install Oracle8 JDBC drivers

The Oracle8 JDBC drivers are included in the Oracle8 Enterprise Edition, and should be installed on your administration server and web server.

Additional Information: *Oracle8 JDBC Drivers*

When the installation asks which JDBC drivers to install, choose the JDK 1.1 JDBC OCI Driver. You may install other drivers if you plan to use them with other software.

4. Install Java Runtime Environment

The Java Runtime Environment should be installed on your administration server. This is provided on the *Oracle Applications Release 11 Interoperability CD*.

Additional Information: [Oracle8 Server and Oracle Tools Requirements](#) in Chapter 1; *Underlying System Components*, *Oracle Applications Concepts*; *Oracle Applications Release 11 Installation Update* for your platform

5. Install Oracle ConText Cartridge

If you plan to install Oracle Service or Oracle Self-Service Applications for the Web, you must install the Oracle ConText Cartridge on the Release 11 database server. The Oracle Context Cartridge is included with the Oracle8 Enterprise Edition.

6. Install Oracle Web Application Server

Install Oracle Web Application Server 3.0.1 only on your forms servers and web servers. The Oracle Web Applications Server is on a separate CD, and includes the *Oracle Web Application Server Installation Guide*.

This installation must be done in two steps. First, run Oracle Installer, and when you reach the Software Asset Manager screen, select Oracle Web Application Server (the first entry). Second, when the installation completes and returns you to the Software Asset Manager screen, expand the options for Oracle Web Application Server, then for Web Application Server Cartridges, and select Oracle7 JDBC Drivers. This installation will ask you which JDBC drivers to install: choose *only* the JDK 1.0 JDBC OCI Driver option.

WARNING: Choosing any other drivers will overwrite the drivers installed in Step 3 and will cause Oracle Applications HTML-based functionality to work improperly.

Release 11 supports other HTTP listeners, such as Netscape Enterprise Server and Microsoft Internet Information Server, but the installation steps in this manual assume you are using the default listener supplied with Oracle Web Application Server.

7. Install Oracle Forms

Install Oracle Forms Web Server on your forms server. This installs all the components needed for running and designing forms. Oracle Forms Web Server is on the *Oracle Developer/2000 Release 1.6.1* CD, and instructions are in the *Oracle Developer/2000 Installation Manual*.

Attention: The platform-specific installation notes will indicate if the exact title of the CD from which you should install Oracle Forms differs from the title given here.

When installing Oracle Forms Web Server, Oracle Installer asks you to select one or more user interfaces for Oracle Forms. At minimum, you must select "Webforms Interface" and "Designer and Generator Executables". You may install other interfaces, but Release 11 only requires these two interfaces.

Installing Developer/2000 tools creates a file named f45webm in the \$ORACLE_HOME/bin directory. The AD Administration utility will create another version of this file for Release 11. To avoid a filename conflict, rename the \$ORACLE_HOME/bin/f45webm file to a different filename. For example:

```
$ cd $ORACLE_HOME/bin
$ mv f45webm f45webm.notapps
```

8. Install Oracle Reports

Install Oracle Reports Web Server on your concurrent processing server. Oracle Reports Web Server is on the *Oracle Developer/2000 Release 1.6.1* CD, and instructions are in the *Oracle Developer/2000 Installation Manual*.

Attention: The platform-specific installation notes will indicate if the exact title of the CD from which you should install Oracle Reports differs from the title given here.

When installing Oracle Reports Web Server, Oracle Installer asks you to select one or more user interfaces for Oracle Reports. At minimum, you must select "Character Mode Interface" and "Reports Designer and Converter Executables". You may install other interfaces, but Release 11 only requires these two interfaces.

9. Set up tablespaces

If this is a fresh install, create a database for Oracle Applications and set up tablespaces for product tables, indexes, and temporary tables. If you are upgrading from an older release of Oracle Applications, verify that you have enough space in the tablespaces for all product objects. For an upgrade, you may also need to create additional tablespaces for new products in this release.

Additional Information: [Tablespace Requirements](#) in Chapter 1

10. Configure Net8

You must configure Net8 on every server on each tier. Create a database alias on each machine to connect to the Release 11 database server. Consult your database administrator to set up Net8.

Additional Information: *Oracle Net8 Administrator's Guide*

Attention: Before continuing, check the *Oracle Applications Release 11 Interoperability* CD for any patches that should be applied to the Oracle8 Server.

Create the Applications Login Account

You need one main applications login to run AutoInstall and to manage Oracle Applications product files. We refer to this login as applmgr.

Attention: Do not use the applmgr login to manage your Oracle8 Server files and database. We recommend you use a login such as oracle for this purpose.

11. Create applmgr login

If you are upgrading, your Applications login has been created already. If you are installing Oracle Applications, create an applmgr login and set it to run under a Bourne or Korn shell.

WARNING: AutoInstall may fail if you use a C shell as the default shell for the applmgr login.

In order to run Server Manager from the applmgr for UNIX account, the applmgr login should belong to the Oracle8 Server DBA group.

If you are setting up parallel concurrent processing, create identical applmgr logins on each server. The applmgr account on each server must be configured to allow remote logins from each of the other applmgr accounts. Typically, this is done by creating a files named .rhosts in the home directory.

Additional Information: The for UNIX man page for the rlogin or rsh commands contains more information on allowing remote logins on your platform.

12. Verify PATH environment variable

Log in as applmgr and check the PATH environment variable:

```
$ echo $PATH
```

The PATH variable should include the applmgr home directory and the following directories:

- . (a period that indicates the current directory)
- /bin

- /usr/sbin
- /usr/ucb

These directories may be different on your platform. The /usr/ucb path must follow /bin and /usr/bin in the PATH value. Exclude /usr/ucb if your system does not have this directory.

- directory containing Java executable (administration server and web server only)

The path on your administration server and web server must include the directory containing the Java executable. The executable is named `jre`, if you installed the Java Runtime Environment, or `java` if you installed the Java Developer Kit, and is located in the directory in which you installed either Java package. The following command is available on many for UNIX platforms and returns the name of the directory containing the Java executable:

```
$ which jre
```

- directory containing ping executable (concurrent processing server only)

If you are setting up parallel concurrent processing, \$PATH must include the directory that contains the for UNIX executable program `ping`. This program is typically in the `/etc` or the `/usr/bin` directory.

If necessary, modify the PATH variable by editing the `.profile` file in the `applmgr` home directory. If you change the PATH variable, run the `.profile` file and verify the changes:

```
$ . .profile
$ echo $PATH
```

Consult the system administrator if the PATH environment variable does not show the necessary directories in the correct order after you modify `.profile`.

13. Verify \$ORACLE_HOME, \$ORACLE_SID, and \$TWO_TASK

Ensure that you point to the database in which you want to install your Oracle Applications products by checking the value of \$ORACLE_HOME and \$ORACLE_SID, if your database is running on this machine.

```
$ echo $ORACLE_HOME
$ echo $ORACLE_SID
```


You can usually set these variables by running the "oraenv" shell script, typically located \$ORACLE_HOME/bin or in a local directory on the filesystem. For example:

```
$ . /usr/local/bin/oraenv
```

Running this script will prompt you for an ORACLE_SID. If the system administrator has set up an oratab file, it will automatically set your ORACLE_HOME for that database, and update your PATH to include \$ORACLE_HOME/bin. If the oratab file is not available, you will be prompted for an ORACLE_HOME also.

Contact your system administrator if you are unsure whether the oraenv script is available on your machine.

You can also set these variables manually if you do not have the oraenv script installed. Here is an example of changing ORACLE_SID:

```
$ ORACLE_SID=proddb; export ORACLE_SID
```

If you are connecting to a database on a remote machine using Net8, check the value of TWO_TASK:

```
$ echo $TWO_TASK
```

Remember that if TWO_TASK is set, it overrides ORACLE_SID.

Additional Information: *Oracle Installation and Configuration Guide* for your platform.

When setting ORACLE_HOME and ORACLE_SID or TWO_TASK manually, be sure to include the \$ORACLE_HOME/bin directory in your PATH setting as well.

If you are installing on a concurrent processing server using Oracle Parallel Server, ORACLE_SID should be set on each concurrent processing server to that server's local database instance.

14. Set umask value (recommended)

We recommend that you set the applmgr login's umask value to 022 by adding this line to the .profile file:

```
umask 022
```

This umask value removes write privilege on files and directories for all logins except applmgr. With this value, the default protection for files created through applmgr is 755 in most cases.

Create Directories

You create directories through the applmgr login, which automatically sets the correct file access privileges.

Additional Information: Directory Structure, *Oracle Applications Concepts*

15. Create top Applications directory

Create an empty directory owned by applmgr. This is the applications top directory. You later assign the environment variable APPL_TOP to this directory. Be sure you have sufficient disk space to unload the AOL and AD files to this directory.

Additional Information: [Product File Sizes](#) in Chapter 1

AutoInstall creates all other product file directories below this for you. This manual uses /d01/appl/110 as an example of the top applications directory.

16. Create product directories (optional)

If you plan to unload files to top directories other than \$APPL_TOP, AutoInstall will try to create the top directory for you. You must make sure the highest-level parent directory exists and is owned by applmgr.

For example, you may want to unload some manufacturing applications to /d02/mfg/110. You need to create the disk mount point /d02 if it does not already exist. You can also create the mfg/110 directory, although this is not strictly necessary if the applmgr user has write permissions to the /d02 disk. If not, have the system administrator modify the write permissions on that directory (/d02), or create the directories and change their ownership so they are owned by the applmgr user.

17. Create Java, HTML, and documentation directories

Create the directories to which AutoInstall will install Java class files, HTML, and context-sensitive help. The applmgr user must have write permissions to these directories. These directories correspond to the paths contained in the JAVA_TOP, OAH_TOP, and OAD_TOP environment variables. You will later modify the adovars.env file and map these environment variables to the directories you created for the Java, HTML and documentation files.

Additional Information: Java and HTML Files, *Oracle Applications Concepts*

18. Create common directory for log and output files (optional)

If you plan to use a common directory for a product group's log and output files, you should create the directory before you run AutoInstall.

If you are installing on a concurrent processing server and setting up parallel concurrent processing, we recommend that you create a common directory on each server that will run concurrent managers.

Additional Information: Log and Output Files, *Oracle Applications Concepts*

Create log and output subdirectories under each common directory. AutoInstall prompts you for the pathname of the common directory and the names of the log and output subdirectories when you install or upgrade each product group. The default names for the subdirectories are log and out, but you can choose any names.

If you plan to use a single common directory for multiple product groups, create separate log and output directories for each product group under the common directory:

```
$ cd <common directory>
$ mkdir <log1> <out1>... <logn> <outn>
```

Here is an example:

```
$ cd /d01/appl/110/common
$ mkdir logtst outtst logdev outdev
```

In this example, the logtst and outtst directories are for the first product group, the logdev and outdev directories are for the second product group, and so on.

19. Create temporary directories (optional)

If you are setting up directories for Oracle Applications temporary files, create those directories before you run AutoInstall. The applmgr user must have write access to these directories. If you are installing on concurrent processing servers and setting up parallel concurrent processing, you may want to create a temporary directory or directories on each parallel server. A concurrent processing server does not need read and write access to the temporary directories on every other server.

Additional Information: Temporary Files, *Oracle Applications Concepts*

Prepare Your Database

Follow Step 20 through Step 26 to prepare your database for upgrade from an earlier release of Oracle Applications. If you are installing Oracle Applications for the first time, skip these steps and proceed with Step 27.

Additional Information: Installation Procedure Updates, *Oracle Applications Installation Update* for your platform

20. Separate product groups

You must separate multiple product groups currently existing in one database to separate databases. The Release 11 architecture uses PUBLIC synonyms and objects in the SYSTEM schema, and requires that you no longer have more than one product group in a database.

A product group refers to a set of Oracle Application product schemas linked together by an Oracle Application Object Library schema, called APPLSYS by default.

Follow the instructions in Appendix A to migrate Oracle Applications by exporting the database in pieces. In other words, export all the schemas related to the product group you wish to separate from the existing database.

Additional Information: [Migrating the Oracle Applications Database](#) in Appendix A

If you have a demonstration product group installed in the same database, you can alternatively delete these tables to free that database space. There are two ways to do this:

- Your database administrator can remove all objects for the demonstration database ORACLE user ID using the DROP USER CASCADE command.
- If the demonstration database is in its own tablespace, your database administrator can remove the tablespace.

21. Drop custom schemas that match APPS%

AutoInstall automatically creates a schema named APPS, and may create additional schemas with names using the prefix APPS. You must export and optionally migrate any custom schemas you have that match APPS%, then drop these schemas so there are no conflicts.

Additional Information: How AutoInstall Creates APPS Schemas, *Oracle Applications Concepts*; Export and Import, *Oracle8 Utilities*

Use the following SQL statement to find any schemas whose name matches APPS%:

```
$ sqlplus <SYSTEM username>/<SYSTEM password>
SQL> select USERNAME from ALL_USERS
      2> where USERNAME like 'APPS%';
```

Attention: Do not drop any APPS schema installed by AutoInstall.

If the SELECT statement returns any custom schema names, export these custom schemas before dropping them. Then use the following command to drop a conflicting schema:

```
$ sqlplus <SYSTEM username>/<SYSTEM password>
SQL> drop user <conflicting schema name> cascade;
```

You must also disable the ORACLE ID for each conflicting schema. Using the System Administrator responsibility, navigate to the Register Oracle IDs window at Security:ORACLE:Register and disable the privileges for each of these schemas.

You must then ensure that no data groups use this ORACLE ID. Use the System Administrator responsibility and navigate to the Define Data Group form at Security:ORACLE:DataGroup.

Additional Information: Register ORACLE IDs, Define Data Group, *Oracle Applications System Administration Reference Manual, Release 10* or *Oracle Applications System Administrator's Guide, Release 11*

If you import your schemas under a new name, follow the instructions in this manual for integrating custom schemas.

Additional Information: Impact on Custom Schemas, *Oracle Applications Concepts*; [Reapply product customizations to files and databases](#) in Chapter 4

22. Verify SYS.DUAL

The SYS.DUAL table must be accessible to all Oracle Applications schemas and must contain exactly one row. For example, use the following command to verify this for APPLSYS:

```
$ sqlplus <AOL username>/<AOL password>
SQL> select count(*) from SYS.DUAL ;
```

If the SELECT does not return exactly one row or there is an ORACLE error, fix the problem or contact Oracle Support Services.

23. Verify custom index privileges

If a schema owns a custom index on an Oracle Applications table, ensure that the schema owning the table has privileges to drop these custom indexes. The ODF Comparison utility may try to drop custom indexes when maintaining Oracle Applications tables.

Use the following command to identify indexes that may cause a problem:

```
$ sqlplus <SYSTEM username>/<SYSTEM password>
SQL> select OWNER, INDEX_NAME, TABLE_OWNER, TABLE_NAME
      2> from DBA_INDEXES
      3> where OWNER != TABLE_OWNER;
```

Each listed TABLE_OWNER must have DROP ANY INDEX privileges. Use the following command to grant this privilege:

```
$ sqlplus <SYSTEM username>/<SYSTEM password>
SQL> grant DROP ANY INDEX to <TABLE_OWNER>;
```

24. Drop conflicting custom public synonyms

Ensure there are no public synonyms with the same name as an Oracle Applications schema. You can execute the following:

```
$ sqlplus <AOL username>/<AOL password>
SQL> select * from ALL_SYNONYMS
      2> where OWNER='PUBLIC'
      3> and SYNONYM_NAME in
      3> (select ORACLE_USERNAME from FND_ORACLE_USERID);
```

Also ensure you do not have any public synonyms that conflict with default usernames for new products. These include AK, AX, AZ, CE, CHV, CS, CZ, EC, FLM, GHR, HXT, ICX, MSC, OTA, PJM, QA, RLA, SSP, VEH, and WH. Drop any conflicting public synonyms. If necessary, you can recreate these as private synonyms in a custom schema after installing or upgrading.

25. Rename custom database objects with Applications prefixes

If you are upgrading from a previous release, AutoInstall automatically runs a script called `addrpbco.sql`, which drops database objects in Applications schemas that match standard Oracle prefixes. For example, if you have a stored procedure named `GL_CUSTOM` in the schema for Oracle General Ledger, it will be dropped during the upgrade. An object named `XXGL_CUSTOM` would not be affected.

If you wish to preserve these database objects, you should relocate them to another schema or rename them to a name that does not use the Oracle standard prefixes.

WARNING: When naming database objects, use at least four characters for your product short name, followed by an underscore. You must use only letters, digits, and underscores when naming database objects. For example, you might define your custom application to use the short name `CUST`, and database objects to begin with a `CUST_` prefix.

Additional Information: Naming Standards, *Oracle Applications Developer's Guide*

26. Perform category 1 upgrade steps

If you are upgrading existing products on a server or stand-alone machine, you should perform category 1 upgrade steps before you unload the installation directory from the CD. You must perform all upgrade steps before you run AutoInstall.

Additional Information: Oracle Applications Upgrade Manual

Set Environment Variables

Verify the settings of the environment variables `APPL_TOP`, `ORACLE_HOME`, `ORACLE_SID` or `TWO_TASK`, and `NLS_LANG` as described in the steps below before you unload the AutoInstall files.

Attention: If you log out of the `applmgr` account before running AutoInstall, make sure that these variables are set correctly when you log back in to the `applmgr` account to start AutoInstall.

In the applmgr Bourne shell or Korn shell, you can set an environment variable with this command:

```
$ <variable>=<value>; export <variable>
```

Do not include the environment variable prefix (\$) when defining the variable.

27. Verify SAPPL_TOP

Check that SAPPL_TOP is set to your applications top directory:

```
$ echo $APPL_TOP
```

If this variable is not set correctly, set the variable to the appropriate directory. Here is an example:

```
$ APPL_TOP=/d01/appl/110; export APPL_TOP
```

28. Set NLS environment variables

Before installing or upgrading, verify that your NLS environment variables are correctly set.

Additional Information: [Initialization Parameters](#) in Chapter 1; and National Language Support, *Oracle8 Reference*

The NLS_LANG environment variable has three parts: language, territory, and character set. The language and territory parts of NLS_LANG must be set to match NLS_LANGUAGE and NLS_TERRITORY. The NLS_LANGUAGE and NLS_TERRITORY database initialization parameters must be set to the American English values for AutoInstall purposes.

Set the character set part of NLS_LANG to a character set that your platform supports. In many cases, this will be the same as the database character set. You can use the following to determine the database character set:

```
$ sqlplus <SYS username>/<SYS password>
SQL> select VALUE "Character Set"
      2> from V$NLS_PARAMETERS
      3> where PARAMETER='NLS_CHARACTERSET' ;
```

The following example assumes that the NLS_LANGUAGE parameter is AMERICAN, the NLS_TERRITORY parameter is AMERICA, and the database was created with the WE8ISO8859P1 character set. In this case the NLS_LANG variable is set as follows:

```
$ NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1; export NLS_LANG
```

WARNING: You must set the NLS_LANG variable. If the variable is not set, AutoInstall will fail.

The NLS_DATE_FORMAT variable must be set to DD-MON-RR to support dates in the 21st Century:

```
$ NLS_DATE_FORMAT=DD-MON-RR ; export NLS_DATE_FORMAT
```

If the NLS_DATE_LANGUAGE or NLS_NUMERIC_CHARACTERS variables are set in your environment, you should ensure that they are set to the following values:

```
$ NLS_DATE_LANGUAGE=AMERICAN ; export NLS_DATE_LANGUAGE
$ NLS_NUMERIC_CHARACTERS="., " ; export NLS_NUMERIC_CHARACTERS
```

Note: The NLS_DATE_LANGUAGE and NLS_NUMERIC_CHARACTERS variables only need to be reset if they were previously set to other values in your environment.

You can reset your NLS variables after you install or upgrade.

Additional Information: [Reset NLS variables](#) in Chapter 4

Unload AutoInstall Files

You need to unload the installation files from CD to run AutoInstall.

The procedures in this section do not unload all the files required to install Oracle Applications products, localizations, or languages. AutoInstall unloads the remaining product files from the CDs when you start the installation or upgrade.

29. Unload installation directory from main CD

You first unload the installation directory (\$APPL_TOP/admin) from the main CD. Check the installation update for your platform for possible changes to the information provided in this step.

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

Most platforms support the following command in which *<device name>* is the operating system name of the CD reader and *<mount directory>* is a directory created as an alias for the reader:

```
$ mount -r <device name> <mount directory>
```

If you are the system administrator and logged in as a for UNIX super user (such as root) to mount the CD, log back into applmgr and set the APPL_TOP environment variable as described above. Change to the mount directory and run the unload.cmd script to unload the main Oracle Applications installation directory:

```
$ cd <mount directory>
$ unload.cmd
```

Here is an example:

```
$ mount -r /dev/dsk/c0t6d0s0 /cdrom
$ cd /cdrom
$ unload.cmd
```

The unload.cmd script asks the following question:

Do you wish to unload the installation directory for:

1. regular product group
2. localized product groups
- x. exit

Enter your choice [1] :

Choose option 1 to unload the installation directory for the regular product group.

30. Edit adovars.env file

The \$APPL_TOP/admin/advvars.env file sets certain environment variables. The file contains information about which variables it sets and what these settings do. Edit this file to define the JAVA_TOP, OA_JDK_TOP, CLASSPATH, OAH_TOP, and OAD_TOP variables, and add any customized settings to Applications variables.

Note: Settings in advvars.env will affect all Applications product groups using this APPL_TOP. If you wish to define any database-specific customizations, you will need to add logic to ensure it only executes for the relevant product group. You can also add it to the end of the main environment file, after the call to advvars.env, however you will need to re-apply this logic any time you regenerate the main environment file.

31. Run the AutoInstall setup script

Run the following setup script to unload the remaining files necessary to run AutoInstall from the CD:

```
$ cd $APPL_TOP/admin
$ adsetup
```

This script first prompts for the name of the machine that has the CD drive. Press [Return] if you are running the script on the machine that has the CD drive. The script then prompts for the drive name and asks you to confirm the unload. The adsetup utility creates two log files, adunload.log and adrelink.log, in \$APPL_TOP/admin/log.

The script unloads some files for Oracle Applications DBA (AD) and Oracle Application Object Library (FND). It also links AD programs with the Oracle8 Server libraries. The script takes approximately one hour to run. The actual time varies with platform and machine capacity.

Note: You cannot run multiple sessions of adsetup concurrently against multiple product groups sharing an APPL_TOP.

32. Unload localization installation directory (optional)

If you are installing Oracle Applications Localizations, you need to unload the localization installation directory. The Oracle Applications Localizations files are included on the main Oracle Applications CD. Choose option 2 from the unload.cmd utility screen to unload the localizations installation directory.

33. Apply pre-installation patches

If you want to apply any patches to fix problems with installation or upgrade scripts, you can apply these now by running AutoPatch in pre-AutoInstall mode. You must apply any pre-installation fixes for AD before you can apply fixes for other products.

Additional Information: [Running AutoPatch in Pre-AutoInstall Mode](#) in Chapter 5.

Attention: Check on any *Oracle Applications Release 11 Patch Sets* CD or check with Oracle Support Services for the most recent pre-installation fixes.

After you have applied any necessary AD pre-installation patches, you can apply fixes for any other products' installation scripts. If you have not yet unloaded your Applications files (by performing a previous Release 11 installation, for example), this will create directories and subdirectories for the products you are patching under the current \$APPL_TOP. When you run AutoInstall later to unload files, it will see that these files already exist and will not overwrite them with the older files from the Release 11 CD.

Attention: After applying pre-installation fixes for a product, if you plan to install that product's files on another disk, AutoInstall will not use the files you just unloaded. In this case, you will need to run AutoInstall to unload your files and create an environment file *first*, then run AutoPatch again in pre-AutoInstall mode to update the files *before* you install or upgrade your database.

Additional Information: [Installing Product Files \(File Tasks\)](#) in Chapter 3.

WARNING: Do not use the `adunload` command-line utility to unload Applications files. This will overwrite the files you just patched in pre-AutoInstall mode.

34. Run preparatory scripts

If you are upgrading your system from an earlier release, omit this task. You will run the preparatory scripts later in the upgrade process.

If you are installing a new Release 11, you should run the three preparatory scripts now. The three scripts are `addb804.sql`, `adsy804.sql`, and `adsysapp.sql`, and you must run them in this order. The `addb804.sql`, `adsy804.sql`, and `adsysapp.sql` scripts run Oracle8 Server and Oracle Applications administration scripts that AutoInstall requires. Each of the scripts creates a spool file, so make sure you have permission to write to the directory from which you run these scripts.

Attention: If your database is running on a separate machine from your administration server, copy the `addb804.sql` and `adsy804.sql` files to the `$ORACLE_HOME/dbs` directory on that machine and run them there. This ensures the scripts for the correct version of your database are run.

When Oracle Applications certifies with later versions of the Oracle8 Server, the first two scripts will have the newly-certified server version as part of the name. For example, if Oracle Applications certifies with Oracle8 Server version 8.0.5, the scripts will be named addb805.sql and adsy805.sql. For this reason, we use the script names addbxxx.sql and adsyxxx.sql in the following instructions.

The addbxxx.sql script sets up the data dictionary for Oracle Applications. It should be run in Server Manager while connected to INTERNAL:

```
$ cd $APPL_TOP/admin
$ svrmgrl
SVRMGR> connect internal
SVRMGR> @addbxxx
```

The adsyxxx.sql script creates SYSTEM objects and should be run in Server Manager as SYSTEM:

```
$ svrmgrl
SVRMGR> connect <SYSTEM username>/<SYSTEM password>
SVRMGR> @adsyxxx
```

The adsysapp.sql script creates Applications-specific objects in the SYSTEM schema, and should be run in SQL*Plus as SYSTEM, only after the previous two scripts have completed. Make sure you are in the \$APPL_TOP/admin directory before running this script. This script takes one argument, the password for the SYSTEM user:

```
$ cd $APPL_TOP/admin
$ sqlplus <SYSTEM username>/<SYSTEM password> @adsysapp.sql <SYSTEM password>
```

Note that the database may contain some invalid packages at this point. This is acceptable, as they will be validated when AutoInstall runs.

35. Run adsysctx.sql

Oracle Self-Service Web Applications and Oracle Service require the Oracle ConText Cartridge. If you plan to install Oracle Self-Service Web Applications or Oracle Service and have not installed the Oracle ConText Cartridge, you should install it now.

If you are not installing either of these two products, run the adsysctx.sql script. The adsysctx.sql script creates the ConText schema (CTXSYS) and several PL/SQL package specifications. The package specifications are not functional by themselves, however they must exist so AutoInstall can complete the install successfully, even though the Oracle ConText Cartridge is not installed.

Run the script from the \$APPL_TOP/admin directory in SQL*Plus, connected to the SYSTEM user. It takes three arguments: the SYSTEM password, the default tablespace to be used for the ConText schema (CTXSYS), and the temporary tablespace to be used for the ConText schema. For example, to create the ConText schema with default tablespace CTXD and tablespace TEMP, you would execute the script as follows:

```
$ sqlplus SYSTEM/MANAGER @adsysctx MANAGER CTXD TEMP
```

The tablespaces you specify must exist before you run this script. The ConText schema tablespace can be as small as you like if you are not using the Oracle ConText Cartridge; the temporary tablespace should ideally be the same as the one you use for the rest of Oracle Applications. If the CTXSYS schema already exists, or the ConText Cartridge is already installed, the script will not have any effect.

36. Apply ORACLE patches

If this is a fresh install, apply any patches that your platform-specific Installation Update says should be applied against the Oracle8 Server. If you are upgrading, skip this step and go on to Step 37.

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

Prepare for Upgrade

If you are installing a new system, omit steps 37 through 51 and proceed to Step 52 in the Set Up AutoInstall Environment section.

WARNING: Users should not attempt to access the database as you perform the following steps. They must sign off Oracle Applications and must not modify the database through tools such as SQL*Plus.

37. Back up Oracle Applications and customizations

Your operating system administrator should back up existing Oracle Applications product files and any product customizations. Your database administrator should back up the Oracle Applications database before you perform the category 2 upgrade steps.

Additional Information: *Oracle8 Backup and Recovery*, and Reapply Product Customizations in Chapter 5.

38. Perform category 2 upgrade steps

If you are upgrading existing products, you can perform category 2 upgrade steps after you unload the installation directory from the CD. You must perform these upgrade steps before you run AutoInstall.

Additional Information: *Oracle Applications Upgrade Manual*

39. Set up AutoInstall installation groups

If you are upgrading from Release 10.4 or Release 10.5, you must explicitly identify installation groups for AutoInstall before upgrading to Release 11. If you use Multi-Org, have only one set of books, or have only one product installation group, skip this step and proceed to Step 40.

You identify installation groups by running a SQL script, `ad1100.sql`, that creates a new column (`FND_PRODUCT_INSTALLATIONS.INSTALL_GROUP_NUM`) and then by ensuring it has the correct values. You must run `ad1100.sql` while connected to the AOL schema (`APPLSYS`).

```
$ cd $APPL_TOP/admin/preupg
$ sqlplus <AOL username>/<AOL password>
SQL> @ad1100.sql
SQL> exit;
```

Next, run the `adgupign.sql` script from the same `$APPL_TOP/admin/preupg` directory:

```
$ sqlplus <AOL username>/<AOL password>
SQL> @adgupign.sql
SQL> exit;
```

This script produces a SQL script in your current directory called `adupdign.sql`. The `adupdign.sql` script can be used to update the installation groups associated with each of the products in your database. Before running `adupdign.sql`, you must examine the statements listed for each product in this file, and make sure that the `INSTALL_GROUP_NUM` value shows the correct logical set of books for the given product and schema (production installation). If necessary, edit the file to assign the correct installation groups to schemas that are incorrect.

Install group numbers should be consecutive whole numbers, where 1 represents the first set of books (or first set of product installations), 2 is the second set of books, 3 is the third set of books, and so on. For example, if you have three sets of books that use a `<prod>_US`, `<prod>_UK`, and `<prod>_CA` naming convention, you

then identify that AP_US and PO_US belong in install group number 1, AP_UK and PO_UK belong in install group number 2, and AP_CA and PO_CA belong in install group number 3. You must ensure the values are correct for all products displayed.

Install group number 0 represents products that need only single installations (AD, ALR, AU, AX, DT, FA, FF, FND, GL, PAY, PER, RG). The script does not return these.

After you have made the necessary edits to `INSTALL_GROUP_NUM`, run the `adupdign.sql` script in SQL*Plus connected to the AOL schema:

```
$ sqlplus <AOL username>/<AOL password>
SQL> @adupdign.sql
SQL> exit;
```

40. Verify ORACLE IDs

If you originally installed Oracle Applications with Release 9.3 or prior, then you must ensure that the IDs for existing ORACLE schemas will not conflict with ones necessary for new schemas created by AutoInstall.

Run the following SQL script to identify potential conflicting ORACLE ID values:

```
$ cd $APPL_TOP/admin/preupg
$ sqlplus <AOL username>/<AOL password>
SQL> @adpuver.sql
```

Any ORACLE ID values listed by the script which are not associated with an APPS schema are potential conflicts which may result in a problem when upgrading to Release 11.

Attention: If you determine there is a potential conflict, contact Oracle Support Services.

41. Disable AOL Audit Trail feature

If you are using the AOL Audit Trail feature, you must disable it before running AutoInstall.

Additional Information: Disabling AuditTrail and Archiving Audit Data, *Oracle Applications System Administration Reference Manual, Release 10* or *Oracle Applications System Administrator's Guide, Release 11*

42. Disable custom triggers, constraints, and indexes

Disable any custom triggers or constraints on Oracle Applications tables as these may cause the upgrade to fail. You may re-enable these triggers after the upgrade. If you have any custom indexes on Oracle Applications tables, you should determine whether or not they will impact performance during the upgrade, and drop them if necessary. If you aren't sure, it is best to drop the indexes and add them again after the upgrade if the new release has not created a similar index.

Additional Information: [Re-enable custom triggers, constraints, and indexes](#) in Chapter 4

43. Drop event alert triggers in custom schemas

Drop all event alert database triggers created in custom schemas. These event alert triggers will be recreated later. To determine event alert triggers, connect to each custom schema and use the following SQL statement:

```
$ sqlplus <custom username>/<custom password>
SQL> select trigger_name from user_triggers
      2> where trigger_name like 'ALR%' ;
```

To drop these triggers:

```
SQL> drop trigger <trigger name> ;
```

44. Perform category 3 upgrade steps

Finish the category 3 upgrade steps for your Oracle Applications products. You must perform these upgrade steps before you run AutoInstall. These upgrade steps may render parts of your Oracle Applications products unusable until you finish the upgrade.

Additional Information: *Oracle Applications Upgrade Manual*

45. Shut down the concurrent managers

Shut down the concurrent managers before you begin the upgrade. To do this, use the Administer Concurrent Managers form on the System Administrator's menu. Navigate to the Control field in this form and select the deactivate option.

46. Back up the Oracle Applications database (recommended)

We recommend that you make a file system backup of the Oracle Applications database. If you encounter problems when running AutoInstall, you can use this backup to restore the database to its state just before running AutoInstall. If you must return to the old release, however, you should revert to the backup you made in Step 37.

Attention: If you do not make this backup and you need to recover the database and run AutoInstall again, you must use the database backup you created in Step 37 and perform all category 2 and category 3 upgrade steps again.

WARNING: Shut down the database with the **NORMAL** option. You may not be able to restore the database from the backup if you use the **IMMEDIATE** or **ABORT** option.

Additional Information: *Oracle8 Backup and Recovery*

47. Upgrade Oracle8 Server and Oracle Tools

If you need to upgrade the Oracle8 Server and the Oracle Tools to the versions certified for this release of Oracle Applications, do so now.

Additional Information: [Oracle8 Server and Oracle Tools Requirements](#) in Chapter 1

48. Run preparatory scripts

If you are upgrading your system from an earlier release, you should run the three preparatory scripts now. If you are installing a new Release 11, you already ran the preparatory scripts in Step 34.

The three scripts are `addb804.sql`, `adsy804.sql`, and `adsysapp.sql`, and you must run them in this order. The `addb804.sql`, `adsy804.sql`, and `adsysapp.sql` scripts run Oracle8 Server and Oracle Applications administration scripts that AutoInstall requires. Each of the scripts creates a spool file, so make sure you have permission to write to the directory from which you run these scripts.

Attention: If your database is running on a separate machine from your administration server, copy the addb804.sql and adsy804.sql files to the \$ORACLE_HOME/dbs directory on that machine and run them there. This ensures the scripts for the correct version of your database are run.

When Oracle Applications certifies with later versions of the Oracle8 Server, the first two scripts will have the newly-certified server version as part of the name. For example, if Oracle Applications certifies with Oracle8 Server version 8.0.5, the scripts will be named addb805.sql and adsy805.sql. For this reason, we use the script names addbxxx.sql and adsyxxx.sql in the following instructions.

The addbxxx.sql script sets up the data dictionary for Oracle Applications. It should be run in Server Manager while connected to INTERNAL:

```
$ cd $APPL_TOP/admin
$ svrmgrl
SVRMGR> connect internal
SVRMGR> @addbxxx
```

The adsyxxx.sql script creates SYSTEM objects and should be run in Server Manager as SYSTEM:

```
$ svrmgrl
SVRMGR> connect <SYSTEM username>/<SYSTEM password>
SVRMGR> @adsyxxx
```

The adsysapp.sql script creates Applications-specific objects in the SYSTEM schema, and should be run in SQL*Plus as SYSTEM, only after the previous two scripts have completed. Make sure you are in the \$APPL_TOP/admin directory before running this script. This script takes one argument, the password for the SYSTEM user:

```
$ cd $APPL_TOP/admin
$ sqlplus <SYSTEM username>/<SYSTEM password> @adsysapp.sql <SYSTEM password>
```

Note that the database may contain some invalid packages at this point. This is acceptable, as they will be validated when AutoInstall runs.

49. Turn off automatic archive logging

If the Oracle8 Server is set up to automatically archive redo log files, turn off archiving with the Oracle Server Manager command ARCHIVE LOG STOP. This reduces the amount of disk space and shortens the time required by the upgrade. If you need to recover to an earlier point, you can restore the database backup from Step 37 and rerun AutoInstall. This is faster than recovering data through archived log files.

50. Verify rollback segment sizing for upgrades

Check that your rollback segments are sufficiently large to handle the upgrade data volume. They should be configured to avoid an ORA-01555 "Snapshot too old" error.

Additional Information: [Rollback Segments](#) in Chapter 1

51. Delete old Oracle Applications files (optional)

If you do not have enough disk space for new Oracle Applications product files, you can delete any product files belonging to older releases. For each product, you may want to retain report output files or customized programs.

Output files are stored in the log and out subdirectories under each product's top directory or under the common directory designated by the environment variable APPLCSF.

Additional Information: [Product Directory Structure and File Types](#) in Appendix A

Set Up AutoInstall Environment

Perform the following steps to set up your AutoInstall environment just before you run AutoInstall.

52. Restrict access to log and restart files

Some AutoInstall output, such as log and restart files, contains passwords to ORACLE accounts related to Oracle Applications, including the SYSTEM account. We recommend that you restrict access to the admin directory that contains these files if Applications users will have access to the administration server.

```
$ cd $APPL_TOP
$ chmod 700 admin
```

Suggestion: Retain log and restart files online until you have implemented your products and tested them thoroughly. Oracle Support Services may need access to these files if you encounter problems implementing or using your system.

53. Turn off Oracle8 Server Parallel Query Option

The Oracle Applications installation or upgrade process is incompatible with the Oracle8 Server Parallel Query Option. Disable Parallel Query Option before you run AutoInstall. To disable Parallel Query Option, the database administrator should comment the following parameters from the init.ora parameter file:

- parallel_default_max_scans
- parallel_default_max_instances
- parallel_default_scansize
- parallel_min_servers
- parallel_max_servers
- parallel_server_idle_time

These parameters are set to their default values if they are not defined in the init.ora file. The default values disable Parallel Query. Restart the database if you change any parameters.

Additional Information: Initialization Parameter Files, *Oracle8 Reference*; Tuning Resource Contention, Tuning Parallel Execution, *Oracle8 Tuning*

54. Check SQL*Plus options

As part of the Oracle8 Server requirements, you should have set the SQL*Plus PAUSE option OFF and the NEWPAGE option to a value other than 0 (zero). Connect to SQL*Plus with any ORACLE user ID to verify these options, such as:

```
$ sqlplus system/manager
SQL> show pause
pause is OFF
SQL> show newpage
newpage 1
SQL> exit
```

If PAUSE is ON or NEWPAGE is 0, change these values in your applmgr SQL*Plus login file before you run AutoInstall.

Additional Information: [SQL*Plus Options](#) in Chapter 1

55. Run AutoInstall environment file

Run the following file to export to your environment the variables necessary to run AutoInstall:

```
$ cd $APPL_TOP/admin  
$ . adsetenv
```

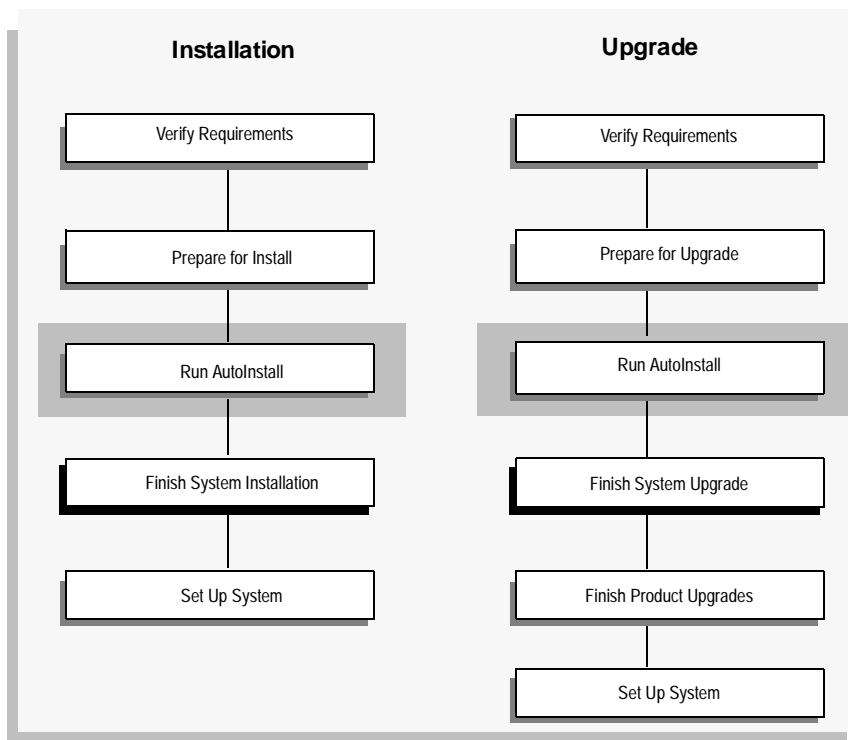
Using AutoInstall

This chapter describes how to run AutoInstall. It explains the questions that AutoInstall asks when it starts up and tells how to use the AutoInstall screens to configure your installation or upgrade. This chapter contains these sections:

- [Your Place in the Installation or Upgrade Process](#)
- [Running AutoInstall \(adaimgr\)](#)
- [AutoInstall Questions](#)
- [Using AutoInstall Menus and Screens](#)
- [Running the Installation or Upgrade](#)
- [Creating an Environment File](#)
- [Monitoring AutoInstall](#)
- [After Running AutoInstall](#)

Your Place in the Installation or Upgrade Process

The following figure illustrates your current place in the Oracle Applications installation or upgrade process. Follow the instructions in this chapter to carry out the step highlighted in this figure.



Running AutoInstall (adaimgr)

Once you have set up your environment, you can start AutoInstall from any directory with this command:

```
$ adaimgr
```

AutoInstall displays an introduction screen and asks questions about your installation or upgrade.

Note: Although AutoInstall can be used with non-graphic terminal devices, we recommend using a window-based terminal when running AutoInstall so you can monitor and control AutoInstall processes in other windows while AutoInstall is running.

Stopping AutoInstall

You can exit from the AutoInstall menus and screens by entering this command at any AutoInstall prompt:

```
abort
```

After you exit, you can restart AutoInstall from where your last session ended or from the beginning.

WARNING: Once AutoInstall begins to install or upgrade products in the database, do not attempt to stop it without first contacting Oracle Support Services.

Attention: If an error occurs while AutoInstall is installing or upgrading the database, you can correct the error without stopping AutoInstall.

Additional Information: [Correcting AutoInstall Errors](#) in this chapter

Restarting AutoInstall

If you logged out of the applmgr account after you stopped AutoInstall, be sure that your environment is set up properly before you restart AutoInstall. Do this by logging back in to applmgr and resetting your environment.

Additional Information: [Set Up AutoInstall Environment](#) in Chapter 2

Restart AutoInstall by re-issuing the command `adainmgr`. When AutoInstall prompts for the name of the log file, specify the file from the previous session or supply a new filename. When you reuse a log file, AutoInstall adds the message "Start of AutoInstall Session" to the end of the log file and appends messages from the new session as it generates them.

Additional Information: [Log and Restart Files](#) in this chapter

AutoInstall then asks if you want to continue with your previous session or start a new one.

Continue Session

Continuing the previous session is the default. When you choose to continue, AutoInstall determines where your last session stopped and restarts at that point. It retains all configuration information you entered in the AutoInstall screens in your last session.

Attention: If the machine failed while AutoInstall was running the installation or upgrade, the AutoInstall restart files may have been corrupted. Contact Oracle Support Services if you encounter discrepancy problems when restarting.

Start New Session

AutoInstall asks you to confirm your choice if you choose not to complete the previous session. It then restarts from the beginning. This resets the AutoInstall configuration screens, so you will have to reenter any previous configuration information that you want to retain.

WARNING: Do not restart AutoInstall from the beginning if it began to install or upgrade products in the database and then stopped. If this happens, determine why AutoInstall stopped, correct the problem, and restart your previous session. You can also restore the last saved database and filesystem, then start the process again from the beginning.

Running AutoInstall in a Multi-Tier Environment

If you are installing Oracle Applications in a multi-tier environment, you must run AutoInstall on every machine to install the necessary files. You only need to run AutoInstall once, on the administration server, to install or upgrade database objects. You may choose to install the database objects on the administration server first, then run AutoInstall on every other server and run the necessary tasks. Or you may choose to do all AutoInstall file-related tasks on each server in turn, then the database tasks on the administration server, and then the generation tasks on all other servers in turn.

Attention: You can not run multiple sessions of AutoInstall simultaneously against a product group. You may run multiple sessions of AutoInstall on separate machines, but only one may be performing database or generation tasks.

Additional Information: [Tasks Screen](#) in this chapter

AutoInstall Questions

When it starts, AutoInstall asks initial questions about your system. The following sections explain these questions and your possible responses. If you restart an AutoInstall session that did not run to completion, AutoInstall may not ask all questions.

When AutoInstall prompts for information, it typically includes a default answer in square brackets:

```
Please enter your AutoInstall log file name or press  
[Return] to accept the default file name shown in brackets.
```

```
Filename [adaimgr.log] :
```

You can accept the default value (adaimgr.log) by pressing [Return]. To specify a new value, type the value and press [Return].

Attention: Read the AutoInstall screen prompts carefully to be sure you supply the correct information.

Applications Top Directory

AutoInstall displays the value of the environment variable `APPL_TOP` and asks if this is the top directory for the new Oracle Applications product files. If the directory is not correct, answer No at the prompt to exit. Resume running AutoInstall after you set `$APPL_TOP` to the correct directory.

Additional Information: [Verify \\$APPL_TOP](#) in Chapter 2

AutoInstall Log File

AutoInstall prompts for a name of the log file that records the entire AutoInstall session. The default filename is `adainmgr.log`. AutoInstall places the log file in the directory `$APPL_TOP/admin/<dbname>/log`, where `<dbname>` is the value of your `ORACLE_SID` or `TWO_TASK` variable. You need to review the log file for error messages after you run the installation or upgrade.

Software Compatibility

AutoInstall asks you to verify software compatibility:

Are you certain you are running a certified release combination [No] ?

The release versions of Oracle Applications, the RDBMS, and the related tools must be compatible. Refer to the *Oracle Applications Installation Update* for your platform to verify that Oracle Applications will run in the current environment. If you are uncertain whether a release combination is certified, contact Oracle Support Services.

Email Notification

You can choose to be notified by email if an AutoInstall worker fails at any time during the installation process. If you select this option, AutoInstall will prompt you to enter your email address, e.g.: `appldba@mycompany.com`). If you want more than one person to be notified, enter each email address separated by a space. The email notification will contain the last 100 lines of the worker log (that failed).

Batch Size

AutoInstall prompts you to enter a batch commit size to be used during the upgrade. If you do not specify a value, it uses a default batch commit size, which is set to a relatively small value to accommodate systems with small rollback segments.

Batch commit size determines the number of rows to commit at one time when certain scripts run. To take advantage of large rollback segments, you must specify a batch commit size larger than the default value.

Installing Files in APPL_TOP

A given APPL_TOP may contain files for one or more servers—a forms server, an administration server, a web server, or a concurrent processing server. If you have the files associated with a particular server installed in your APPL_TOP, the machine can perform the tasks associated with that type of server. For example, if you have the files needed for installing or upgrading the database installed in this area, you can install or upgrade the database from this APPL_TOP, and this machine will be an administration server.

The following table summarizes the AutoInstall questions and possible responses:

AutoInstall Question: "Do you currently have or want to install..."	A Yes response means that...
...files used for installing or upgrading the database?	the APPL_TOP will be part of an administration server.
...Java and HTML files for HTML-based functionality?	the APPL_TOP will be part of an Applications web server. *
...Oracle Applications forms files?	the APPL_TOP will be a part of a forms server.
...concurrent program files?	the APPL_TOP will be a part of a concurrent processing server.
* The AutoInstall prompt may read "...Java and HTML files for Self-Service Web Applications"; however, these files are used for other HTML-based functional components, such as the Workflow Monitor, and document attachments.	

ORACLE Database

AutoInstall lists the database and database directory you are using. It asks you to confirm that you want to install or upgrade Oracle Applications in this database with the files in this directory. Here is an example:

```
You are about to install or upgrade Oracle Applications
product tables in your ORACLE database 'apptest' using
ORACLE executables in '/d01/app/oracle/prod/8.0.4'.
```

```
Is this the correct database [Yes] ?
```

If AutoInstall does not list the correct database and directory, answer No at the prompt to exit. Resume running AutoInstall after you change the values of \$ORACLE_SID or \$TWO_TASK, and \$ORACLE_HOME to identify the correct database and directory.

Additional Information: Verify Environment Variables in Chapter 2

AutoInstall Action

AutoInstall must determine whether to perform a new installation or update an existing installation.

AutoInstall prompts for your ORACLE SYSTEM schema password, then connects to the schema. AutoInstall then attempts to read the FND_PRODUCT_INSTALLATIONS table in the database. If the table exists, AutoInstall notifies you that it will upgrade the existing product group. If the table does not exist, AutoInstall will perform a new installation of Oracle Applications.

Note: You cannot have more than one product group per database instance. If AutoInstall finds more than one product group, it will stop with a fatal error.

If AutoInstall is performing a new installation, it assigns a default username and password for the Oracle Application Object Library. The default username/password is APPLSYS/APPS. You can change your password after the installation.

If you are upgrading from an earlier release, AutoInstall asks for the ORACLE password for your Oracle Application Object Library schema. AutoInstall then connects to this schema to read information about the current installation.

Temporary Tablespace

If AutoInstall is performing a new installation, you must enter the temporary tablespace to be used by all Applications schemas at this point. If you are upgrading from an earlier release, AutoInstall does not ask this question. For products that are already installed, AutoInstall will use the temporary tablespace settings from your previous release; for new products, AutoInstall will use the temporary tablespace defined for the Application Object Library schema.

Organization Type

AutoInstall prompts you to indicate whether you use Oracle Applications in a commercial organization. The questions differ depending on whether you are installing or modifying a system.

When Installing a System

If you are installing an Oracle Applications system, AutoInstall asks you what type of organization you have:

1. Oracle Applications for a commercial or for-profit organization
2. Oracle Applications for a government, education or not-for-profit organization

Choose option 1 to install the commercial versions of Oracle Financials products and/or any other licensed Oracle Applications products. Choose option 2 to install Oracle Public Sector Financials products and/or any other licensed Oracle Applications products.

When Upgrading or Modifying a System

If you are upgrading an existing product group, AutoInstall automatically determines whether your products are installed for commercial use. If you installed the commercial versions of products, AutoInstall prompts you to select one of these options:

- 1) Continue to use Oracle Applications for Commercial or for-profit use.
- 2) Convert Oracle Applications to government, education or not-for-profit use.

Choose option 2 to convert commercial Oracle Financials products to Oracle Public Sector Financials products during the upgrade.

AutoInstall does not ask these questions if you previously installed public sector, education, or not-for-profit products. Once not-for-profit products are installed, they cannot be converted to commercial products.

Territory and Language

AutoInstall asks you about the language you plan to use. The questions differ depending on whether you are installing or modifying a system. Table A-5 in Appendix A lists the language codes.

Attention: Release 11 of Oracle Applications supports operation in only one language. If you need to configure your system to support multiple languages, contact Oracle Consulting Services.

When Installing a System

If you are installing an Oracle Applications system, AutoInstall asks for the language you want to use:

1. Install AMERICAN only
2. Install one language other than AMERICAN
3. Install multiple languages (requires Oracle Consulting Services support)

If you choose option 2 or option 3, AutoInstall prompts for language codes. For example, you must supply the language code F if you plan to install Oracle Applications in French. Do not include US when specifying language codes because American English is installed by default. AutoInstall then asks you to specify the base language. The base language is the language of the Oracle Applications seed data.

When Upgrading or Modifying a System

AutoInstall automatically displays the language or languages that are currently installed and the base language. The subsequent AutoInstall questions ask you to update the list of installed languages. The prompts will differ depending on whether you currently have one or more than one language installed.

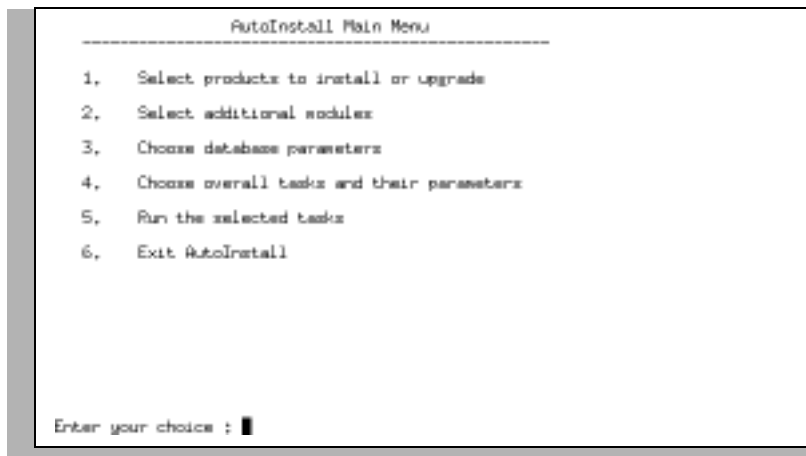
You can install an additional language or languages. You can also choose to deactivate a currently installed language (except American English, which cannot be deactivated). After updating your language information, AutoInstall asks you to specify which language will be the base language.

Using AutoInstall Menus and Screens

The user interface for AutoInstall consists of a main menu and several screens that present installation information. Review this section to see how these screens work.

Main Menu

AutoInstall displays the Main Menu after you answer the initial AutoInstall questions. This menu presents the options you use to configure and run your installation or upgrade.



Type the option number to select an option. You should generally choose the options in numeric order. The "Choose database parameters" option only appears if you are installing or upgrading on an administration server.

1: Select products to install or upgrade Choose this option to install the Oracle Applications products you licensed or to change the default upgrade actions for your existing products.

2: Select additional modules Choose this option to install localized modules for Oracle Applications products.

3: Choose database parameters Choose this option to set the ORACLE user ID (schema name), sizing factor, and tablespaces for each Oracle Applications product. This option only appears if you are installing or upgrading on an administration server.

4: Choose overall tasks and their parameters With this option, you specify the tasks that AutoInstall performs. These tasks include actions such as unloading new product files, linking executable programs with the Oracle8 Server, and installing database objects.

5: Run selected tasks Choose this option to start the installation or upgrade. Do this only after you have configured the installation or upgrade with the preceding options.

6: Exit AutoInstall Select this option to exit AutoInstall and complete your AutoInstall session. If you have not completed all the tasks needed for the type of server installation you are performing, save the configuration information so you can run AutoInstall again later without re-entering all information.

About the AutoInstall Installation Screens

All AutoInstall screens present installation information at the top and list your options at the bottom. The cursor appears at the end of the command line.

Installation Information

Options

Command Line

AutoInstall - Select products to install or upgrade

Product # Name	ORACLE User ID	Current Status	Current Version	New Version	Translate Convert	Action	I	I
1 Application Object Library	APPLSYS	Installed	6.1.1	11.0.28	Upgrade	N	N	
2 Application Utilities	APPLSYS	Shared	1.10.7	11.0.28	Upgrade	N	N	
3 Applications DBA	APPLSYS	Shared	2.3.19	11.0.28	Upgrade	N	N	
4 Oracle Alert	APPLSYS	Installed	5.0.29	11.0.28	Upgrade	N	N	
5 Global Accounting Engine	AK	Installed	1.2.23	11.0.28	Upgrade	N	N	
6 Oracle Common Modules-AK	AK	Installed	1.2.19	11.0.28	Upgrade	N	N	
7 Oracle General Ledger	GL	Installed	9.0.9	11.0.28	Upgrade	N	N	
8 Application Report Generator	RG	Installed	4.0.9	11.0.28	Upgrade	N	N	

There are 48 Oracle Applications. Enter U/D to scroll up/down.

<Product #> - To change the action for a product
<Product #>D - To change the product dependencies or other details
U / D / T / B - Press up/down/top/bottom to see other products
[Return] - To return to the AutoInstall Main Menu

Enter your choice (for example 5 or 5D) :

Selecting Options

You select an option by entering the letter, number, or combination of number and letter that designates the option. AutoInstall displays an error message if your choice is invalid. When you select a valid option, AutoInstall carries out an action, prompts for more information, or displays another screen.

For example, in the screen example, Oracle General Ledger is listed as product number 7. To change the dependencies and details for Oracle General Ledger, you press 7, the letter D, and then [Return]. AutoInstall then displays the product's Product Detail Screen.

In most screens you can enter multiple values using a space to separate, for example:

7 9 21

toggles the install action for these three products.

Viewing Additional Screen information

When an AutoInstall screen contains additional information, select the U or D option to scroll up or down through the list of information. Use the T or B option to scroll to the top or bottom of the list.

For example, the screen in the example lists eight Oracle Applications products. To display information about products 9 through 16, you press D and [Return] to scroll down the list. You press B and [Return] to scroll to the bottom of the list and display the last eight products.

Returning to the Previous Screen

Press just [Return] to exit an AutoInstall screen and return to the previous screen. This verifies and saves any changes you have made.

Select Products Screen

The Select Products screen lists the Oracle Applications products and indicates what actions AutoInstall will perform for each product.

```

AutoInstall - Select products to install or upgrade

Product      ORACLE   Current   Current   New        Translate
# Name       User ID   Status    Version  Version    Action    I  I
-----
1 Application Object Library  APPLSYS  Installed 6,1,1    11,0,28  Upgrade  N  N
2 Application Utilities       APPLSYS  Shared    1,10,7    11,0,28  Upgrade  N  N
3 Applications DBA            APPLSYS  Shared    2,3,19    11,0,28  Upgrade  N  N
4 Oracle Alert               APPLSYS  Installed 5,0,29    11,0,28  Upgrade  N  N
5 Global Accounting Engine    AX       Installed 1,2,23    11,0,28  Upgrade  N  N
6 Oracle Common Modules-AK    AK       Installed 1,2,19    11,0,28  Upgrade  N  N
7 Oracle General Ledger       GL       Installed 9,0,9     11,0,28  Upgrade  N  N
8 Application Report Generator RG        Installed 4,0,9     11,0,28  Upgrade  N  N

There are 48 Oracle Applications. Enter U/D to scroll up/down.

<Product #> - To change the action for a product
<Product #>D - To change the product dependencies or other details
U / D / T / B - Press up/down/top/bottom to see other products
[Return] - To return to the AutoInstall Main Menu

Enter your choice (for example 5 or 5D) : █

```

Screen Columns

The columns of the Select Products screen list the basic information about the Oracle Applications products you can install or upgrade.

Lists the product numbers. You use these numbers to change the installation actions or product details. When installing or upgrading multiple copies of a product, check the ORACLE User ID column to ensure that you are selecting the appropriate copy. You can also type *all* at the prompt to select all products.

Product Name Lists the names of the Oracle Applications products. If you have multiple copies of a product, the copies have the same name but different ORACLE user IDs.

ORACLE User ID Lists the ORACLE schema for each product. Each copy of a product has a unique schema. For example, two copies of Oracle Payables may have schema names of AP and AP2. You can change user IDs for new product installs through the Database Parameters screen.

Additional Information: [Database Parameters Screen](#) in this chapter

Current Status and Current Version Lists Installed for each fully installed product and Shared for each dependent product. The Current Version column lists the product version numbers. These columns are blank for products that are not currently licensed.

New Version Lists the version numbers for the Oracle Applications products in the new software release.

Action Indicates what AutoInstall will do with the product during the installation or upgrade. An explanation of what these installation actions mean and how to change them appears in the following paragraphs.

Convert Automatically set to Y if you will install or upgrade products for public sector, education, or not-for-profit use, and cannot be changed on this screen. This flag is also used in a special way for the Oracle Projects product, to indicate a conversion from Project Costing to Project Billing.

Translate Indicates if AutoInstall is installing a language other than American English. This column's value is set by the language you choose earlier in the AutoInstall questions, and cannot be changed on this screen.

Setting Installation Actions

Enter the product number to change a product's installation action. On a new installation, select the products you have licensed. AutoInstall changes their action to Install and automatically sets the actions of their dependent products to Shared. On an upgrade, AutoInstall sets products to the Upgrade action automatically.

The following entries summarize the installation actions:

- None

AutoInstall does not install runtime files for the product if the action is None. When you select a product currently set to None, AutoInstall changes the action to Install. It also changes the actions of the product's dependent products to Shared if they are set to None. If the product requires a full installation of other products, these other products have their action changed to Install.

- Install

AutoInstall fully installs a product that is set to Install. When you select a product currently set to Install, AutoInstall changes the action to None or Shared, depending on whether other products need it as a dependent product. AutoInstall also changes the product's dependent products to None if they are not shared by other products currently set to Install.

WARNING: Make sure that you set only the products you have licensed to the Install action. You cannot de-install a product once it is installed.

- Shared

AutoInstall sets this action automatically when the product is set to None and another product you set to Install uses the product as a dependent product. You can fully install a product set to Shared by selecting the product. You cannot directly change the action from Shared to None or from None to Shared.

- Upgrade

When you are upgrading an existing product group, AutoInstall automatically sets this action for products not yet upgraded to the current release. You cannot change an Upgrade action by selecting the product.

Changing Select Products Screen Information

When you are installing, change the installation action to Install for your licensed products by entering the product numbers.

When you are upgrading, be sure to:

- Verify that AutoInstall has set the Action, Convert, or Translate columns correctly for each product.
- Change the action to Install for any additional products you have licensed in this release.
- Verify through the Product Detail screens that dependencies for multiple copies of a product are set correctly. You display a Product Detail Screen by entering the product number and D.

WARNING: AutoInstall will fail during the upgrade if the dependencies have not been set correctly for every product in each product installation group.

Additional Information: [Set up AutoInstall installation groups](#) in Chapter 2

Upgrading Multiple Sets of Books

In Release 11, you no longer install multiple sets of books as in earlier releases. If you need multiple sets of books functionality, you must use the Multiple Organization architecture instead.

Additional Information: Multiple Organization Architecture, *Oracle Applications Concepts*

Note: If you installed multiple sets of books in an earlier release, you can upgrade your existing multiple sets of books database. You can also add sets of books, as in the earlier release. Please note that the multiple sets of books architecture used in earlier releases is functionally different than Multi-Org architecture.

Although the Select Products screen no longer shows option A for additional products, the option is still available for compatibility with earlier releases.

You can use the A option to add another set of books within the product group. When you select the A option, AutoInstall prompts you to confirm the action, then copies the MOA (Multiple Oracle Account) products for another set of books.

The copied products display at the bottom of the Select Products Screen. If there are 48 products listed in the Select Products Screen before you select the A option, the first copied product displays as product number 49.

Note: You cannot install a localization against a particular set of books because the localization is applied to all sets of books.

AutoInstall assigns sequential user IDs (usernames and passwords) to the copied products. For example, it assigns the ID INV2/INV2 to the first copy of Oracle Inventory, INV3/INV3 to the second copy, and so on. You can change these default IDs through the Database Parameters screen.

Additional Information: [Database Parameters Screen](#) in this chapter

AutoInstall initially sets the installation action for each copy of a product to None. Change the installation action to Install as described above to install a product in a set of books. Leave the installation action set to None on products for which you do not want to install additional copies.

Product Detail Screens

Each copy of an Oracle Applications product has a Product Detail screen where you can set specific installation information for that product. You access these screens from the Select Products screen.

Product: Oracle Bills of Material		ORACLE User ID: BOM	
Put files in: /d1/appl/110			
#	Dependent Product Name	Oracle User ID	
1	Application Object Library	APPLSYS	
2	Global Accounting Engine	AX	
3	Oracle Common Modules-AK	AK	
4	Application Implementation	AZ	
5	Oracle General Ledger	GL	
6	Oracle Inventory	INV	
7	Oracle Purchasing	PO	
8	Oracle Payables	AP	
9	Oracle Assets	FA	
10	Oracle Receivables	AR	
#	Dependent Product Name	Oracle User ID	
11	Oracle Order Entry	OE	
12	Oracle Sales and Marketing	OSM	
13	Oracle Human Resources	HR	
14	Oracle Payroll	HR	
15	Oracle Mfg Menu	MFG	
16	Oracle Engineering	ENG	
17	Oracle Master Scheduling/MRP	MRP	
18	Oracle Capacity	CRP	
19	Oracle Work in Process	WIP	
20	Oracle Projects	PA	
<p>F - To change the directory where AutoInstall should put the files</p> <p>[Return] - To return to the AutoInstall Select Products Menu</p> <p>Enter your choice : █</p>			

When you are upgrading, use this screen to verify product dependencies for product installation groups. You can use the F option to change the directory where AutoInstall unloads the product files for either a first-time installation or an upgrade.

Changing Product Detail

The F option is the only one available unless you are upgrading a multiple set of books architecture product group.

Option F: Change Directory for Product Files By default, AutoInstall unloads the product's files under the Applications top directory, \$APPL_TOP. You can use the F option to unload the product files under a different directory. For example, you could specify this directory for Oracle Bills of Material:

```
/d2/mfg/110
```

During the installation or upgrade, AutoInstall would create this directory (if it did not exist already) and also create the product and version directories (/bom/11.0.28). It would unload the product files under the version directory (11.0.28) and set the BOM_TOP environment variable to the entire path:

```
/d2/mfg/110/bom/11.0.28
```

Note: AutoInstall will not create the disk mount point (/d2 in these examples). You must create the mount point before AutoInstall attempts to create the directory.

Additional Information: [Create Directories](#) in Chapter 2

Because all copies of a product use the same set of product files, you need to specify the new directory for only one copy of a product. AutoInstall automatically sets this path in the Product Detail screens for all other copies of the product.

Product #: Change Dependent Product IDs To change which copy of a dependent product the product uses, enter the number of the dependent product from the list at the top of the screen. Then enter the ORACLE user ID for another copy of the dependent product. This copy must have been created already. Note the following when changing the dependent product IDs:

- Each dependent product must correspond to the product installation group that holds the product. For example, a copy of Oracle Purchasing in a set of books

must use as a dependent product a copy of Oracle Payables in that same set of books.

- You cannot change the dependent product user ID for these products:
 - DateTrack
 - FastFormula
 - Global Accounting Engine
 - Oracle Alert
 - Oracle Application Object Library
 - Oracle Applications DBA
 - Oracle Self-Service Web Applications
 - Oracle Applications Report Generator
 - Oracle Applications Utilities
 - Oracle Assets
 - Oracle Common Modules
 - Oracle General Ledger (commercial or public sector)
 - Oracle Human Resources
 - Oracle Payroll
 - Oracle Quality
 - Oracle SSP/SMP
 - Oracle Training Administration

Option C: Create Another Entry The C option adds another entry for the product to the Select Products screen. When upgrading, you may need to use this option to associate multiple copies of dependent products with multiple Sets of Books.

Additional Information: [Upgrading Multiple Sets of Books](#), and [Verifying Dependencies in Multiple Sets of Books](#), in this chapter

Attention: Do not use this option to add a sets of books when upgrading from an earlier release. Use the A option from the Select Products Screen instead. The A option automatically copies the necessary products and sets the correct product dependencies.

When you choose the C option, AutoInstall prompts for an ORACLE username and password. The username must be unique. The bottom of the Select Products Screen then lists the new entry for this product. The installation action is initially set to None.

Option D: Delete Entry Select the D option to delete this entry of the product from the list of products in the Select Products Screen. You can delete an entry only on these conditions:

- You have not installed the product previously.
- The installation action is None.
- The Select Products Screen lists another entry for this product.

Verifying Dependencies in Multiple Sets of Books

When you upgrade a multiple sets of books installation, check the dependent products for each fully installed product in each set of books. Verify that each product's dependent products are in the same set of books as the installed product. For example, if Oracle Purchasing is installed in a certain set of books, its dependent product Oracle Payables must be installed in that set of books as well.

A product installed in your sets of books may acquire a new dependent product in a new release. If that dependent product must be in a set of books that you are upgrading and AutoInstall did not copy the product to the set of books automatically, you must copy the product manually and associate the copy with the fully installed products in that set of books.

Additional Information: Table 1–1, Dependent Products in Chapter 1

You do not have to copy DateTrack, FastFormula, Global Accounting Engine, Oracle Alert, Oracle Application Object Library, Oracle Applications DBA, Oracle Applications Report Generator, Oracle Applications Utilities, Oracle Assets, Oracle Common Modules, Oracle General Ledger (commercial or public sector), Oracle Human Resources, Oracle Payroll, Oracle Quality, Oracle Self-Service Web Applications, Oracle SSP/SMP, or Oracle Training Administration for use as dependent products. A product that shares one of these products can use the main installation of that product.

AutoInstall warns you about problems with product dependencies. For example, you may see a message like the following when you exit the Select Products screen:

AutoInstall error:

You have chosen that both PO2 and PO depend on AP, where
PO2 is a copy of Oracle Purchasing,
PO is a copy of Oracle Purchasing,
and AP is a copy of Oracle Payables.

In this case, you need to create a new entry for Oracle Payables and make it a dependent product of PO2. If AP2 already exists, you can go on to Step 4.

1. Display the Product Detail Screen for Oracle Payables (AP).
2. Select the C option and choose a new ORACLE user ID such as AP2.
3. Return to the Select Products screen.
4. Display the Product Detail screen for the second entry of Oracle Purchasing (PO2). Enter the product number for Oracle Payables (AP) from the list at the top of the screen. When prompted, enter AP2 as the new ID. This sets the second entry of Oracle Purchasing (PO2) to use the second entry of Oracle Payables (AP2) as a dependent product.

Attention: You can often avoid errors such as these by properly assigning the installation group numbers in advance. You can still do this now using the scripts in Step 39 of Chapter 2. However, if you make changes after running the scripts at this point, you must restart AutoInstall “from scratch” (that is, restart AutoInstall and do not choose to use existing restart files).

Select Modules Screen

The Select Modules screen lists the Oracle Applications localization modules and indicates what actions AutoInstall will perform.

AutoInstall - Select modules to install or upgrade						
#	Territory Name Localization Module	Current Status	Current Version	New Version	Translate Convert	I
1	ARGENTINA					
	Argentina Localizations			11.0.28	None	N N
	Regional Localizations			11.0.28	None	N N
2	AUSTRIA					
	Austria Localizations			11.0.28	None	N N
	Regional Localizations			11.0.28	None	N N
3	AUSTRALIA					
	Australia Localizations			11.0.28	None	N N
	Regional Localizations			11.0.28	None	N N

There are 49 territories. Enter U/D to scroll up/down.

<Territory #> - To change the actions for modules in a territory
 R - Reset modules to None except for the territory you are in
 U / D / T / B - Page up/down/top/bottom to see other territories/modules
 [Return] - To return to the AutoInstall Main Menu

Enter your choice : █

You select the localization modules you want to install on this screen, not the Select Products screen. You use the Product Detail screen to control where AutoInstall unloads the localization. For example, you determine where the JL (Latin America) localizations are unloaded on the Product Detail screen, then whether to install the Argentina Localizations on the Select Module screen.

Select the R option to reset the installation action to None for all modules except those that are already installed.

Additional Information: [Select Products Screen](#), and [Product Detail Screens](#) in this chapter

When you are upgrading, use this screen to verify that AutoInstall has set the Upgrade action correctly for each module. Whether installing or upgrading, you can use this screen to enter the territory number to install the Localization modules (if any) that you require.

Screen Columns

The columns of the Select Modules screen list the basic information about the localization modules.

The # column lists the module numbers. Select the territory # to install the Localization modules required.

Territory Name - Localization Module The Territory Name - Localization Module column lists the names of the Oracle Applications modules by territory.

Current Status and Current Version The Current Status column lists Installed for each fully installed module. The Current Version column lists the module version numbers. These columns are blank for modules that are not installed. Modules are not installed as shared.

New Version The New Version column lists the version numbers for the localization modules in the new software release.

Action The Action column tells what AutoInstall will do with the module.

Convert The Convert column is always set to N in Release 11.0 and cannot be reset. This column is reserved for compatibility with future releases.

Translate The Translate column is set to Y if you are installing a translation.

Database Parameters Screen

Use the Database Parameters screen to control the placement and sizing factor of products in the Oracle8 Server database.

```

AutoInstall - Choose database parameters

  Product      Action  - O -   - S -   --- M ---   --- I ---   --- D ---
# Name         I      ORACLE Sizing Main   Index   Default
                I      User ID Factor Tablespace Tablespace Tablespace
-----
1 Application Object Lib I  APPLSYS  100  APPLSYSD  APPLSYSX  APPLSYSD
2 Application Utilities S  APPLSYS  100  APPLSYSD  APPLSYSX  APPLSYSD
3 Applications DBA      S  APPLSYS  100  APPLSYSD  APPLSYSX  APPLSYSD
4 Oracle Alert         S  ALR      100  ALRD      ALRX      ALRD
5 Global Accounting Engi AX  100  AXD      AXX      AXD
6 Oracle Common Modules- AK  100  AKD      AKX      AKD
7 Oracle General Ledger GL  100  GLD      GLX      GLD
8 Application Report Gen RG  100  RGD      RGX      RGD

There are 48 Oracle Applications. Enter U/D to scroll up/down.

<Product #><Letter> - To change a database parameter for a product;
                     INCLUDE the LETTER ABOVE the COLUMN you want to change
U / D / T / B      - Press up/down/top/bottom to see other products
[Return]           - To return to the AutoInstall Main Menu

Enter your choice (for example, 1M) : █

```

The default tablespace names follow a standard naming convention using the product's abbreviation followed by a D for data or an X for indexes.

When you are installing, use this screen to:

- Change the default ORACLE user ID for each product with the O option. (This is not recommended.)
- Change the sizing factor for each product with the S option.
- Specify the tablespaces for each product with the M, I, and D options.

When you are upgrading, use this screen to:

- Verify that tablespaces are set correctly.
- Change the default ORACLE user ID for a new product with the O option. (This is not recommended, unless it is essential to do so.)
- Set the sizing factor for new objects for a product.
- Specify the tablespaces for each new product you are installing with the M, I, and D options.

Additional Information: [Product Table and Index Tablespaces](#) in Chapter 1

Changing Parameters

To change a parameter, type a product number and the parameter letter, as shown on the Database Parameters screen (O, S, M, I, or D). You can change a database parameter for all products by entering A (instead of a product number) along with a parameter letter.

Action Column The Action column indicates the current installation action for the product:

C	convert
I	install
S	install as shared
U	upgrade
T	translate

A blank entry means that the product is not licensed. However, the database objects for all products are installed regardless of license status. This column is displayed so that you can set the sizing factor larger for licensed products (those you use at runtime), as needed. See the [Option S: Sizing Factor](#) section later in this chapter.

You can change installation actions on the Select Products screen or, for Localizations, the Select Modules screen. An I or U action does not necessarily mean that the C or T actions are not performed. Check the Select Product screen for details.

Additional Information: [Setting Installation Actions](#) in this chapter

Option O: ORACLE User ID This column lists the ORACLE username that owns each product's database objects. You can use the O option to change a product's user ID (username and password) when you install a product, but not when you upgrade a product.

Attention: We recommend you do not change ORACLE User IDs unless it is necessary to do so.

After you select the O option, enter the new ORACLE username, which can be up to 30 characters long. Then enter the new password. To change only the password, press [Return] when AutoInstall prompts for the username.

Observe these restrictions and guidelines when changing user IDs:

- You cannot change the usernames and passwords for the following products:
 - Oracle Alert
 - Oracle Application Object Library
 - Oracle Applications DBA
 - Oracle Applications Utilities

These products use the Oracle Application Object Library username determined by AutoInstall. You can change the password for Oracle Application Object Library manually after you run AutoInstall.

Additional Information: [Changing Oracle Applications Passwords](#) in Appendix A

- Oracle Human Resources, Oracle Payroll, FastFormula, and DateTrack must be installed under the same schema. If you change the schema for one of these products, AutoInstall automatically changes the schema for the remaining ones.

Note: Except for these Oracle Human Resources products, you should install each product under its own separate schema.

- When adding an additional product installation group, adopt a naming convention that identifies which group contains each product. For example, append 2 to the usernames of products in the second product installation group.

Option S: Sizing Factor This column shows the sizing factor that AutoInstall applies to new product tables and indexes. To change a sizing factor, select the product with the S option and enter the new sizing factor at the prompt.

If you are upgrading, the sizing factor affects only the new database objects added when upgrading. It does not affect objects already installed in the database.

Additional Information: Sizing Factor, *Oracle Applications Concepts*.

Note: Database objects for all Oracle Applications products are installed, even if you did not license the product. Non-licensed products are installed with the sizing factor shown on the Database Parameters Screen. By leaving the sizing factor at the default (100), you will later be able to license a new product and use it immediately. Your DBA may then set the newly-licensed product's table and index next extent based on use.

Options M, I, D: Main Tablespace, Index Tablespace, and Default Tablespace The Main Tablespace and Index Tablespace columns show the Oracle8 Server tablespaces in which AutoInstall will place new products' tables and indexes. The Default Tablespace column shows the tablespace used for operations that do not specify a tablespace, and defaults to the main tablespace for that product.

Use the M, I, or D option to change a product's main tablespace, index tablespace, or default tablespace, respectively. These tablespaces must exist before you start the installation. AutoInstall verifies tablespace names when you enter them, and when returning to the Main Menu.

If you specify new main or index tablespaces when upgrading, AutoInstall places only new database objects in those tablespaces. It does not move existing tables or indexes to the new tablespaces.

Attention: Entering AM or AI will not change the tablespaces for products that are already installed in the database. You can change the tablespaces for these products, however, by entering the product number explicitly, such as <product #>M or <product #>I.

Tasks Screen

Use the AutoInstall Tasks screen to choose the tasks that AutoInstall will perform during an installation or upgrade.

```

AutoInstall - Choose overall tasks and their parameters
-----
# Task                                     Do it?  Parameters
-----
1 Check init.ora parameters               YES
2 Unload files                           YES
3 Create Applications environment file     YES      apptest.env
4 Relink Applications programs             YES
5 Verify files necessary for install/upgrade YES
6 Check SYS_DUAL table                   YES
7 Install or upgrade database objects     YES
8 Load US data or other language data    YES
9 Compile flexfield data in AOL tables    YES

There are 14 tasks. Enter U/D to scroll up/down.

<Task #>      - To change YES to NO or NO to YES
                (You cannot change a task marked with a *)
<Task #>P    - To change the parameters of a task
U / D        - To page up/down to see other tasks
[Return]     - To return to the AutoInstall Main Menu

Enter your choice (for example 2 or 2P) :

```

For both installations and upgrades, the tasks default to values based on your configuration:

- Verify that you want to perform the tasks already set to YES.
- Change optional tasks from NO to YES as needed.
- Modify any applicable Parameters.

Attention:

- There are multiple pages to this screen. Be sure to review the setting of every task.
 - The AD Administration utility (adadmin) lets you run many of the same tasks to maintain products that are installed after you have finished installing or upgrading the database using Auto-Install.
-
-

Additional Information: [AD Administration Utility \(adadmin\)](#) in Chapter 5

Setting Tasks

The following table lists AutoInstall tasks and the associated default settings. Type the task number to toggle the task's setting between YES and NO. To change a parameter, type the task number, the letter P, and [Return]. Then enter a new parameter value. Currently, you can only modify the parameters when creating an environment file.

Attention: Later sections explain which tasks you should set based on the types of files being installed in the current APPL_TOP.

Task 1: Check init.ora parameters When set to YES, AutoInstall checks the database initialization (init.ora) parameters when you start your installation or upgrade. It notifies you if any parameter values fall below the minimum values for numeric parameters, or do not match the values of non-numeric parameters listed in Table 1-4, Initialization Parameter Values.

Additional Information: [Initialization Parameters](#) in Chapter 1, and [Running the Installation or Upgrade](#) in this chapter

Task 2: Unload files Unloads product files from CD when you install or upgrade a product group.

Additional Information: [Installing Product Files \(File Tasks\)](#) in this chapter

Attention: Set this task to NO if you have already unloaded the files you need.

Task 3: Create Applications environment file Creates an environment file that defines Oracle Applications environment variables. It accepts the filename as a parameter. The environment file can have a name of up to 30 characters long. It should have the .env extension.

The default filename is <dbname>.env, where <dbname> is the name of the database in which the current product group is installed.

Task 4: Relink Applications programs Links Oracle Applications programs with the Oracle8 Server, networking, and Oracle tools libraries. These programs must be linked to function in your configuration. This task defaults to No if the action is a language translation only.

Task 5: Verify files necessary for install/upgrade Verifies that all files necessary to install or upgrade Oracle Applications are present.

Additional Information: [Checklist for This Chapter](#) in Chapter 4

Attention: AutoInstall will stop at this point if there are any missing files.

Task 6: Check SYS.DUAL table Checks for the existence of and permissions to access the SYS.DUAL table, which is necessary to run Oracle Applications.

Task 7: Install or upgrade database objects Installs or upgrades the product group's database objects.

Task 8: Load US data or other language data Loads language seed data into product tables.

WARNING: No matter which language you are using, leave this task set to YES if you are installing or upgrading database objects. Otherwise you will not be able to use Oracle Applications in any language.

Task 9: Compile flexfield data in AOL tables Compiles flexfield data structures in Oracle Application Object Library tables. If you choose not to compile the structures through AutoInstall, each is compiled the first time a user accesses the flexfield.

Task 10: Generate message files Generates message binary files (extension .msb) from Oracle Application Object Library tables. Oracle Applications uses the message binary files to display messages for Oracle Forms 4.5 users and for some concurrent requests.

Task 11: Generate form files Generates Oracle Forms 4.5 binary form files from the form definition files included with the product files. Oracle Applications uses the binary form files to display data entry forms. By default, this task is always set to NO. Set this task to YES only if you are installing translations or localizations, or if the installation notes for your platform indicate to set this.

Task 12: Generate report files If set to YES, AutoInstall regenerates the binary report files (extension .rdf) for your platform. Oracle Applications ships .rdf files in the standard distribution. Set this task to YES only if you are installing translations or localizations, or if the installation notes for your platform indicate to set this.

Task 13: Verify files necessary for runtime Verifies that all product files necessary to run Oracle Applications are present. AutoInstall lists any missing files, but does not stop.

Task 14: Compile APPS schema(s) Attempts to compile all invalid objects in the APPS schema(s). This task runs in parallel mode. AutoInstall lists any invalid objects in the AutoInstall worker log file (adworkxx.log). It stops if this is a new install and there are fatal problems. It does not stop if this is an upgrade and there are invalid objects, because this is an intermediate point in the entire upgrade process.

Note: If you are upgrading, you may see some invalid packages at this point in the upgrade. This is normal, and will usually be resolved after performing the post-install and post-upgrade steps.

The defaults for tasks vary depending on the types of files you are installing in the current APPL_TOP. Table 3–1 lists default values in the following columns:

- A. Files for installing and upgrading the database (administration server)
- W. Java and HTML files (web server)
- F. Applications form files (forms server)
- C. Concurrent program files (concurrent processing server)

Note: Default tasks settings are cumulative. For example, if you choose to install Applications form files and concurrent program files in the same APPL_TOP, the tasks in column *C* and column *F* will be set (✓).

Table 3–1 Task Defaults

Task	A	W	F	C
1. Check init.ora parameters	✓	✗	✗	✗
2. Unload files	✓	✓	✓	✓
3. Create Applications environment file	✓	✓	✓	✓
4. Relink Applications programs	✓	✓	✓	✓
5. Verify files necessary for install/upgrade	✓	✗	✗	✗
6. Check SYS.DUAL table	✓	✗	✗	✗
7. Install or upgrade database objects	✓	✗	✗	✗
8. Load US data or other language data	✓	✗	✗	✗
9. Compile flexfield data in AOL tables	✓	✗	✗	✗
10. Generate message files	✓	✓	✓	✓
11. Generate form files	✗	✗	✗	✗
12. Generate report files	✗	✗	✗	✓
13. Verify files necessary for runtime	✗	✓	✓	✓
14. Compile APPS schema(s)	✓	✗	✗	✗

Installing Product Files (File Tasks)

The tasks you can perform in AutoInstall are divided into three categories: file tasks, database tasks, and generation tasks. You may perform all tasks at once, or you can separate them into multiple AutoInstall sessions, which you will need to do if you are running in a multi-tier environment.

To perform all tasks at once, type *all* at the prompt on the Choose overall tasks screen. This will set the action to YES for all tasks that are relevant to the types of files you are installing in your current APPL_TOP.

You may want to only install Oracle Applications product files, then run AutoInstall later to install or upgrade the database objects and create files necessary for runtime. For example:

- You may need to review or edit Oracle Applications product files before you install or upgrade database objects.
- You may want to minimize the time the database is unusable by first performing all tasks not related to the database.
- You may want to install product files on all servers in a multi-tier environment.

If you choose to run separate AutoInstall sessions for each of the various sets of tasks, you *must* perform the file-related tasks first, then the database-related tasks, then finally the generation tasks.

To install only the product files, select the products through the Select Products and Select Modules screens, specify any different unload directories using the Product Details screen, and skip the Database Parameters Screen. Then type *files* at the Task screen prompt. This sets the following tasks to YES:

- ☐ Unload files
- ☐ Create Applications environment file
- ☐ Relink Applications programs
- ☐ Verify files necessary for install/upgrade

After choosing the option to run the selected tasks, exit AutoInstall and choose to save the restart files. You can then re-run AutoInstall later to perform the database tasks or runtime tasks, depending on the types of files you have in the current APPL_TOP.

Installing or Upgrading Database Objects (Database Tasks)

If you are installing on an administration server, you must perform file tasks and database tasks. If you chose to separate these tasks by following the steps in the previous section, you must now perform the database-related steps.

Restart AutoInstall, and answer "Yes" when it asks if you want to continue your previous session. When you get to the Main Menu, make sure the options are set properly in the Database Parameters Screen, then proceed to the Tasks screen and type *database* at the prompt. This will set the following tasks to YES:

- ☐ Check init.ora parameters
- ☐ Check SYS.DUAL table

- ☐ Install or upgrade database objects
- ☐ Load US data or other language data
- ☐ Compile flexfield data in AOL tables
- ☐ Compile APPS schema(s)

When you install or upgrade multiple product groups that use the same set of product files, you should perform all necessary tasks when installing or upgrading the first product group. With subsequent product groups, however, you generally need to run only the database-specific tasks, which are listed above.

Once you have installed or upgraded database objects, you can perform the generation tasks listed in the next section.

Installing Files Needed for Runtime (Generation Tasks)

If you are installing on a forms server or concurrent processing server, you will need to perform the generation tasks to generate forms, reports, and message files from data in the database. These tasks can only be run *after* you have performed the file-related and database-related tasks listed in the previous sections.

Restart AutoInstall, and answer "Yes" when it asks if you want to continue your previous session. When you get to the Main Menu, make sure the options are set properly in the Database Parameters Screen, then proceed to the Tasks screen and type *runtime* at the prompt. This will set one or more of the following tasks to YES, depending on the types of files installed in your current APPL_TOP:

- ☐ Generate message files
- ☐ Generate form files

This task is set to YES only if you are installing on a forms server. In general, you do not need to generate forms unless you are installing translations or localizations.

- ☐ Generate report files

This task is set to YES only if you are installing on a concurrent processing server. In general, you do not need to generate reports unless you are installing translations or localizations.

- ☐ Verify files necessary for runtime

These are the last tasks you must perform using AutoInstall. Once you have completed these tasks, you may exit AutoInstall without saving your restart files.

Attention: Once you have completed your install or upgrade, you must use the adadmin utility to perform any maintenance tasks.

Additional Information: [Maintaining Applications Files](#) in Chapter 5

Running the Installation or Upgrade

After you have configured the installation or upgrade through the AutoInstall screens, you run the complete set of installation or upgrade steps from the Main Menu. This section explains what information AutoInstall prompts for before it runs the tasks you chose.

The tasks you set on the Tasks screen determine which of the following questions AutoInstall asks. After you answer all the questions that are required for the tasks you selected, AutoInstall automatically starts the installation or upgrade.

Environment File

If you chose to create an environment file, AutoInstall asks a series of questions about parallel concurrent processing, file naming conventions, directory paths for log, output, and temporary files, the mail product you are using, and so on. The Creating an Environment File section explains these AutoInstall questions.

Additional Information: [Creating an Environment File](#) in this chapter

Number of Workers

If you are installing, upgrading, translating, converting, or compiling database objects, AutoInstall will launch multiple worker processes to perform these steps in parallel. The default value for the number of workers is automatically determined by AutoInstall to be two plus the number of CPUs on the machine where AutoInstall is running. For example, on single-processor machines, the default is 3.

AutoInstall prompts you here for the number of workers that you want to use. To run in parallel, choose one or two workers more than the number of CPUs on your database server machine, up to 32 workers. It is inefficient to have more than this because the CPU may waste time switching between the processes.

Initialization Parameters

If you chose to verify initialization (init.ora) parameters, AutoInstall verifies the parameters and displays a warning message if a parameter has an incorrect value:

Your parameter value enqueue_resources is 200 and should be >= 5000

Your parameters in init.ora file are not set up correctly.
Do you wish to continue [No] ?

If a parameter value is incorrect, press [Return] to quit AutoInstall. Change the initialization parameter to the recommended value and shut down and restart the database. You can then restart AutoInstall and continue your session.

WARNING: AutoInstall may fail during the installation or upgrade if the initialization parameters are not set correctly.

Additional Information: [Initialization Parameters](#) in Chapter 1

Unloading Files

If you are installing or upgrading a regular product group and the task to unload files is set to YES, AutoInstall asks you to confirm that you want to unload the product files. You can answer No at the prompt if you do not need to unload the product files.

Media Device

If you are unloading files, insert the appropriate Oracle Applications CD into the CD-ROM drive.

AutoInstall prompts for the name of the machine that contains the media drive. Press [Return] if you are running AutoInstall from the machine that contains the drive. AutoInstall then prompts for the drive name and asks you to confirm the unload.

AutoInstall verifies that the drive operates properly and that you have inserted the correct CD. It then begins to unload the product files. If the files are on more than one CD, AutoInstall instructs you to mount each CD separately and to specify the drive name before it unloads each CD.

If AutoInstall encounters an error while unloading, it asks if you wish to continue as if the unloading was successful. We recommend that you press [Return] to select the default answer No. Review the `adunload.log` file for errors. The section on AutoInstall Log Files later in this book provides more information on how to check log files for errors. Contact Oracle Support Services if you do not understand the meaning of an error message.

Verifying Files

If you selected the tasks to Verify Files Needed For Install/Upgrade or Verify Files Needed For Runtime, AutoInstall asks for the name of the log file in which the output from these tasks should be stored:

Please enter the filename you wish to use or press [RETURN] to accept the default filename [`adiuvf.lst`] :

These log files are stored in the same directory as your other AutoInstall log files. The default file names are `adiuvf.lst` (files needed for installation) and `advrunvf.lst` (files needed for runtime).

If any files needed for install or upgrade are missing, AutoInstall displays a failure message so you can correct the problem before continuing.

When verifying files needed for runtime, AutoInstall does not fail. You should therefore review the log file from this task after the installation is complete to ensure there are no missing files.

Monitoring AutoInstall

Monitor AutoInstall as it runs to check for problems with the installation or upgrade. The Monitoring AutoInstall section later in the chapter explains how to do this in further detail.

Exiting AutoInstall

The installation or upgrade of the Oracle Applications product files and database objects is complete when the Main Menu appears on the screen. You can then exit AutoInstall.

Attention: You should always exit AutoInstall after running the selected tasks. If you wish to perform more tasks in the current environment, choose the option to Exit AutoInstall, answer No when asked if you wish to delete your AutoInstall restart files, then restart AutoInstall and select the new tasks.

Creating an Environment File

An environment file sets environment variables necessary to use an Oracle Applications product group. This section explains the questions AutoInstall and the adadmin maintenance utility ask if you choose to create an environment file.

Environment File Name

If you are running adadmin, you are prompted for the filename adadmin should use when creating the environment file. The default is <dbname>.env, where <dbname> is the name of the database in which the current product group is installed. AutoInstall does not ask this question, as the filename is provided as a parameter on the Tasks screen.

Parallel Concurrent Processing

AutoInstall and adadmin ask whether you want to enable parallel concurrent processing. The option you select determines whether the utilities set the environment variable APPLDCP to OFF, ON, or OSQ, respectively:

1. Not enabled
2. Enable generic parallel concurrent processing
3. Enable parallel concurrent processing with operating system queue

Choose option 1 (the default) if you do not plan to distribute concurrent processing among multiple concurrent processing servers. Choose option 2 to enable parallel concurrent processing without integration with any load-balancing or queue management features that may be built into your platform's operating system.

If your platform has features such as load-balancing or queue management, and these features are supported by Oracle Applications, choose option 3 to integrate parallel concurrent processing with them. Refer to your installation update for information about whether your platform operating system includes such features supported by Oracle Applications.

Additional Information: Parallel Concurrent Processing, *Oracle Applications Installation Update* for your platform

DOS-Compatible Filenames

The utilities ask if you want to use the 8.3 filename convention. Answering YES sets the APPCPNAM environment variable to the value REQID, which tells the concurrent manager to use filenames that meet DOS naming requirements.

Directories for Log and Output Files

The utilities prompt for information about log and output file directories. You can first enter the full pathname of a common directory for all log and output files created through the product group. The utilities set the variable APPLCSF to this directory in the environment file.

Press [Return] at this prompt if you plan to use the default storage method that puts log and output files in subdirectories under each product's top directory. In this case, the utilities do not define APPLCSF in the environment file.

Additional Information: [Create common directory for log and output files \(optional\)](#) in Chapter 2

You next specify the subdirectories that hold the log and output files, respectively. The utilities set the environment variables APPLLOG and APPOUT to these directories in the group's environment file. Use the default values "log" and "out" if one of these conditions applies:

- This is the only product group you are installing or upgrading.
- This is one of multiple product groups and each group has its own common directory for log and output files.

When you have multiple product groups that use the default storage method or a single common directory for all groups, each group needs its own log and output subdirectories to keep its files separate from those of other groups. In this case, enter unique subdirectory names at the prompts. For example, you could enter logtst and outtst as the directory names for a test product group.

Additional Information: *Log and Output Files, Oracle Applications Concepts*

Directories for Temporary Files

The utilities prompt for the pathname of the directory for Oracle Applications temporary files. It sets the variable APPLTMP to this directory in the environment file. A second prompt asks for the directory for Oracle Reports temporary files. The utilities set the REPORTS25_TMP variable to this directory.

Specify the same directory at both prompts if you have set up only one temporary directory. If you have not set up any temporary directories, press [Return] at the prompts to use the default values.

Additional Information: *Temporary Files, Oracle Applications Concepts*

The utilities now require a location for temporary PL/SQL output files. The directory you choose here must be listed in the `utl_file_dir` parameter in your `init.ora` file. Before prompting for a location, the utilities show you the current value of this parameter, as it appears in the `V$PARAMETER` table in your database. Choose one of the directories listed for the location of temporary PL/SQL output files.

Additional Information: [Initialization Parameters](#) in Chapter 1

If the parameter is not set, or you wish to store these files in a directory other than those listed, exit `AutoInstall` or `adadmin`, set or modify the `utl_file_dir` parameter in your `init.ora` file, then shut down and restart the database to read the new parameter setting. You may then restart `AutoInstall` or `adadmin` to continue from this point.

Note: `AutoInstall` and `adadmin` will fail at this point if the `utl_file_dir` parameter is not set at all.

Mail Product Used for Oracle Alert Linking

If you chose to generate an environment file, the utilities ask you to specify the electronic mail product you use:

With which Oracle mail product should Oracle Applications be linked:

1. Link with Oracle InterOffice
2. Do not link with Oracle InterOffice

Your choice determines how the utilities set the environment variable `APPLMAIL`, which is used when linking Oracle Alert executables with Oracle8 Server libraries. Option 2 is the default. Choose it if you do not use Oracle InterOffice or do not plan to use Oracle Alert.

Web Server

Context-sensitive help for Oracle Applications is in HTML format, and is accessed from the desktop client through a connection to a web server. `AutoInstall` and `adadmin` need to know the name of the machine on which this web server is running. The utilities prompt you for the following information:

- Name of the machine (including domain name) that will host the web server used for accessing Applications forms. For example, websrvr1.mycompany.com.
- The port that the Applications forms web server is running on. The default is 80.

This information is then recorded in the FORMS45_MAPPING variable in your environment file.

Note: The utilities also create a security file in JAVA_TOP for web use. If JAVA_TOP does not exist or does not have write privileges for the applmgr account, AutoInstall generates an error at this point. You must supply the JAVA_TOP information in the \$APPL_TOP/admin/adovars.env file before you can continue.

Additional Information: [Edit adovars.env file](#) in Chapter 2

Monitoring AutoInstall

This section describes the AutoInstall status messages you see on the screen, tells you where to find AutoInstall log and restart files, and explains how to correct AutoInstall errors.

Manager and Worker Messages

When running in parallel, AutoInstall acts as a *manager* that coordinates a number of *workers*, assigning them jobs that they run to install or upgrade, translate, convert, or compile objects. You see messages like the following on the screen:

```
Assigned: file afmisc.drv   on worker 1   for product fnd username APPLSYS.
Assigned: file afpt.drv     on worker 2   for product fnd username APPLSYS.
Assigned: file alaf.drv     on worker 3   for product alr  username APPLSYS.
Completed: file afmisc.drv  on worker 1   for product fnd username APPLSYS.
Assigned: file afform.drv   on worker 1   for product fnd username APPLSYS.
Completed: file afpt.drv    on worker 2   for product fnd username APPLSYS.
```

These messages indicate what each worker is doing. The example shows three workers running upgrade scripts on two products, which are identified by their short names (fnd and alr).

Worker Failed Message

The manager displays a message like the following if a worker cannot complete its job:

```
FAILED: file afform.drvc on worker 1 for product fnd username APPLSYS.
```

This example shows that worker 1 failed on the file afform.drvc for Oracle Application Object Library (fnd). The worker, the failed job, and all jobs that depend on the failed job are idle after a worker fails. The manager continues to assign jobs not dependent on the failed job to the other workers, however.

Your AutoInstall session is not complete until all jobs run successfully. When a worker fails, determine the cause of failure, fix the problem, and restart the worker. The following sections explain how to do this.

Log and Restart Files

The manager and workers maintain separate log files that fully record the actions they perform, as well as separate restart files that they refer to when restarting jobs. The log files record AutoInstall actions far more extensively than the messages you see on the screen. You will need to review the log files if AutoInstall encounters errors.

Attention: See also the section on correcting AutoInstall errors for important information about reviewing a log file.

Additional Information: [Correcting AutoInstall Errors](#) in this chapter

Manager Log and Restart Files

The manager has a log file that you name at the beginning of the AutoInstall session. The default name is adaimgr.log. The file resides in the directory \$APPL_TOP/admin/<dbname>/log, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable. The manager uses the following restart files, which reside in \$APPL_TOP/admin/<dbname>/restart:

- airfinit.rf9, the restart file that holds your responses to the initial AutoInstall questions
- airfprod.rf9, the product restart file that records each product's installation action, user ID, tablespaces, and shared dependencies

- `airfmod.rf9`, the module restart file that records localization modules to be installed or upgraded
- `airftask.rf9`, the task restart file that records the installation or upgrade tasks you chose to perform
- `airfmain.rf9`, the main restart file that records which installation or upgrade steps have been completed
- `airftodo.rf9`, the "to do" file that records which installation or upgrade steps are left

AutoInstall refers to these restart files when you restart a session that did not run to completion. Note that when AutoInstall continues past a step, it will not go back and try it again.

Attention: If AutoInstall fails and you choose to continue past the error, you must correct the problem manually. If you do not correct the problem, AutoInstall may not be able to continue.

By default, AutoInstall deletes these files (but leaves backup versions with the extensions `.bak`, `.bk2`, or `.bk3`) when the installation or upgrade finishes. AutoInstall does not delete these files if you answer No to the following prompt when exiting from the Main Menu:

```
Do you wish to delete your AutoInstall restart files? [Yes]
```

Worker Log and Restart Files

Worker log files reside in the directory `$APPL_TOP/admin/<dbname>/log`, where `<dbname>` is the value of your `ORACLE_SID` or `TWO_TASK` variable, and are named `adworkxx.log`, where `xx` indicates the worker number. For example, worker 1 uses the log file `adwork01.log`.

Each worker also has a restart file called `adworkxx.rf9`. These files are stored in `$APPL_TOP/admin/<dbname>/restart`. The worker creates the restart file when the manager assigns it a job. It deletes the restart file when it finishes the job.

WARNING: Do not modify or delete any worker restart files.

Correcting AutoInstall Errors

When a worker fails its job, you do not have to wait until the other workers and the manager stop. You can fix the problem and restart the worker while the manager is running by performing the following steps.

1. Log in as applmgr and verify environment.

Log in as applmgr from another terminal or terminal window. Ensure that your ORACLE environment variables are set to the appropriate database and home directory. Also ensure that \$APPL_TOP is set to the top Applications directory.

Additional Information: [Verify \\$APPL_TOP](#) in Chapter 2

2. Run the environment file.

AutoInstall creates the product group's environment file at the beginning of the installation or upgrade. Load this file into your current environment with the following command:

```
$ . $APPL_TOP/<FILENAME>.env
```

Here is an example:

```
$ . $APPL_TOP/APPLSYS.env
```

If this environment file is not available, load adsetenv instead:

```
$ cd $APPL_TOP/admin
$ . adsetenv
```

3. Split or copy the worker log file.

Verify that the worker log file is not too large for your system's file editor:

```
$ cd $APPL_TOP/admin/apptest/log
$ ls -l adwork*.log
-rw-rw-r-- 1 applmgr appl 9151702 Jan  6 17:11 adwork01.log
-rw-rw-r-- 1 applmgr appl 4387663 Jan  6 17:07 adwork02.log
-rw-rw-r-- 1 applmgr appl 3017393 Jan  6 17:11 adwork03.log
```

The sample response shown above indicates that the log file for worker 2 is 4.3 Megabytes. You can use the tail command to extract the end of a large log file:

```
$ tail -100 adworkxx.log > <filename>.log
```

Here is an example:

```
$ tail -100 adwork02.log > short02.log
```

The tail command shown writes the last 100 lines of the worker log file to the specified filename.

Suggestion: You can also use the split command to split a large file into several smaller files. See your online man pages for information about the split command.

If the worker log file is small, copy the entire file to a different filename. Here is an example:

```
$ cp adwork02.log edit02.log
```

Review the copied log file, which is edit02.log in the example above. This prevents errors if the worker tries to write to its log file while you are reviewing the file.

4. Review the end of the log file to find the problem.

Review the end of the log file to find out why the worker failed. Here is an example of a worker failure message:

AD Worker error:

The following ORACLE error:

```
ORA-01630: max # extents (50) reached in temp segment in tablespace TSTEMP  
  
occurred while executing the SQL statement:
```

```
CREATE INDEX AP.AP_INVOICES_N11 ON AP.AP_INVOICES_ALL (PROJECT_ID, TASK_ID)  
NOLOGGING STORAGE (INITIAL 4K NEXT 512K MINEXTENTS 1 MAXEXTENTS 50  
PCTINCREASE 0 FREELISTS 4) PCTFREE 10 MAXTRANS 255 TABLESPACE APX
```

AD Worker error:

Unable to compare or correct tables or indexes or keys because of the error above

In the example, the worker could not create the index AP_INVOICES_N11 because it reached the maximum number of extents in the temporary tablespace.

5. Fix the problem.

Fix the problem if you know how. In the example, you would have to increase the size of the next extent or the maximum number of extents in the temporary tablespace. Contact Oracle Support Services if you do not know how to fix the problem.

6. Run `adctrl` to restart the failed job.

Run `adctrl` to review the worker status and restart the job.

Additional Information: [AD Controller Utility \(adctrl\)](#) in Chapter 5

Acceptable Errors

If a failure occurs while AutoInstall is running in parallel mode, the AutoInstall worker will fail and record an error message in the worker log file. AutoInstall itself may fail upon encountering an error if it is not running in parallel mode at the time.

If you encounter a failure that is caused by any of the reasons described in this section, or if you suspect that the problem may be a concurrency issue, restart the failed worker with the AD Controller Utility (`adctrl`). Contact Oracle Support Services if the worker encounters the same error while running the job again.

Additional Information: [AD Controller Utility \(adctrl\)](#) in Chapter 5

ORACLE Error Messages

The following ORACLE error messages indicate acceptable problems that are usually from steps that are intended to make scripts as rerunnable as possible:

```
ORA-00942: table or view does not exist
ORA-00955: name is already used by an existing object
ORA-01418: specified index does not exist
ORA-01430: column being added already exists in table
ORA-01434: private synonym to be dropped does not exist
ORA-01442: column to be modified to NOT NULL is already NOT NULL
ORA-01451: column to be modified to NULL cannot be modified to NULL
ORA-04043: object <object name> does not exist
ORA-04080: trigger '<trigger name>' does not exist
```

The following ORACLE error messages may also indicate acceptable problems:

```
ORA-00054: resource busy and acquire with NOWAIT specified
ORA-00060: deadlock detected while waiting for resource
ORA-00604: error occurred at recursive SQL level #
ORA-01555: snapshot too old: rollback segment number # with name name too small
```

Additional Information: Oracle8 Error Messages

If you are receiving ORA-1555 errors consistently, the problem may be due to insufficient rollback space. Try increasing the size of your rollback segments or additional addition rollback segments before restarting the workers.

Additional Information: [Rollback Segments](#) in Chapter 1

Relinking Error Messages

You may see various warnings while relinking like the following:

```
Warning: not setting LXXXXLIB
```

These warnings indicate that one or more product libraries are not being included because the product (typically denoted by XXX) is not installed. These warnings may be safely ignored.

AutoInstall displays a message like the following if it cannot relink a product with the Oracle8 Server:

```
An error occurred while relinking application programs.  
Continue as if it were successful [No] :
```

You must ensure all relinking problems are resolved before performing the database-related or generation-related tasks in AutoInstall, or before attempting to run Oracle Applications. We recommend that you choose the default answer to exit AutoInstall and examine the cause of the error. Review the file `adrelink.log` to determine the cause of the error. The file is located in `$APPL_TOP/admin/<dbname>/log`, where `<dbname>` is the value of your `ORACLE_SID` or `TWO_TASK` variable. When you have resolved the error, restart AutoInstall and continue from where the previous session ended.

If you are only performing file-related tasks in the current AutoInstall session, you can answer Yes to continue with AutoInstall and fix the problem when AutoInstall completes. At the end of the AutoInstall session, correct the errors that caused the relink to fail, then use the `adrelink` program to manually relink the executables which AutoInstall failed to build properly. If you have a large number of executables that failed to relink, you can run AutoInstall and choose the Relink Applications programs task again.

Additional Information: [Relink Utility \(adrelink\)](#) in Chapter 5

Database Object Differences

Several different types of warnings may be reported when AutoInstall compares database objects in an upgrade from a previous release of Applications. These warnings may include the following:

Warning: The missing column is NOT NULL

Warning: The existing index is UNIQUE and the new index is nonunique.

Warning: NO default value can be applied.

AutoInstall may also indicate that extra database objects exist, or that there are differences in the constraints of columns. Such warnings and differences are usually not a problem. If you are upgrading, there may be differences due to database customizations, or because obsolete objects or columns were not dropped by Oracle Applications during the upgrade. If you have custom database objects that rely on these Applications objects, you may wish to review these warnings after the installation to determine whether you need to modify your customizations.

AutoInstall will display messages like the following when finding missing, extra, or incorrectly installed database objects:

```
CREATE SEQUENCE GL_JE_CATEGORIES_S MINVALUE 1 MAXVALUE 2147483647 INCREMENT BY 1
START WITH 1 CACHE 20 NOCYCLE ORDER
```

These messages will also appear in the manager or worker log files, which you should review after the installation is complete. These messages usually include a SQL statement that corrects the problem. Contact Oracle Support Services before creating missing objects, as there may be additional steps such as creating grants and synonyms.

Import Error Messages

AutoInstall may run the Oracle Import utility when installing or upgrading Oracle Applications. The following IMP error messages may indicate acceptable problems:

IMP-00041: Warning: object created with compilation warnings

Warning: the objects were exported by SYSTEM, not by you

DataMerge Error Messages

A message in the following format indicates a problem with a DataMerge temporary file:

```
addmimp: <function name> : error : <error with temporary file>
```

Here, <function name> is the name of a DataMerge function such as dmmactin or dmmactwrt. The error descriptions vary, but always specify a problem with a temporary file.

This type of problem is usually intermittent, and may not occur again if you immediately restart AutoInstall or the AutoInstall worker. Persistent errors of this type may indicate a problem with the APPLTMP variable, a lack of available space in the temporary directory, insufficient rollback space, or incorrect access privileges on the directory.

Flexfield Compiling Error Message

Toward the end of your AutoInstall session, you may receive a flexfield compiler warning like this:

```
Compiling all application flexfields.
```

```
/d01/appl/110/fnd/11.0.28/bin/fdfcmp APPLSYS/FND 0 Y
Log filename      : /d01/appl/110/admin/apptest/log/l460306.req
Report filename   : /d01/appl/110/admin/apptest/out/SYSADMIN.460306
```

```
An error occurred while compiling application flexfields.
Continue as if it were successful [No] :
```

You can answer Yes to continue the AutoInstall tasks while you determine the cause of the problem. To investigate the problem, inspect the compiler log file, such as l460306.req, to determine which flexfield caused the error. The flexfield may not compile if it was improperly set up. You may need to log on to Oracle Applications after the post-install or post-upgrade steps are complete and examine the flexfields for setup errors. You can recompile flexfields through the adadmin utility after you have finished installing or upgrading the database using AutoInstall.

Additional Information: Define Descriptive Flexfield Segments, and Define Key Flexfield Segments, *Oracle Applications Flexfields Guide*

Form Generation Warnings

When generating form files, you may encounter messages similar to the following:

```
WARNING generating form forms/US/POXPOVCT.fmx from input file
      /d01/apps/110/au/11.0.28/forms/US/POXPOVCT.fmb
```

If this occurs, you need look in the worker log files to determine if this was due to an error that requires attention or if it was an ignorable warning. Certain warnings are acceptable and occur while generating many of the Applications forms. This is due to complex coding in the forms that is not common. The Forms Generator displays a warning message so that developers can verify that the complex coding was intentional. Following is the current list of warnings which may occur when generating Applications forms. These errors may be safely ignored:

```
FRM-30188: Warning! no default value given and other values are not allowed
FRM-30351: Warning! No list elements defined for list item
FRM-30370: Warning! (<block>.<field>) Relation's detail block is a control block
FRM-30371: Warning! (<block>.<field>) Relation's master block is a control block
```

Note: Form generation warnings can be ignored if you are performing an installation or upgrade. Generally, they may demand attention only when applying patches.

After Running AutoInstall

After you run AutoInstall, you can finish the product group's installation or upgrade immediately, or install or upgrade another product group.

Finishing Your Installation or Upgrade

After you run AutoInstall, you need to perform manual steps that complete the product group's installation or upgrade. If you are installing multiple product groups, you can perform these steps after you have installed all the groups.

If you are upgrading multiple product groups, you should perform the manual steps for each group immediately after upgrading the group with AutoInstall. This brings each product group back into service as soon as possible.

Additional Information: Chapter 4, Finishing Your Installation or Upgrade

Installing Another Product Group

If you plan to install or upgrade another product group, make sure that you prepare your system for that installation or upgrade before you run AutoInstall again.

Additional Information: Chapter 2, Preparing to Install or Upgrade

After Running AutoInstall

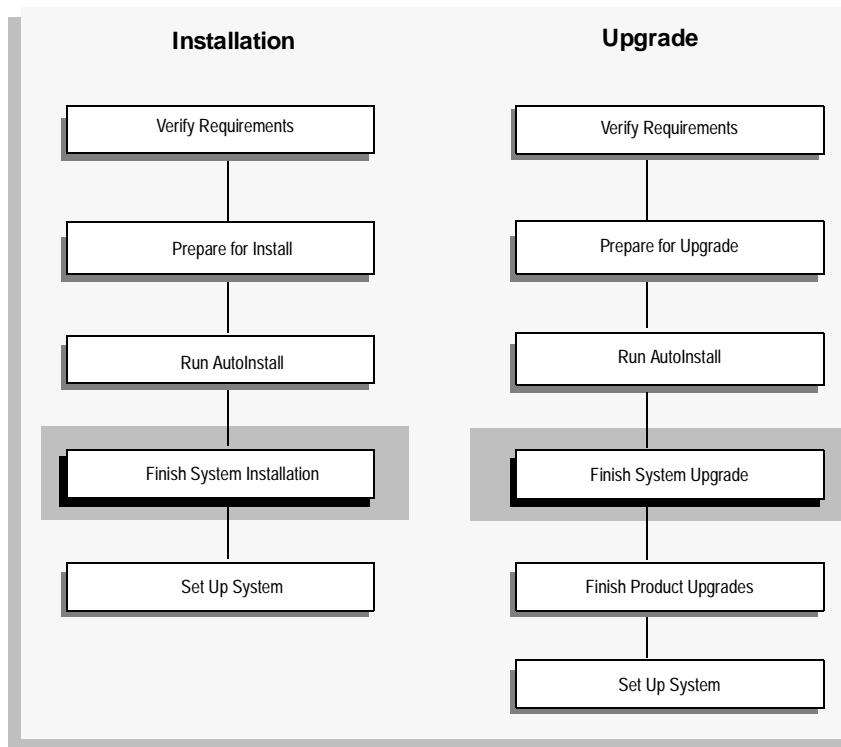
Finishing Your Installation or Upgrade

This chapter describes the steps to take to finish your installation or upgrade of Oracle Applications. Perform these steps after you run AutoInstall. This chapter explains the following tasks:

- [Your Place in the Installation or Upgrade Process](#)
- [Checklist for This Chapter](#)
- [Review AutoInstall Log Files](#)
- [Set Oracle8 Server Parameters](#)
- [Configure the Web Server](#)
- [Configure the Forms Server](#)
- [Set Up Concurrent Managers](#)
- [Install Parallel Concurrent Processing](#)
- [Set Up the Report Review Agent](#)
- [Start Oracle Applications](#)
- [Complete Post-Upgrade Steps](#)
- [Re-Enable and Integrate Customizations](#)
- [Tasks To Do Any Time After Installing or Upgrading](#)

Your Place in the Installation or Upgrade Process

The following figure illustrates your current place in the Oracle Applications installation or upgrade process. Follow the instructions in this chapter to carry out the highlighted steps.



Checklist for This Chapter

Each of the following steps is explained in this chapter.

Review AutoInstall Log Files

- ☐ 1: Search the adaimgr.log and adworkxx.log files for errors and differences
- ☐ 2: Search the adunload.log for errors
- ☐ 3: Search the adrelink.log file for errors
- ☐ 4: Verify APPS Schema(s)

Set Oracle8 Server Parameters

- ☐ 5: Reset NLS variables
- ☐ 6: Reconfigure rollback segments for normal use (optional)

Configure the Web Server

- ☐ 7: Configure virtual directories
- ☐ 8: Configure the Java Cartridge
- ☐ 9: Create an Oracle Web Application Server Database Access Descriptor (DAD)
- ☐ 10: Modify Web Application Server security level

Configure the Forms Server

- ☐ 11: Define Oracle Forms Cartridge parameters
- ☐ 12: Configure Developer/2000 Metrics Server
- ☐ 13: Modify the initial HTML signon file
- ☐ 14: Start HTTP listener and Forms Server Listener processes

Set Up Concurrent Managers

- ☐ 15: Set up electronic mail
- ☐ 16: Create log and output files in each product top directory
- ☐ 17: Define startup parameters

- ☐ 18: Customize startmgr script (optional)
- ☐ 19: Restrict access to concurrent processing servers
- ☐ 20: Start the concurrent managers

Install Parallel Concurrent Processing

- ☐ 21: Create new environment files on each concurrent processing server
- ☐ 22: Run the environment file
- ☐ 23: Edit dcstart on each concurrent processing server

Set Up the Report Review Agent

- ☐ 24: Configure network configuration files
- ☐ 25: Verify file permissions for Report Review Agent executable

Start Oracle Applications

- ☐ 26: Ensure client software is installed
- ☐ 27: Access signon screen from a web browser
- ☐ 28: Set up system administration

Complete Post-Upgrade Steps

- ☐ 29: Run final product upgrade steps
- ☐ 30: Drop obsolete operating unit schemas

Re-Enable and Integrate Customizations

- ☐ 31: Reapply product customizations to files and databases
- ☐ 32: Verify installation group numbers
- ☐ 33: Integrate custom objects and schemas
- ☐ 34: Re-enable custom triggers, constraints, and indexes

Tasks To Do Any Time After Installing or Upgrading

- ☐ 35: Change Oracle Application schema passwords (recommended)
- ☐ 36: Pin SGA packages and sequences (recommended)
- ☐ 37: Install help text
- ☐ 38: Delete obsolete product files (optional)
- ☐ 39: Back up Oracle Applications (recommended)
- ☐ 40: Apply patches

Review AutoInstall Log Files

Even if AutoInstall did not display any error messages on your terminal screen, you should review the log files for reported errors or differences. All files are in \$APPL_TOP/admin/<dbname>/log, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable. If you have multiple sets of product files, make sure that \$APPL_TOP is set correctly before you change directories. Contact Oracle Support Services if you do not understand a message.

Suggestion: Review log files online instead of printing them. Most platforms have a split command you can use to break up large files and make them easier to review. See the for UNIX online manual pages for more information.

Errors and Warnings

Search for the following text to locate errors and warnings that may need attention:

```
Error:
error:
ERROR:
Warning
warning
WARNING
ORA-
APP-
FRM-
REP-
```

Additional Information: [Acceptable Errors](#) in Chapter 3, and [Getting Help](#) in the [Preface](#)

1. Search the adaimgr.log and adworkxx.log files for errors and differences

The file adaimgr.log is the default log file for the AutoInstall manager. You may have specified a different filename when you ran AutoInstall or created more than one log file. The files adworkxx.log (such as adwork01.log) are the worker log files created when you run AutoInstall in parallel. There is one file for each worker.

Additional Information: [Log and Restart Files](#) in Chapter 3

2. Search the adunload.log for errors

This file contains information about the unloading of product files from CD. Search for general errors and warnings. If you chose the "Verify files necessary for runtime" task and it did not report any missing files, then there should not be any problems in the unload log file.

3. Search the adrelink.log file for errors

This file contains information about relinking Oracle Applications programs with the Oracle8 Server and tools. Search for general errors and warnings. If you chose the "Verify files necessary for runtime" task and it did not report any missing files, then there should not be any problems in the unload log file.

4. Verify APPS Schema(s)

If you are upgrading from Release 10.6 or Release 10.7 of Oracle Applications, we recommend that you run the AD Administration Utility to verify the integrity of your APPS Schema(s). The "Verify APPS Schema(s)" option on the Maintain Database Objects menu will run a SQL script that finds database objects that have problems or may potentially cause problems with your Applications Installation. After running this task, review the information in the log file produced to see if any corrective action is needed.

Additional Information: [AD Administration Utility \(adadmin\)](#) in Chapter 5.

Set Oracle8 Server Parameters

Omit this step if you plan to run Oracle Applications in American English.

5. Reset NLS variables

The language and format of the data returned to the desktop client is determined by the settings of the NLS variables when the Forms Server Listener and HTTP listener are started. NLS variables may be set in the environment, or in the database initialization parameters (init.ora) file. Settings in the environment take precedence over those in the initialization parameters file.

Attention: We recommend leaving the initialization parameter settings as you configured them in Chapter 1, and use the NLS environment variables to override these settings as required. This allows you to switch among different languages without having to shutdown and restart the database each time.

The NLS environment variables are defined in the main Oracle Applications environment file. You set these variables by running this environment file. The environment file includes default settings for the variables NLS_LANG, NLS_DATE_FORMAT, NLS_NUMERIC_CHARACTERS, and NLS_SORT.

Check that the value of the NLS_LANG variable in the environment file is set to the required runtime language and territory settings. The language and territory part of NLS_LANG should be set to the language and territory in which the you plan to run Oracle Applications. The character set portion of the NLS_LANG variable must remain set to the character set value used when installing the Oracle Applications software.

Additional Information: [Set NLS environment variables in Chapter 2](#)

For example, if you plan to run Oracle Applications using French as the language, France as the territory, and using a WE8DEC terminal, you would set NLS_LANG as follows:

```
$ NLS_LANG="FRENCH_FRANCE.WE8DEC"; export NLS_LANG
```

The value specified for NLS_DATE_FORMAT in the environment file must not be changed. Oracle Applications does not support running the Applications software with any other setting of the NLS_DATE_FORMAT variable.

The NLS_NUMERIC_CHARACTERS variable is set in the environment file to use the default numeric character format value of ".,". You must keep the NLS_NUMERIC_CHARACTERS variable set to this value in the environment file. If you wish to use a local numeric character format for your forms server and web

server operations, the system administrator can temporarily reset the NLS_NUMERIC_CHARACTERS variable to the local numeric character format before starting the forms server listeners and web listeners. Remember to reset the variable back to the "., " setting afterwards.

Attention:

- The use of local numeric character formats is only supported for the running of the Forms Server Listener and HTTP listener. All other operations must use the required "., " numeric format.
 - It is important to note that all Forms Server Listeners and HTTP listeners connected to a given Oracle Applications database must use the same numeric character format value. If you are using an alternative format, ensure that Forms Server Listeners and HTTP listeners are started with the same setting of the NLS_NUMERIC_CHARACTERS environment variable.
-
-

The NLS_SORT variable determines the linguistic sort sequence used when running Oracle Applications. It is set to BINARY in the environment file. You can change the NLS_SORT to an alternative sort method, however we recommend that you use BINARY sort to avoid the potential performance degradation associated with using some of the linguistic sort sequences. Update the NLS_SORT setting in the environment file if you require an alternative sort method.

To avoid the potential performance degradation associated with using some of the linguistic sort sequences, we recommend using binary sort regardless of the language specified. The sorting method is determined from the NLS_SORT variable. To override the NLS_SORT default settings, NLS_SORT should be set for each user:

```
$ NLS_SORT=BINARY; export NLS_SORT
```

The NLS_DATE_LANGUAGE determines the language of the month section of a date. For example, if NLS_DATE_LANGUAGE is set to French, a date will appear as 21-FEV-97, or if set to Polish, a date will appear as 21-LUT-97. The NLS_DATE_LANGUAGE variable is not set by the environment file. This variable defaults to the value of the language portion of NLS_LANG. Set NLS_DATE_LANGUAGE if your user requires an alternate value.

6. Reconfigure rollback segments for normal use (optional)

Your database administrator may want to reconfigure the Oracle8 Server rollback segments and temporary tablespaces for normal use. See Table 1–5 and Table 1–6 for size recommendations.

Configure the Web Server

7. Configure virtual directories

Map the following virtual paths on your web server to the physical locations of your Oracle Applications files. Consult your web server documentation for information on how to change these settings.

File System Directory	Flag	Virtual Path
<location of your Applications temporary files>/	NR	/OA_TEMP/
<location of your HTML files>/	NR	/OA_HTML/
<location of your HTML files>/bin/	CN	/OA_HTML/bin/
<location of your Java files>/	NR	/OA_JAVA/
<location of your on-line help files>/	NR	/OA_DOC/
<location of your media files>/	NR	/OA_MEDIA/

For example, if your Oracle Applications temporary files are located in /tmp, your HTML files are at /u01/public/html, your online documentation files are installed at /u01/appsdoc, other media files are at /u01/public/media, and your Oracle home is /u01/oracle, you would enter the following:

File System Directory	Flag	Virtual Path
/tmp/	NR	/OA_TEMP/
/u01/public/html/	NR	/OA_HTML/
/u01/public/html/bin/	CN	/OA_HTML/bin/
/u01/oracle/forms45/java/	NR	/OA_JAVA/
/u01/appsdoc/	NR	/OA_DOC/
/u01/oracle/forms45/java/oracle/apps/media/	NR	/OA_MEDIA/

OA_JAVA, OA_DOC, and OA_MEDIA should be set to the same paths as the environment variables of the same names respectively. If you are not certain about what file system directory to enter for these virtual paths, check the values of these environment variables set by adovars.env. For example, to find the value of OA_MEDIA from the for UNIX prompt:

```
$ echo $OA_MEDIA
```

If the online help files are installed on a machine running a separate web server, omit the /OA_DOC/ setting above, and set this instead in the web server on the machine containing the online help files.

8. Configure the Java Cartridge

Oracle Applications HTML-based functionality requires that you configure Web Application Server to use the correct JDBC drivers. Follow these steps to do so:

- Go to the Web Application Server Administration page on your web server.
- Navigate to the Cartridge Administration page by selecting Web Applications Server Manager, then Oracle Web Application Server, then Cartridge Administration.
- Select the administration page for the Java cartridge, then click on JAVA Cartridge specific parameters.
- Add "%ORACLE_HOME%/jdbc/lib/classes102.zip" to the end of the default setting for CLASSPATH.
- Add "%ORACLE_HOME%/lib" to the end of the default setting for LD_LIBRARY_PATH.
- Add a new row defining the following setting:

```
SYSTEM_PROPERTY    =    APPS_OCI_LEVEL=7
```

- Save the information by clicking the Modify Cartridge Configuration button at the bottom of the page.
- Next, follow the link to return to the JAVA Cartridge Configuration page, and click on Web Request Broker parameters for JAVA.
- In the "Virtual Paths" section, add the following entry:

Virtual Path	Physical Path
/OA_JAVA_SERV	<location of your Java files>

For the physical directory path, enter the location of your Java files as denoted by the \$OA_JAVA environment variable.

- Save the information by clicking the Modify Cartridge button at the bottom of the page.
- Restart the HTTP listeners and Web Request Broker on your web server.

WARNING: Be sure the CLASSPATH variable is set to use the JDK 1.0 library file (classes102.zip) and not the JDK 1.1 file. Otherwise, Oracle Applications HTML-based functionality will fail to operate properly.

9. Create an Oracle Web Application Server Database Access Descriptor (DAD)

You must create a Database Access Descriptor, or DAD, for the Oracle Applications APPS schema, and register the DAD with the Web Request Broker.

- Go to the Web Application Server Administration page on your web server.
- Choose Web Application Server Manager to connect to the Oracle Web Application Server Administration Home page.
- Choose Oracle Web Application Server, then DAD Administration.
- Choose Create New DAD and enter the values appropriate for your schema in the Create New DAD page. For example:

Parameter	Value
DAD Name:	<DAD name>
Database User:	<APPS schema username>
Identified by:	Password
Database User Password:	<APPS schema password>
Confirm Password:	<APPS schema password>
ORACLE_HOME:	<location of your ORACLE_HOME>
SQL*Net V2 Service:	<Net8 database alias>
...	
NLS Language:	<language portion of your NLS_LANG variable>
Check the box for "Store the user name and password in the DAD"	

Note: Refer to your *Oracle Web Application Server* online documentation for more details on how to create a new DAD.

- From the Oracle Web Application Server Administration Home page, choose Oracle Web Application Server, then choose Cartridge Administration.
- In the Cartridge Administration page, choose PL/SQL Cartridge.
- Choose Create New PL/SQL Agent and enter the values appropriate for your schema in the Create New PL/SQL Agent page. For example:.

Parameter	Value
Name of PL/SQL Agent:	<Agent name>
Name of DAD to be used:	<DAD name you created above>
Protect PL/SQL Agent:	True
Authorized Ports:	<port for HTTP listener>

Attention: Make sure you select the option "Install Web Application Server Developer's Toolkit PL/SQL packages" when you create the DAD for the schema.

- Return to the Cartridge Administration page and choose Cartridge Summary (Web Request Broker).
- In the Web Request Broker Administration page, choose Directories in the left-hand frame to display the "Applications and Directories" section in the right-hand frame.
- Scroll to the bottom of the "Applications and Directories" section and add the following line for your PL/SQL agent:

Virtual Path	App	Physical Path
/<agent name>/plsql	PLSQL	<ORAWEB_HOME>/bin

Replace <agent name> with name of the PL/SQL agent you created previously. Replace <ORAWEB_HOME> with the path of your Oracle Web Application Server home.

This step enables you to access your PL/SQL agent by appending the virtual path specified above to your web listener base URL as follows:

```
http://<web server>[:<HTTP listener port>]/<agent name>/plsql
```

For example:

```
http://websvr1.mycompany.com/my_plsqlagent/plsql
```

To invoke any of Oracle Workflow's web services, you simply append the appropriate procedure and arguments to this new base URL. For example, to access the Worklist web page, you would connect to the URL:

```
http://<web server>[:<HTTP listener port>]/<agent name>/plsql/  
wfa_html.worklist
```

10. Modify Web Application Server security level

When setting up a DAD in Oracle Web Application Server, the default behavior is to allow any stored package or procedure in the DAD schema to be executed through a URL. A special security level exists for Oracle Applications so that only stored packages and procedures listed in an AOL table can be invoked in this manner. This prevents users from accessing and executing routines that should never be called directly.

To update this security setting, change to the \$FND_TOP/admin/sql directory, then run SQL*Plus, connecting to your APPS schema, and run the AFOAUTHB.pls script. For example:

```
$ cd $FND_TOP/admin/sql  
$ sqlplus <APPS schema username>/<APPS schema password> @AFOAUTHB.pls
```

WARNING: Failure to run this step will result in a security problem that will allow any user to access Applications data directly.

Configure the Forms Server

The Oracle Applications forms servers must be configured to indicate how they should connect to the Release 11 database server and how the desktop clients should launch the Oracle Forms client applet. You can set up the forms servers in one of two ways: *statically* or *dynamically*. A static configuration appears the same to all users, using the same database connection and applet parameters. A dynamic configuration provides override to default settings and even distributes desktop client connections across multiple forms servers.

We recommend you use the dynamic configuration for easier administration and scalability. In order to do this, you must configure the Oracle Forms Cartridge Handler. If you plan to set up a static configuration, skip to Step 13.

Additional Information: Chapter 2 in *Oracle Developer/2000 Release 1.6: Deploying Applications on the Web* explains how to set up the cartridge and define the cartridge parameters for Version 3 of Oracle Web Application Server. The steps in this section supplement this material.

11. Define Oracle Forms Cartridge parameters

Follow the steps below to configure the Oracle Forms Cartridge on your Oracle Web Application Server.

Note: These steps assume you are using Oracle Web Application Server Version 3.0.1. Contact Oracle Support Services if you are using a later certified version of Oracle Web Application Server.

- Go to the Web Application Server Administration page on your web server.
- Choose Web Application Server Manager to connect to the Oracle Web Application Server Administration Home page.
- Choose Oracle Web Application Server, then Cartridge Administration.
- Choose Add New Cartridge, then follow the link for Add New Cartridge with Manual Configuration.
- Enter the following values on the New Cartridge Configuration screen. Replace <ORACLE_HOME> with the location of the ORACLE_HOME directory on your web server.

In the "New Cartridge Configuration" section:

Parameter	Value
Cartridge Name	apps110
Object Path	<ORACLE_HOME>/lib/f45webc.so
Entry Point	form_entry

In the "Virtual Paths" section:

Virtual Path	Physical Path
/apps110 *	<ORACLE_HOME>/lib

* Be sure the virtual path you choose has *not* also been defined as a virtual directory by the HTTP listener.

Additional Information: For information on the remaining parameters on this screen, click on the Help icon at the bottom of the page in your web browser.

- Click on Register New Cartridge to return to the Cartridge Administration page, then choose the link for the cartridge you just created (apps110), followed by the link for apps110 Cartridge specific parameters.
- Enter the following cartridge parameters on the Cartridge Configuration screen. Replace the values in brackets with the corresponding environment variable values. For example, replace <APPL_TOP> with the value of \$APPL_TOP on your web server.

Parameter Name	Parameter Value
baseHTML	<full path to your initial HTML file>
HTMLdelimiter	%
archive	/OA_JAVA/oracle/apps/fnd/jar/fndewt.jar, /OA_JAVA/oracle/apps/fnd/jar/fndforms.jar, /OA_JAVA/oracle/apps/fnd/jar/fndaol.jar, /OA_JAVA/oracle/apps/fnd/jar/fndctx.jar, /OA_JAVA/oracle/apps/fnd/jar/fndtcf.jar, /OA_JAVA/oracle/apps/fnd/jar/fndhier.jar, /OA_JAVA/oracle/apps/ak/jar/akobjnav.jar
serverPort	<port on which the Forms Server is running>

Parameter Name	Parameter Value
module	<APPL_TOP>/fnd/11.0.28/forms/US/FNDSCSGN
userid	<GWYUID>@<TWO_TASK>
fndnam	<FNDNAM>

Attention: You may need to list additional JAR files for the `archive` parameter if you have installed any patches providing new Java functionality. For example, the 11.0.1 AOL patch set is required to enable the AOL Hierarchy Editor. Refer to the README files accompanying these patches for instructions on what JAR files should be listed to enable the new functionality.

Note: The `code`, `codebase`, and `serverApp` parameters are specified in the initial HTML file, and should not be changed. They do not have to be listed on the Cartridge Parameters screen.

- Once these parameters are set, click the Modify Cartridge Configuration button to save your changes. If you run out of rows to add new parameters, you can click Modify Cartridge Configuration to save your current entries and redraw the screen with additional empty rows.
- Restart the HTTP listeners and Web Request Broker on your web server.

12. Configure Developer/2000 Metrics Server

If you have multiple forms server, you may configure the Developer/2000 Metrics Server to perform load balancing among the servers. This allows client requests to the web server to be redirected to the least loaded forms server for better performance and scalability.

Refer to the section Setting up Load Balancing in the *Oracle Development/2000 Release 1.6: Deploying Applications on the Web* manual for directions on configuring a Metrics Server. When following these instructions, be sure to base your HTML file on the sample HTML file in this release, not on any sample HTML files described in the Developer/2000 manual.

13. Modify the initial HTML signon file

Two versions of a sample initial HTML file are provided for signing on to Oracle Applications. This is the file users will launch from their desktop client machines to sign on to Oracle Applications. Before using one of these sample HTML files, you must edit it to contain specific information about your installation.

The two files, afsampled.htm and afsamples.htm, are located in the \$OAH_TOP/html/US directory. If you did not follow the instructions in Step 11 to set up your forms server dynamically, use the afsamples.htm file as a template for your static initial HTML file. Otherwise, use afsampled.htm, which will work in conjunction with the Forms Cartridge Handler you set up in the previous step to obtain its parameters.

Additional Information: Choosing an HTML file construction method, *Oracle Development/2000 Release 1.6: Deploying Applications on the Web*

▀▀ Editing the afsampled.html file

First, change the title of the page to something other than SAMPLE FILE. Next, modify the serverHost and serverPort parameters. The values you enter will depend on whether you are using load balancing, you are running the Oracle Forms Server on a separate machine from your web server, or you are running the Oracle Forms Server Listener on a port other than the default. Follow the instructions in the comments of the file to determine which lines should be uncommented and modified.

When running over slow network connections, you may want to improve the Applications start-up time by having users download the Oracle Forms client applets from a local file server that is *different* from your forms server. To do this, mount the directory containing the Java files onto each of the desktop client machines and change the value of the archive cartridge parameter on the Cartridge Parameters screen to the new location. You can also specify this value directly in the initial HTML file if you do not want users to be able to override the location in the URL. To do this, change the lines for the archive parameter in the HTML file:

```
archive="/OA_JAVA/oracle/apps/fnd/jar/fndewt.jar,  
        /OA_JAVA/oracle/apps/fnd/jar/fndforms.jar, ...
```

to point to the location of the Applications JAR files using a file-formatted URL. For example, if the Applications JAR files are located at x:\public\java, you would modify the lines in the HTML file as follows:

```
archive="file:///x:/public/java/fndewt.jar,  
        file:///x:/public/java/fndforms.jar, ...
```

Note: If you have configured load balancing among multiple forms servers, do *not* use the LEASTLOADEDHOST token in specifying the URL for the Oracle Forms client applet. Doing so will prevent the applet code from being cached on the desktop client, forcing it to reload every time the user starts up Applications. Refer to the comments in the sample HTML files for assistance.

When you have finished editing this file, save it with a different name. We recommend when possible that you make a link to this file from a common, centralized web page on your organization's intranet, so it can easily be found by users.

► Editing the afsamples.html file

First, change the title of the page to something other than SAMPLE FILE. Next, find the following line in the file:

```
<PARAM name="serverArgs" value="module=%APPL_TOP%/fnd/11.0.28/forms/US/FNDSCSGN  
userid=%GWYUID%@@%TWO_TASK% fndnam=%FNDNAM%">
```

Replace each of the variables surrounded by percent signs with their respective values:

APPL_TOP	This is the top-level directory where your Applications files were unloaded.
TWO_TASK	The TWO_TASK variable points to the Oracle Applications database. Refer to your Net8 documentation if you do not know the variable format required for your network.
FNDNAM	The FNDNAM variable is the ORACLE schema to which the System Administrator responsibility connects. The default username is APPS. The username may be different on your server.
GWYUID	The Gateway ID (GWYUID) is the public ORACLE username and password that grants access to the Oracle Applications signon form. The default username/password is APPLSYSPUB/PUB. The username/password may be different on your server.

The FNDNAM and GWYUID settings can be found in the Oracle Applications environment file.

For example, if you installed the Oracle Applications files in /d01/apps/110, and your database name is “PROD”, your line in this file may look like this:

```
<PARAM name="serverArgs"
value="module=/d01/apps/110/fnd/11.0.28/forms/US/FNDSCSGN
userid=APPLSYSPUB/PUB@PROD fndnam=APPS">
```

Attention: All text after the `value=` parameter should be in one pair of double quotes, and on a single line with no line breaks.

Finally, when running over slow network connections, you may want to improve the Applications start-up time by having users download the Oracle Forms client applets from a local file server that is *different* from your forms server. To do this, mount the directory containing the Java files onto each of the desktop client machines and change the lines for the archive parameter in the HTML file:

```
archive="/OA_JAVA/oracle/apps/fnd/jar/fndewt.jar,
/OA_JAVA/oracle/apps/fnd/jar/fndforms.jar, ...
```

to point to the location of the Applications JAR files using a file-formatted URL. For example, if the Applications JAR files are located at x:\public\java, you would modify the lines in the HTML file as follows:

```
archive="file:///x:/public/java/fndewt.jar,
file:///x:/public/java/fndforms.jar, ...
```

When you have finished editing this file, save it with a different name. We recommend when possible that you make a link to this file from a common, centralized web page on your organization's intranet, so it can easily be found by users.

14. Start HTTP listener and Forms Server Listener processes

At this point, you should start the HTTP listener for your web server and the Oracle Forms Server Listener process. Consult the documentation for information on how to start the HTTP listener.

You must set your Applications environment before starting the Forms Server Listener. For example, if you named your environment file `PRODUS.env`, from a Bourne or Korn shell type:

```
$ . $APPL_TOP/PRODUS.env
```

For more information on starting and stopping the Forms Server Listener, refer to Chapter 2 in *Oracle Developer/2000 Release 1.6: Deploying Applications on the Web*.

Attention: Make sure your PATH variable lists the \$FND_TOP/bin directory *before* the \$ORACLE_HOME/bin directory before starting the Forms Server Listener process. This will ensure the correct version of f45webm is used when starting Oracle Forms.

Set Up Concurrent Managers

Oracle Applications concurrent managers run processes in the background. You must set up and start the concurrent managers for each product group before you can use your Oracle Applications products. Complete the following steps whether you are installing or upgrading.

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

15. Set up electronic mail

Set up electronic mail for the applmgr user before you start the concurrent managers. The concurrent managers send mail to the applmgr user if the Internal concurrent manager stops running. The log file for the Internal concurrent manager shows an error if electronic mail is not set up correctly.

16. Create log and output files in each product top directory

Follow the instructions in this step only if the following are true:

- you are finishing an installation or upgrade on a concurrent processing server,
- you installed multiple product groups,
- you plan to place log and output files under each product's top directory.

Omit this step if you installed only one product group or you set up a common directory (or directories) for use with multiple product groups.

Note: This instructions in this step assume that all product groups use the same set of product files.

Because the default storage mechanism places log and output files under each product's top directory (such as `$GL_TOP`), you need to create log and output directories for each product group under the product top directories:

```
$ cd $<PROD>_TOP
$ mkdir <log1> <out1> ... <logn> <outn>
```

Make these subdirectories for each installed or dependent product. Use the log and output subdirectory names you selected for each product group when you ran AutoInstall. Make sure the directory names match the values of the `APPLLOG` and `APPLOUT` environment variables in the environment file for that product group. Here is an example of creating multiple log and output directories for Oracle General Ledger:

```
$ cd $GL_TOP
$ mkdir logtst outtst logdev outdev
```

In this case, the `logtst` and `outtst` directories are for the first product group, the `logdev` and `outdev` directories are for the second group, and so on.

17. Define startup parameters

The parameters you pass to the concurrent manager startup script, `startmgr`, affect concurrent manager operation. Before running the script, determine the best method to set startup parameter values. You can set these values through any combination of these methods:

- Set values in the `startmgr` script.
- Enter values on the command line when you run `startmgr`.
- Pass values to concurrent managers through your environment.
- Define no values and accept the defaults.

These methods are listed in the order in which they apply. That is, parameter values set in `startmgr` override any other values, and command line values override environment and default values, and so on.

If you use parallel concurrent processing, you should set your startup parameter values in the `startmgr` script. Be sure that you use the same `startmgr` script on all concurrent processing servers.

When the Internal concurrent manager migrates to a new server, it cannot pass any options entered from the command line. Since the Internal concurrent manager migrates to its primary server after it starts, parameters set when you run `startmgr` from any other server are lost.

Additional Information: [Concurrent Managers](#) in Appendix A

Suggestion: Do not set a default value for the sysmgr parameter. This keeps the Oracle Application Object Library password from displaying in the script. You will be prompted for the password when you start the managers.

▮ Setting startup parameter values in the startmgr script

Setting startup parameter values in the startmgr script ensures that the managers always start up with the same values. This is not a flexible way to set parameter values, however, because you must edit the script to change the values, the values apply to all product groups, and the startmgr script is overwritten when you upgrade.

Setting startup parameter values in the startmgr script ensures that the managers always start up with the same values. This is not a flexible way to set parameter values, however, because you must edit the script to change the values, the values apply to all product groups, and the startmgr script is overwritten when you upgrade.

To edit the startmgr script, move to its directory with this command:

```
$ cd $FND_TOP/bin
```

Display the startmgr script with the file editor of your choice and find the following line:

```
eval $@ batchmgr
```

Set startup values by adding the parameters and their values between \$@ and batchmgr. Here is an example:

```
eval $@ mailto=jsmith sleep=30 batchmgr
```

▮ Setting startup parameter values through your environment

To set parameter values through your environment, create a Bourne shell script that defines any number of parameters in this format:

```
<parameter name>=<parameter value>; export <parameter name>
```

If you have multiple product groups, create a different script for each group. Here is an example of a Bourne shell script that exports startup parameters and then calls the startmgr script:

```
:
# This file sets startup parameters for Oracle Applications
# concurrent managers in 'apps' database.
sleep=30; export sleep
mailto=jsmith; export mailto
. $FND_TOP/bin/startmgr
```

With this type of script, the managers inherit the defined parameter values unless different values are set in startmgr. If the script calls startmgr, you cannot enter startup values on the command line. Therefore, ensure that all necessary parameters are defined in the script or in startmgr.

18. Customize startmgr script (optional)

You may want to make additional customizations to startmgr. Because AutoInstall creates a new version of startmgr when you upgrade, you must add to the new script any customizations you made to startmgr in the last release.

19. Restrict access to concurrent processing servers

In Release 11, log and output files are viewed on the desktop client through Report Review Agent and the Web Report Review tool. It is therefore unnecessary for users to have direct access to the concurrent processing servers to view these files. To ensure security of Oracle Applications log and output files, you should limit user access to the concurrent processing servers.

If it is necessary for users to have direct access to the concurrent processing servers for other reasons, follow the steps below to restrict access to the Applications log and output directories.

a. Create a for UNIX group for directory users

The system administrator should create a for UNIX group consisting of applmgr and the users who need access to the directory. For example, the system administrator might create the group glusers for all users of Oracle General Ledger. Groups are defined in the group file of the /etc directory.

b. Change directory file protection

From the applmgr account, change the file protection of a directory so that it has no global privileges:

```
$ chmod 770 <directory>
```

In the following example, the "r" and "x" shown in the system response indicate that there are global read and execute privileges for the directory:

```
$ cd $GL_TOP
$ ls -ldg out
drwxrwxr-x  2 applmgr  misappl   29423   Jan 12 10:27 out/
      ^  ^
```

After the directory privilege is changed, the system response lists three dashes for global privileges, indicating that all global privileges have been revoked:

```
$ chmod 770 out
$ ls -ldg out
drwxrwx---  2 applmgr  misappl   29423   Jan 12 10:27 out/
      ^  ^
```

c. Change directory group

Change a directory's group to the new for UNIX group:

```
$ chgrp <group> <directory>
```

Here is an example:

```
$ chgrp glusers out
$ ls -ldg out
drwxrwx---  2 applmgr  glusers   29423   Jan 12 10:27 out/
```

The system response indicates that the group has been changed from misappl to glusers.

d. Alter file privilege for concurrent manager files (optional)

The operating system user who starts the concurrent managers (applmgr) owns all log and output files created by the managers. Oracle Applications uses the for UNIX command umask to remove write privilege to these files from all users except the owner. You can change the file privileges by changing the value of umask used by Oracle Applications.

The startmgr script uses the umask value set by the environment variable \$APPLMSK in fndenv. You can change this value simply by modifying \$APPLMSK. For higher security, however, override the value of \$APPLMSK in startmgr. If you are setting up parallel concurrent processing, do the same for dcpsbatch.

To modify the umask value in startmgr or dcpsbatch, change to the directory \$FND_TOP/bin and display the script with the file editor of your choice. Find the following line:

```
umask ${APPLMSK-"022"}
```


This line indicates that if \$APPLMSK is not set, the umask command uses the value 022. You can modify the alternate value by substituting a new value. Here is an example:

```
umask ${APPLMSK-"026"}
```

To override \$APPLMSK, use only a three-digit value as the umask argument. Here is an example:

```
umask 022
```

Additional Information: The for UNIX man page for umask contains more information on the umask command and mask settings.

If you change umask values, remember that the owner of a file should always have read and write privilege for the file.

e. Restrict access to scripts (optional)

To enhance security, give only the applmgr login write permission for startmgr and batchmgr. If you are using parallel concurrent processing, do the same for dcpstart and dcpbatch. Also change the restrictions on any custom scripts that call these files. The following commands prevent all users except applmgr from altering the scripts:

```
$ cd $FND_TOP/bin
$ chmod 700 startmgr batchmgr dcpstart dcpbatch
```

20. Start the concurrent managers

You may want to tailor concurrent manager operation through the Oracle Applications System Administration menus before you start the managers. For example, you can create specialized concurrent managers and define primary or secondary servers for parallel concurrent processing.

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

Before starting the managers, log on to the concurrent processing server and run the environment file for the product group. Then run your startup script or call startmgr and enter parameter values directly from the command line:

```
$ startmgr [<parameter> <parameter> ...]
```

Additional Information: [Concurrent Managers](#) in Appendix A

Running the startmgr script starts the internal concurrent manager (parent process). If you have not yet defined additional concurrent managers, the Internal concurrent manager starts the Standard concurrent manager, which by default can process three simultaneous requests.

Additional Information: Administer Concurrent Managers, *Oracle Applications System Administrator's Guide*

Install Parallel Concurrent Processing

Skip to Step 24 if you are not using parallel concurrent processing, or if you are setting up parallel concurrent processing and you installed Oracle Applications product files on a local disk of each concurrent processing server. In this case, each server already has an environment file that defines the server's environment variables.

21. Create new environment files on each concurrent processing server

Each concurrent processing server must have its own environment file so the Oracle Applications environment is properly set up when requests are run from that server. If all product directories on one machine are identically named on other machines, and you do not want to change any log, output, or temporary directory settings, you can copy your existing environment file to all concurrent processing servers. Otherwise, you will have to create new environment files on each server using the AD Administration Utility.

a. Make sure the AD product is locally installed.

If you have not run the adsetup program to install Applications DBA, you must do this before continuing. Follow the steps in Chapter 2, from the beginning if necessary, up to and including the step where you run the AutoInstall setup script (adsetup).

b. Set up your environment

Verify that APPL_TOP is set properly and points to the correct location of the *local* Oracle Applications top directory. Then run the adsetenv script to set up your initial environment:

```
$ cd $APPL_TOP/admin
$ . adsetenv
```

If you just went through the steps to install the AD files, your environment should already be set up correctly.

Note: If you have customizations in your environment files, you should add them to the `adovars.env` file, located in the `$APPL_TOP/admin` directory. See the "[Edit adovars.env file](#)" step in Chapter 2 for more information about this file.

c. Start the AD Administration Utility.

Type the following to start the AD Administration Utility:

```
$ adadmin
```

d. Create an Applications environment file.

Choose the option to Create Applications Environment File from the Maintain Applications Files menu. You should pay special attention to the values you choose for your log, output, and temporary directories.

Note: If you are sharing a common `APPL_TOP` directory across all servers, be sure to choose a name for the environment file that will not overwrite any existing ones in that `APPL_TOP`.

Additional Information: [AD Administration Utility \(adadmin\)](#) in Chapter 5, and [Creating an Environment File](#) in Chapter 3

22. Run the environment file

The environment filename is typically `<dbname>.env`, where `<dbname>` is the name of the database in which the current product group is installed, such as `PRODUS.env`. The following example runs the `PRODUS.env` environment file:

```
$ . $APPL_TOP/PRODUS.env
```

Attention: If you have more than one environment file, run the correct file for a product group before you attempt to finish the installation or upgrade steps for that group.

23. Edit dcpstart on each concurrent processing server

If you are setting up parallel concurrent processing, you need to edit the file dcpstart in the \$FND_TOP/bin directory of each concurrent processing server to set environment variable values for the servers. Copy dcpstart to each server's applmgr home directory.

To edit a dcpstart file, open the file with an editor and find the following lines:

```
ORACLE_HOME=<pathname>; export ORACLE_HOME
ORACLE_SID=<database>; export ORACLE_SID
#TWO_TASK=<database connect string>; export TWO_TASK
```

Change the variable values to the correct values on the server. Either ORACLE_SID or TWO_TASK should be set. You can leave the TWO_TASK line with the comment marker at the beginning of the line (#) if you do not use that variable. If you set TWO_TASK, however, comment the ORACLE_SID line. Here is an example if TWO_TASK is set:

```
ORACLE_HOME=/d01/oracle/8.0.4; export ORACLE_HOME
#ORACLE_SID=proddb; export ORACLE_SID
TWO_TASK=aprod1; export TWO_TASK
```

Below these lines, find the line that runs the environment file. The environment filename is typically <dbname>.env, where <dbname> is the name of the database in which the current product group is installed.

Specify here the full path and filename of the concurrent processing server's environment file. If you installed product files on a local disk, the environment file is in the server's top applications directory. Otherwise it is in the applmgr home directory. Here is an example:

```
. /d01/appl/110/apptest.env
```

If you edited dcpstart in the server's local \$FND_TOP/bin directory, copy the file to the server's applmgr home directory. Each server needs a copy of the edited dcpstart file in its applmgr home directory and, if the server has the Oracle Applications product files on a local disk, in the local disk's \$FND_TOP/bin directory.

Note: The dcpstart file sets the server variables when the applmgr login is accessed remotely from another server. The applmgr .profile file is not run in this case. You also need to set each applmgr .profile file to run its server's environment file. This sets the correct variables when a user logs in as applmgr.

Set Up the Report Review Agent

Oracle Applications uses a tool called the *Report Review Agent* to view concurrent processing files online. This utility resides on the Release 11 database server and uses Net8 to transfer files from the Release 11 database server to the forms server. It then transmits the file contents one page at a time or in its entirety to a desktop client. The transfer is subject to restrictions the system administrator imposes on file transfer size.

24. Configure network configuration files

Attention: If you already configured your concurrent processing server to use the Report Review Agent, you will need to update the directory paths in your Net8 configuration files for the new release. After you have done that, proceed to Step 25.

Part of the installation process in Chapter 2 required you to install and configure Net8 to enable your forms server to communicate with your Release 11 database server. You must now extend this configuration by creating a second database alias for the Report Review Agent on both the forms server and the concurrent processing server.

The following instructions guide you through the process of configuring your network files to use Report Review Agent. These instructions in this section assume that you configured Net8 using a Local Naming resolution method. Contact Oracle Support Services if you are using a different resolution method, such as Host or Centralized Naming, and need assistance with these instructions.

You must create a database alias for the Report Review Agent executable, FNDFS. You should add this database alias to those that were created for the Release 11 database and any others that may already exist in your environment. We include sample listener.ora and tnsnames.ora files on the *Oracle Applications Release 11 Interoperability* CD that you can use as templates. Copy listener.ora from the CD to a temporary directory on your concurrent processing server, and tnsnames.ora to a temporary directory on your forms server. Both files contain sample settings for values such as your hostname or your database name. Open each file with a text editor and update these settings to reflect the configuration on your concurrent processing server.

The sample values and the values to which they should be set are:

Sample Value	Value to replace with
db11	<Applications database SID>
r110sun	<host name of the concurrent processing server>
/d1/oracle	\$ORACLE_HOME on the concurrent processing server
/d2/apps/fnd/11.0.28	\$FND_TOP on the concurrent processing server

These default files assume you use TCP/IP protocol. You may require other modifications if you use a different network protocol.

Additional Information: *Oracle Net8 Administrator's Guide.*

After modifying these files, check the \$ORACLE_HOME/network/admin directory on the forms server for an existing tnsnames.ora file. Then check the \$ORACLE_HOME/network/admin directory of the concurrent processing server for an existing listener.ora file. If either of these files exists, *merge* the contents of the modified template with the corresponding existing file. If you do not already have a tnsnames.ora or listener.ora file, you can copy the new template files directly to the \$ORACLE_HOME/network/admin directory on the correct server. Copy the tnsnames.ora file to the forms server, and the listener.ora file to the concurrent processing server.

Note: You can also follow the instructions in this step to set up the Order Entry Transaction Manager (OEORPC), which is included in the sample files.

You can now restart the Net8 listener on your concurrent processing server so it can recognize the new settings.

Note: Several profile options exist to help you further control the behavior of the Report Review Agent and Order Entry Transaction Manager. You can set these profile options after you have completed the steps in this chapter.

Additional Information: *User Profiles in Oracle Applications System Administrator's Guide*

If you have multiple Release 11 product groups using one or more APPL_TOP directories on a single concurrent processing server, you can use a single Report Review Agent (in other words, the same FNDFS executable) for all product groups, or you can have a separate Report Review Agent service for each one. If you choose to set up only one Report Review Agent service, all product groups will make use of the single FNDFS location you specify when configuring the service. If you want to set up separate Report Review Agent services for each product group using different FNDFS executables, you can set the profile option RRA: Service Prefix to change the default prefix of FNDFS_ in the service name used for that product group.

Attention: If you have multiple *releases* of Oracle Applications sharing the same concurrent processing server, you cannot use the same FNDFS executable for all releases until you have applied Net8 patch 660484. See the *Oracle Applications Release 11 Release Notes* for more information.

25. Verify file permissions for Report Review Agent executable

The Report Review Agent uses an executable in \$FND_TOP/bin called FNDFS. When you configure your network, you must specify the exact location of this file. When the Report Review Agent is activated, the Net8 listener will spawn an FNDFS process that serves files to the forms server. This process runs under the same account as the listener, which is typically the oracle account. This account must have for UNIX file permissions to read the log and report output files for all products, and to read and execute FNDFS. Therefore, if the directories containing log and report output files or the \$FND_TOP directory reside on a remote machine, the account that starts the Net8 listener must have *read* privileges to the report output and log files and *execute* privileges to the \$FND_TOP/bin/FNDFS executable on that machine.

Start Oracle Applications

26. Ensure client software is installed

Verify that all desktop clients have a certified Java-enabled web browser or appletviewer to run the Oracle Forms Java client applet. See the Release Notes for more information on certified clients and special instructions that may apply to this release.

27. Access signon screen from a web browser

Using a Java-enabled web browser or appletviewer on the desktop client, enter the URL of the initial HTML file to access the Oracle Applications signon screen. The signon screen prompts for an Oracle Applications username and password.

Attention: Verify that the Forms Server listener, HTTP listeners, and the Net8 network listeners are running before you attempt to log on to Applications.

On the signon screen, enter a username and password that has been defined by the system administrator. If no usernames or passwords are defined in the database, enter SYSADMIN as the username and the password to log in with the System Administrator responsibility.



28. Set up system administration

To perform system administration tasks, enter the username SYSADMIN and the password SYSADMIN in the Oracle Applications sign-on security form. Refer to the User's Guide for information on using Oracle Applications menus and forms.

Additional Information: *Oracle Applications User's Guide*

Refer to the implementation manual for your product family and then to the *Oracle Applications System Administrator's Guide* for information on setting up system administration. These tasks include registering users and printers. If you are

upgrading from a previous release of Oracle Applications, you only need to review the System Administration manual for any additional functionality that must be implemented in Release 11.

Attention: Be sure to check for any Documentation Updates to the System Administrator's Guide. The Release Notes will indicate whether any patches may need to be applied in order to fully implement System Administration.

To register printers in the Register Printers form of Oracle Applications, your system administrator needs to know each printer's operating system name. Your installation update tells you where to find the printer names for your platform.

Additional Information: Operating System Names for Printers, *Oracle Applications Installation Update* for your platform

Oracle Applications also supports executable programs for printing. You need to create an executable program only for highly complex printing jobs that you cannot format through Oracle Applications printer drivers.

Additional Information: [Printing](#) in Appendix A

Complete Post-Upgrade Steps

29. Run final product upgrade steps

For each product you upgraded, perform the After Upgrading steps described in the *Oracle Applications Upgrade Manual*. You will not be able to implement and use your upgraded products until you have completed these steps. You need to perform these steps once for each product group installed on a server or stand-alone system.

Additional Information: *Oracle Applications Upgrade Manual*

30. Drop obsolete operating unit schemas

If you implemented Multi-Org in Release 10.6, you should drop the obsolete operating unit schemas, as all database access now uses only the APPS schema in Release 11. Use the following commands to drop each operating unit schema:

```
$ sqlplus <SYSTEM username>/<SYSTEM password>
SQL> drop user <operating unit schema> cascade ;
```

It may require several hours to drop each user. The upgrade automatically sets the registered ORACLE ID as Disabled and changes data groups to use the APPS schema instead.

Re-Enable and Integrate Customizations

If you are installing Oracle Applications for the first time, you do not need to perform any of the steps in this section, and can skip to Step 35.

31. Reapply product customizations to files and databases

If your site had any product customizations, such as customized versions of Oracle Applications forms, reports, programs, or libraries, you may need to reapply changes. You must always test that they work with the new release.

Customizations that were not stored in the standard Oracle Applications directory structure are not affected by the upgrade.

If your product customizations were previously stored in the standard Oracle Applications directory structure, you must reintegrate them using a custom application directory structure. This isolates your changes and makes applying upgrades and patches easier. You may need to retrieve your customizations from the backup you made before the upgrade.

Some of your customizations may no longer work correctly or may no longer be necessary. If the version of the Oracle Applications file you customized has not changed, you need do nothing extra. Depending on the nature of the changes, you can copy the new version of the Oracle Applications files to your custom application directory and then reapply your customizations. You should test all of your customizations thoroughly after you reapply them.

If you made customizations to any of the Applications environment files in the past (APPLSYS.env, fndenv, or devenv), you should re-integrate these into the adovars.env file, located in the \$APPL_TOP/admin directory.

Additional Information: [Edit adovars.env file](#) in Chapter 2.

Attention: After making changes to environment files, you must re-read the environment files and restart the concurrent managers in order for concurrent programs to run in the new environment.

In addition, all custom seed data must be registered under a custom application, not an Oracle Applications product. For example, a custom report would be owned by Custom AP not Oracle Payables. This ensures your custom seed data is not deleted or modified on upgrade.

Additional Information: [Back up the Oracle Applications database \(recommended\)](#) in Chapter 2

32. Verify installation group numbers

Ensure that the Installation Group number is set correctly for custom ORACLE IDs. As System Administrator, navigate to Security:ORACLE:Register, query your custom ORACLE IDs, and set the value of the Installation Group appropriately.

Additional Information: [Set up AutoInstall installation groups](#) in Chapter 2

33. Integrate custom objects and schemas

If you are upgrading from a previous release and have custom objects or custom schemas that need to be tightly integrated with Oracle Applications, follow the steps in Appendix A to re-integrate these customizations with the APPS schema.

Additional Information: [Integrating Custom Objects and Schemas](#) in Appendix A.

If you have custom database objects, You should follow certain naming standards so your custom object names will not conflict with Oracle Applications.

WARNING: When naming database objects, use at least four characters for your product short name, followed by an underscore. You must use only letters, digits, and underscores when naming database objects. For example, you might define your custom application to use the short name CUST, and database objects to begin with a CUST_ prefix.

Additional Information: Naming Standards, *Oracle Applications Developer's Guide*

34. Re-enable custom triggers, constraints, and indexes

An upgrade may modify data your custom triggers or constraints would have processed. Since you disabled these triggers and constraints to ensure the upgrade proceeded correctly, you need to identify any data updates that must occur before you re-enable your custom triggers.

If you dropped any custom indexes before the upgrade for performance reasons, review the new data model to determine if the index is still necessary before redefining it.

Additional Information: [Disable custom triggers, constraints, and indexes](#) in Chapter 2

Tasks To Do Any Time After Installing or Upgrading

The following steps should be performed whether you installed a new system or upgraded an existing system. They can be done any time after the installation, but we recommend doing them before users begin to access Oracle Applications for the first time.

35. Change Oracle Application schema passwords (recommended)

To enhance database security, change the password for every Oracle Application schema.

Additional Information: [Changing Oracle Applications Passwords](#) in Appendix A

36. Pin SGA packages and sequences (recommended)

Oracle Applications requires space in the ORACLE System Global Area (SGA) to execute stored packages and functions. If SGA space is fragmented, there may not be enough space for a package or function. You should pre-allocate space in the SGA shared pool for packages, functions, and sequences by "pinning" them.

Appendix A explains how to run the package pinning scripts and how to tell what objects are currently pinned in the SGA.

Additional Information: [Pinning Packages and Sequences](#) in Appendix A.

37. Install help text

The *Oracle Applications Documentation* CD contains four versions of Oracle Applications Help text in compressed (zip) format. Choose the version of help text appropriate to your installation. The four versions are:

File Name	Description
comnls.zip	Worldwide version (source for translation)
comuk.zip	United Kingdom commercial version
comus.zip	United States commercial version
govus.zip	United States public sector version

Only the documentation in the comnls.zip file is available in translations for use with NLS releases of Oracle Applications. If you are installing Oracle Applications in more than one language, you may direct users to the comnls version of help text so that the same documentation will be available in all languages.

To install the help text, go to the \$OAH_TOP directory, then use the unzip program to extract the contents of the archive file to that location. For example, if your CD-ROM is mounted as /cdrom, you would enter the following commands to extract the files from the comus.zip file into the \$OAH_TOP directory:

```
$ cd $OAH_TOP
$ unzip /cdrom/comus.zip
```

The unzip program should be in your path, or you should specify the full path to the program on your local machine. If the unzip program is not available on your machine, we have included a copy of InfoZip's implementation of unzip for several different platforms on this CD. You can also download the program via the World Wide Web at <http://www.cdrom.com/pub/infozip/>.

After you have installed the on-line help files, make sure you do the following:

- Verify the web server hosting the on-line help has the correct setting for the virtual directory /OA_DOC

Additional Information: Step 7 in this chapter.

- Set the "Help System Base URL" profile option to the URL that will be used to access the on-line help

Additional Information: User Profiles in *Oracle Applications System Administrator's Guide*.

- Review the *Oracle Applications Installation Release Notes* for any additional steps that need to be done for software running on the desktop clients.

38. Delete obsolete product files (optional)

After you are satisfied with the upgrade, you can delete the product files for the previous release of Oracle Applications (if you have not done so already). For each product, you may want to retain report output files or customized programs. Output files are stored in the old log and output subdirectories under each product's top directory, or under the log and output directories you created, or under a common directory.

To remove the obsolete files for an old release, change to the top directory of the old release and enter the following command:

```
$ rm -r <old APPL_TOP>
```

Here is an example:

```
$ rm -r /d01/appl/r10
```

39. Back up Oracle Applications (recommended)

Your operating system administrator should back up the Oracle Applications product files. On a server or stand-alone machine, your database administrator should also back up the Oracle Applications database.

40. Apply patches

Apply any other patches provided by Oracle.

Additional Information: [AutoPatch Utility \(adpatch\)](#) in Chapter 5

Installation Utilities

This chapter contains information about the utilities that you use when you install, upgrade, patch, or use Oracle Applications:

- [AD Controller Utility \(adctrl\)](#)
- [AD Administration Utility \(adadmin\)](#)
- [AutoPatch Utility \(adpatch\)](#)
- [Unload Utility \(adunload\)](#)
- [Relink Utility \(adrelink\)](#)
- [Character Set Conversion Utility \(adncnv\)](#)
- [ODF Comparison Utility \(adodfcmp\)](#)
- [AD File Identification Utility \(adident\)](#)
- [AD Configuration Utility \(adutconf.sql\)](#)

AD Controller Utility (adctrl)

By using the AD Controller utility, you can determine the status of AutoInstall workers and restart failed AutoInstall tasks. Log in as applmgr and run the product group's environment file. Then give the following command to start the utility:

```
$ adctrl
```

The utility displays a menu after you do the following:

- Confirm the value of \$APPL_TOP.
- Specify an AD Controller log file (the default is adctrl.log). The log file resides in \$APPL_TOP/admin/<dbname>/log, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable.
- Supply the Oracle Application Object Library username and password of the product group.

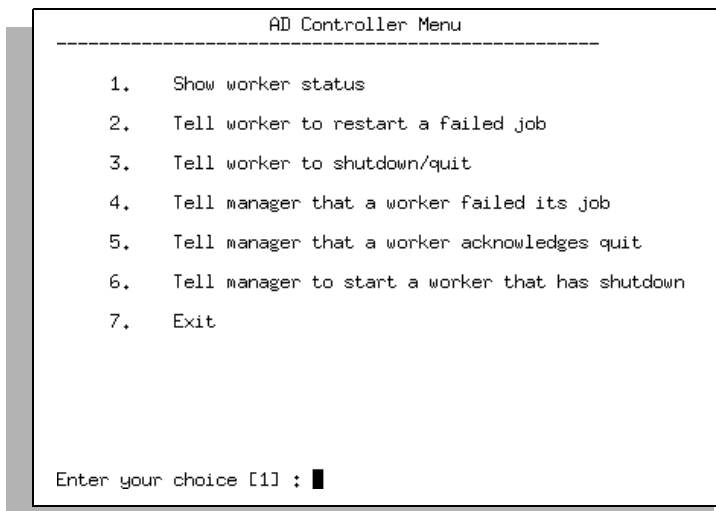


Figure 5–1 AD Controller main menu

Type the menu number to select an option. You typically use only options 1, 2, and 7. You can return to the menu after selecting an option by pressing [Return].

Additional Information: [Monitoring AutoInstall](#) in Chapter 3.

Attention: At various times during the installation process, AutoInstall will try to drop the FND_INSTALL_PROCESSES table, which AD Controller uses to determine job statuses. Exit AD Controller when you are not using it so that AutoInstall can successfully drop this table.

Running the AD Controller Utility

Do the following before you run the AD Controller utility:

- ❑ Log in as applmgr.
- ❑ Run the environment file for the appropriate product group.
- ❑ Check that \$ORACLE_HOME is set to the proper database directory, and that \$ORACLE_SID or \$TWO_TASK identify the correct database.

Additional Information: 13: Verify \$ORACLE_HOME, \$ORACLE_SID, and \$TWO_TASK in Chapter 5.

Start the utility with this command:

```
$ adctrl
```

The utility then asks you a series of questions.

Log File

If an error occurred or if you are unsure of messages returned by AD Controller, review the file adctrl.log in \$APPL_TOP/admin/<dbname>/log, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable. As it runs, the AD Controller utility appends information to the end of this file.

You can delete the adctrl.log file if it is large and contains no necessary information. In this case, the controller utility creates a new adctrl.log file the next time it runs.

Reviewing Worker Status

When you select option 1 to show worker status, AD Controller displays a summary of current worker activity. Here is an example:

Control				
Worker	Code	Context	Filename	Status
-----	-----	-----	-----	-----
1	Run	Installing at R11028	afasdfa.dat	Running
2	Run	Installing at R11028	aftxt.drv	Failed
3	Run	Installing at R11028	afatsaf2.sql	Running
4	Run	Installing at R11028		Wait
5	Run	Installing at R11028		Wait

The following table explains the entries that you may see in the Status column. Usually, a Failed status indicates a problem you need to correct.

Table 5–1 Work Statuses

Status	Meaning
Assigned	The manager assigned a job to the worker and the worker has not noticed it yet.
Completed	The worker completed the job and the manager has not yet assigned it a new job.
Failed	The worker encountered a problem.
Fixed, Restart	You fixed the problem and the worker should retry whatever failed.
Restarted	The worker is retrying a job or has successfully restarted a job. (Note that the status does not change to Running.)
Running	The worker is running a job.
Wait	The worker is idle. The manager cannot assign a job to the worker because other jobs have to complete first.

Restarting a Failed Worker

Perform the following steps to restart a failed worker after you have taken whatever actions are necessary, if any, to correct the problem that caused the failure:

1. Choose option 1 to review worker status and confirm the Failed status of the worker. The Filename column lists the name of the file that failed to run.
2. Choose option 2 to tell the worker to restart a failed job. When prompted, enter the number of the worker that failed. If all workers failed, type “all”.
3. Choose option 1 again. The Status column for the worker that failed should now say Fixed, Restart or Restarted.

If a worker fails on the same job after a restart and you do not know the reason, contact Oracle Support Services.

Note: If a change to the environment variables is done while AutoInstall is running, the new settings do not take effect until AutoInstall is cancelled and all workers are restarted.

Restarting a Terminated Worker

If a worker fails a job, it usually reports the failure to the manager and waits for its status to be changed. For severe errors, however, the worker itself may terminate without reporting failure. Here is an example of termination for a severe error from a worker log file:

AD Worker error:

The following ORACLE error:

```
ORA-02709: osnpop: pipe creation failed; max open file descriptors exceeded
occurred while executing the SQL statement:
```

```
SELECT status, control_code, context,
       pdi_product, pdi_username
       command, file_product, subdirectory, filename
       arguments
FROM fnd_install_processes
WHERE worker_id = 1
```

AD Worker is complete.

In this case, the worker process terminated because it could not select from the FND_INSTALL_PROCESSES table used to communicate with the manager. Follow these steps to restart the worker after you have fixed the problem that caused the failure:

1. Choose option 1 to show worker status. The Status column will show the worker as Running or Restarted instead of Failed.
2. Choose option 4 to indicate that the worker has failed.
3. Now use option 6 to restart the worker. When prompted, enter the number of the worker that failed.

Reactivating the Manager

A restarted worker resumes the failed job immediately as long as the manager is active. The other workers will change to a Waiting status if they cannot run any jobs because of dependencies on the failed job, or because there are no jobs left. When no workers are able to run, the manager becomes idle. You will see messages like the following on the terminal or window running AutoInstall when the manager becomes idle:

```
FAILED: file aftwf01.sql on worker 3 for product fnd username APPLSYS.
```

ATTENTION: All workers either have failed or are waiting:

```
FAILED: file cedropcb.sql on worker 1.  
FAILED: file adgrnctx.sql on worker 2.  
FAILED: file aftwf01.sql on worker 3.
```

ATTENTION: Please fix the above failed worker(s) so the manager can continue.

To restart a worker after the manager has stopped, set the worker to restart as described above. AutoInstall automatically reactivates, then restarts the worker processes, displaying a message like the following:

```
Restarted: file aftwf01.sql on worker 3 for product fnd username APPLSYS.
```

Shutting Down AutoInstall

Perform the following steps if you must stop AutoInstall to shut down the database or the machine.

1. Select AD Controller option 3 and enter *all* for the worker number. Each worker stops once it completes or fails its current job.
2. Verify that no worker processes are running with the following command. This is an example of a typical command; note that the command arguments may vary on different platforms.

```
$ ps -a | grep adworker
```

Any worker processes that show up are still running.

3. When all workers have shut down, AutoInstall exits.

Controlling the Number of Workers

If you find that you often have workers in Waiting status, you may be running more workers than necessary. In this case, you can shut down individual workers, rendering them unavailable for the duration of the operation. You can not, however, add additional workers, if you find that your operation might be more beneficial with additional parallelization. You therefore might want to choose a large number when prompted for the number of workers to run, then shut down any workers that appear to be idle most of the time. This is a good method for determining the ideal number of workers needed for an upgrade in your hardware environment.

AD Administration Utility (adadmin)

With the AD Administration utility, you can perform maintenance tasks on an installed Oracle Applications system. The utility has two menus:

- **Maintain Applications Database Objects**
Use to perform actions with installed database objects, such as recreating grants and synonyms between base schemas and the APPS schema.
- **Maintain Applications Files**
Use to perform actions with product files, such as relinking product executable files with the Oracle8 Server.

Running the Administration Utility

Do the following before you run the AD Administration utility:

- ❑ Log in as applmgr.
- ❑ Run the environment file for the appropriate product group.
- ❑ Check that \$ORACLE_HOME is set to the proper database directory, and that \$ORACLE_SID or \$TWO_TASK identify the correct database.

Additional Information: [Verify \\$ORACLE_HOME, \\$ORACLE_SID, and \\$TWO_TASK](#) in Chapter 2.

- ❑ Shut down the concurrent managers if you plan to relink Oracle Applications product files or choose any of the tasks from the Maintain Applications Database Objects menu.

Additional Information: Administer Concurrent Managers, *Oracle Applications System Administration Reference Manual*

Start the utility with this command:

```
$ adadmin
```

The utility then asks you a series of questions.

Log File

If an error occurred or if you are unsure of messages returned by AD Administration, review the file `adadmin.log` in `$APPL_TOP/admin/<dbname>/log`, where `<dbname>` is the value of your `ORACLE_SID` or `TWO_TASK` variable. As it runs, AD Administration appends information to the end of this file.

You can delete the `adadmin.log` file if it is large and contains no necessary information. In this case, the Administration utility creates a new `adadmin.log` file the next time it runs.

AD Administration Questions

The following sections explain the questions the AD Administration utility asks you about your system. If you restart a session that did not run to completion, the utility may not ask all of these questions.

Stopping and Restarting AD Administration

You can exit the utility by entering `abort` at any prompt. Restart the utility by giving the command `adadmin` again. When you restart the utility, you can specify the log file from the previous session or enter a new filename. When you reuse a log file, the utility adds the message “Start of AD Administration session” to the end of the file and appends messages from the new session as it generates them.

The utility then asks if you want to continue with your previous session or start a new session:

- Continue Session

Continuing the previous session is the default. When you choose to continue, the utility determines where your last session stopped and restarts at that point.

- Start New Session

The utility asks you to confirm your choice if you choose not to complete the previous session. It then restarts from the beginning.

AD Administration Prompts

When the AD Administration utility prompts for information, it typically includes a default answer in square brackets:

AD Administration records your AD Administration session in a text file you specify. Enter your AD Administration log file name or press [Return] to accept the default file name shown in brackets.

Filename [adadmin.log] :

You can accept the default value (adadmin.log) by pressing [Return]. To specify a new value, type the value and press [Return].

Applications Directory

The utility first shows the value of the environment variable APPL_TOP and asks if this is the top applications directory. Press [Return] to accept this default value. If the default directory is not correct, answer No at the prompt to exit. Resume running the utility after you set \$APPL_TOP to the correct directory.

Additional Information: [Verify \\$ORACLE_HOME, \\$ORACLE_SID, and \\$TWO_TASK](#) in Chapter 2.

Log File

The utility prompts for the name of a log file that records the session. The default filename is adadmin.log. The utility places the log file in \$APPL_TOP/admin/<dbname>/log, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable. You should review the log file for error messages after you run the utility. Note that there may also be one or more worker files if you are running database steps that operate in parallel mode. Review these adwork<number>.log files for detailed information about the error.

Email Notification

You can choose to be notified by email if an AutoInstall worker fails at any time during the installation process. Enter your email address, e.g.: appldba@mycompany.com) at the AutoInstall prompt. If you want more than one person to be notified, enter each email address separated by a space. The email notification will contain the last 100 lines of the worker log (that failed).

Batch Size

AutoInstall prompts you to enter a batch commit size to be used during the upgrade. If you do not specify a value, it uses a default batch commit size, which is set to a relatively small value to accommodate systems with small rollback segments.

Batch commit size determines the number of rows to commit at one time when certain scripts run. To take advantage of large rollback segments, you must specify a batch commit size larger than the default value.

Files Installed in APPL_TOP

The AD Administration tool verifies what types of files you have installed in your current APPL_TOP. Answer yes or no to each of these questions to indicate to adadmin what functions the current APPL_TOP is performing.

Additional Information: [Installing Files in APPL_TOP](#) in Chapter 3.

ORACLE Database

The AD Administration utility lists the database and database directory you are set to use. It asks you to confirm that you want to use or modify Oracle Applications in this database with the files in this directory. Here is an example:

```
You are about to use or modify Oracle Applications product tables
in your ORACLE database 'apptest'
using ORACLE executables in '/d01/app/oracle/prod/8.0.4'.
```

```
Is this the correct database [Yes] ?
```

If adadmin does not list the correct database and directory, answer No at the prompt to exit. Resume running the AD Administration Utility after you change the values of \$ORACLE_SID or \$TWO_TASK, and \$ORACLE_HOME to identify the correct database and directory.

Additional Information: [Verify \\$ORACLE_HOME, \\$ORACLE_SID, and \\$TWO_TASK](#) in Chapter 2.

SYSTEM and AOL user passwords

AutoInstall next asks for the server's SYSTEM password. It then connects to the database and determines the ORACLE username for the AOL schema that uniquely identifies your existing product group. AutoInstall prompts for the password of this schema, then verifies the connection.

AutoInstall will now obtain the rest of the information it needs directly from the database, including the installed languages, module information, and product dependencies. Once this is finished, the AD Administration Main Menu will appear.

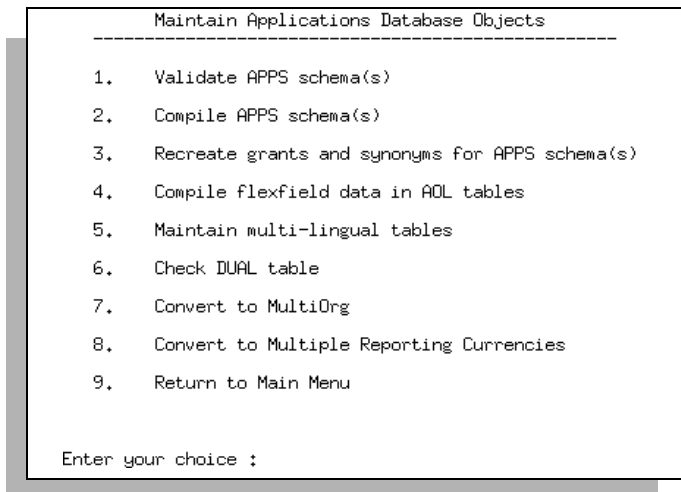
Main Menu

From the Main Menu, you can choose to maintain Applications database objects, maintain Applications files, or exit the utility. The following sections describe the first two menu options in further detail.

Maintaining Applications Database Objects

You use the Maintain Applications Database Objects menu to perform tasks that involve Oracle Applications database objects. Select the task number to perform a task. You should only run the tasks in this section on your administration server.

Note: Some tasks may not be available depending on the configuration of your Applications database. The options and respective numbers shown in the figure below may differ slightly with those for your installation.



```

-----
Maintain Applications Database Objects
-----
1.  Validate APPS schema(s)
2.  Compile APPS schema(s)
3.  Recreate grants and synonyms for APPS schema(s)
4.  Compile flexfield data in AOL tables
5.  Maintain multi-lingual tables
6.  Check DUAL table
7.  Convert to MultiOrg
8.  Convert to Multiple Reporting Currencies
9.  Return to Main Menu

Enter your choice :
  
```

Figure 5–2 *Maintain Applications Database Objects menu*

The first time you select a task from the Database Objects menu, AD Administration checks your init.ora parameters. It only checks the parameters for the first task you select, and does not recheck if you select other tasks in the same AD Administration session.

- Validate APPS schema(s)

This task runs a SQL script against each of your APPS schemas, verifying the integrity of each schema, and checking for some conditions which are undesirable, but will not produce fatal problems.

An output file, <APPS schema name>.lst, is produced for each APPS schema. The file is located at \$APPL_TOP/admin/<dbname>/out, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable. You should review this output file and fix all problems reported in order to avoid potential problems while running or maintaining Applications. The output file for each APPS schema contains information on how to fix each reported problem.

- Compile APPS schema(s)

This task spawns parallel workers to compile invalid database objects in your APPS schemas. It uses the same parallel compile phases as AutoInstall.

- Recreate grants and synonyms for APPS schema(s)

This task recreates the grants and synonyms for the Oracle Applications public schema (APPLSYSPUB), recreates grants on some packages from SYSTEM to APPS, then spawns parallel workers to recreate grants and synonyms linking sequences and tables in the base schemas to the APPS schemas.

- Compile flexfield data in AOL tables

This task compiles flexfield data structures in Oracle Application Object Library tables. If you choose not to compile the structures, each is compiled the first time any user accesses the flexfield.

- Maintain multi-lingual tables

This task calls PL/SQL routines to maintain multilingual tables for some Applications (including AOL, AK, and AZ) by adding any missing, untranslated rows. You generally only need to run this task when instructed to by Oracle Support Services.

- Check DUAL table

This task verifies that the DUAL table exists in the SYS schema, is accessible by Applications, and contains only one row.

- Maintain MultiLingual schema(s)

This task only displays on the Database Objects menu if you have Multiple Language Support functionality installed in your database.

The Maintain MultiLingual schema(s) task updates system privileges and grants to your existing Multiple Language Support (MLS) schema(s), then calls a PL/SQL procedure which maintains database objects in your MLS schema(s).

For each APPS schema, MLS is implemented using an extra schema, called an “adjunct schema”, which contains synonyms to objects in the APPS schema, exact copies of some objects in the APPS schema, and modified copies of other objects in the APPS schema.

Choose this task after applying any database patch in order to synchronize the database objects in your MLS schemas with those that may have been updated in your APPS schemas.

- **Maintain Multiple Reporting Currency schema(s)**

This task only displays on the Database Objects menu if you have Multiple Reporting Currencies functionality installed in your database.

Multiple Reporting Currencies (MRC) is an Oracle Applications feature that you can install using the "Convert to Multiple Reporting Currencies" option on the AD Administration database menu (see below).

For each APPS schema, MRC is implemented using an extra schema, called an “adjunct schema”, which contains synonyms to objects in the APPS schema, exact copies of some objects in the APPS schema, and modified copies of other objects in the APPS schema.

Additional Information: Multiple Reporting Currencies in Chapter 3 of *Oracle Applications Concepts*.

The Maintain Multiple Reporting Currency schema(s) step does the following:

- Asks for the number of parallel workers (for compiling invalid objects in parallel)
- Updates system privileges and grants to your existing MRC schema(s)
- Calls a PL/SQL procedure which maintains database objects in your MRC schema(s)
- Compiles all invalid objects in your MRC schema(s) in parallel

Choose this task after applying any database patch in order to synchronize the database objects in your MRC schemas with those that may have been updated in your APPS schemas.

- Convert to Multi-Org

This task only displays on the Database Objects menu if you do not already have Multi-Org installed in your database and if you do not currently have Multiple Sets of Books Architecture installed in your database.

This task converts a standard product group (not Multiple Sets of Books and not Multi-Org) into a Multi-Org product group with one operating unit defined at the site level. Before running this step, you must define an Operating Unit and set the site-level AOL profile option MO: Operating Unit to use your new operating unit. This profile option tells AD Administration what operating unit it should use when converting your existing data. If you define additional operating units, the seed data will be replicated for all operating units.

WARNING: To avoid the possibility of data corruption, you must shut down all concurrent managers and ensure all users are logged off the system prior to and during this step.

Attention: The Convert to Multi-Org task may take anywhere from a few hours, for a newly implemented fresh install database, to much longer, for an existing production database with much data. Part of the conversion process involves updating every row in every organization-sensitive table with the ORG_ID for the default operating unit.

The Convert to Multi-Org task does the following:

- Asks for the number of parallel workers (for compiling invalid objects in parallel)
- Confirms that you want to run this task
- Creates scripts to disable and re-enable triggers in your APPS schema
- Disables all triggers in your APPS schema
- Converts seed data and transaction data to Multi-Org in parallel
- Re-enables all previously-enabled triggers in your APPS schema

- Convert to Multiple Reporting Currencies

This task only displays on the Database Objects menu if you do not already have Multiple Reporting Currencies functionality installed in your database.

WARNING: To avoid the possibility of data corruption, all users must log off the system prior to and during this step, and you must shut down the concurrent managers.

The Convert to Multiple Reporting Currencies task does the following:

- Asks for the number of parallel workers (for compiling invalid objects in parallel)
- Confirms that you want to run this task
- Creates your Multiple Reporting Currencies (MRC) schema(s)
- Creates the correct system privileges and grants in your MRC schema(s)
- Registers your MRC schema(s) with Oracle Applications
- Sets up one or more data groups for your MRC schema(s)
- If running in a configuration using Multiple Set of Books Architecture, makes multiple copies of MRC responsibilities in all other sets of books
- Enables the Multiple Reporting Currencies feature
- Calls a PL/SQL procedure which creates database objects in your MRC schema(s)
- Compiles all invalid objects in your MRC schema(s) in parallel
- Creates MRC triggers in your APPS schema(s)

- Return to main menu

Select this task to return to the AD Administration main menu.

Maintaining Applications Files

You use the Maintain Applications Files menu to perform tasks related to the Oracle Applications product files.

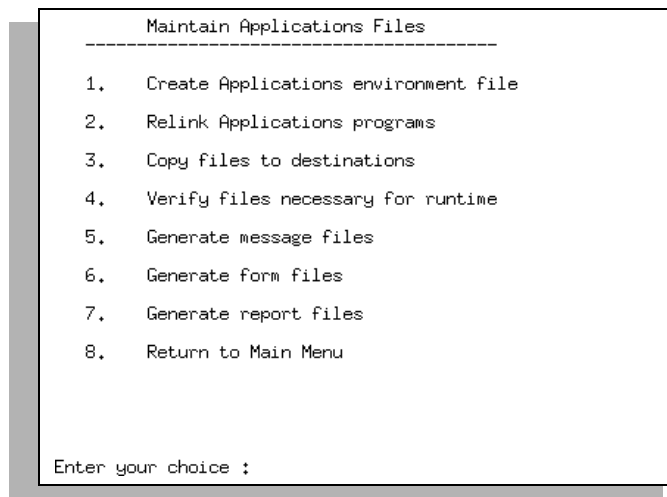


Figure 5–3 *Maintain Applications Files menu*

Select the task number to perform a task.

1. Create Applications environment file

This task creates an environment file that defines Oracle Applications environment variables. It asks for the filename as a parameter. The environment file can have any name, but it should use the .env extension.

The default name is the name of your database (denoted by the value of the environment variable ORACLE_SID or TWO_TASK) and the extension .env. If your database name is “apptest”, the environment file will be created with the name apptest.env.

Attention: Each product group should have an environment file with a unique name. When you recreate an environment file, make sure that you do not overwrite the file for a different product group sharing the same APPL_TOP.

After you supply the filename, the utility asks for information necessary for the Applications environment.

Additional Information: [Creating an Environment File](#) in Chapter 3.

After the utility generates the environment file, you can customize and run the file as necessary.

2. Relink Applications programs

This task relinks Oracle Applications executable programs with the Oracle8 Server libraries. These programs must be linked to function with the Oracle8 Server database.

You can relink executables for all products or for a specific set of products. For each product, you can also choose whether to link all executables, or only specific ones.

You also have the option of relinking executables with debugging information intact if asked to by Oracle Support Services. By default, AD Administration relinks all executables without debug information.

AD Administration does not link executables for the product AD. If you need to relink AD executables, you must run the adrelink utility manually.

Additional Information: [Relink Utility \(adrelink\)](#) in this chapter.

3. Copy files to destinations

This task copies files from each product area into central locations where they can be easily referenced by Applications programs. You may choose to overwrite existing files in the other destinations, or you may choose to only copy files to the other destinations if they currently do not exist in the destination directory.

The file types and their respective destinations are shown in the following table:

These files...	are copied to...
Java files	\$JAVA_TOP
HTML files	\$OAH_TOP
Media files	\$OAM_TOP

Additional Information: Edit adovars.env file in Chapter 2.

4. Verify files necessary for runtime.

This task verifies that all files needed to run Oracle Applications for the current configuration are in the current APPL_TOP.

5. Generate message files

This task generates message binary files (extension .msb) from Oracle Application Object Library tables. Oracle Applications uses the message binary files to display messages for users.

6. Generate form files

This task generates binary Oracle Forms files (extension .fmx) for all installed languages from the form definition files (extension .fmb). The definition files are located under the \$AU_TOP directory, whereas the binary files are stored under each product's directory. Oracle Applications uses the binary form files to display data entry forms.

The "Generate form files" step does the following:

- Prompts for the number of parallel workers (for generating Oracle Forms files in parallel)
- Displays the current character set (from NLS_LANG) and asks if you want to generate Oracle Forms objects in this character set
- Asks if you want to regenerate Oracle Forms PL/SQL library files
- Asks if you want to regenerate Oracle Forms menu files
- Asks if you want to regenerate Oracle Forms executable files
- Asks for which products you want to generate Oracle Forms objects
- Asks if want to generate specific forms objects for each selected product
- Creates a list of all Oracle Forms objects to generate
- Displays the list of Oracle Forms objects to be generated and allows you to select whether to regenerate specific objects or all objects. (This step is only done if you chose to generate specific Oracle Forms objects for each selected product.)
- Generates all selected Oracle Forms objects for all selected products in parallel

Finally, if any Oracle Forms objects did not generate successfully, it displays the list of objects that had warnings or errors, and asks if you want to continue as if

successful. You may need to review the adadmin.log file to determine whether or not the problems require additional attention.

Additional Information: [Form Generation Warnings](#) in Chapter 3.

7. Generate report files

This task generates the binary Oracle Reports report files (extension .rdf) for all installed languages.

The "Generate report files" step does the following:

- Displays the current character set (from NLS_LANG) and asks if you want to generate Oracle Reports objects in this character set
- Asks if you want to regenerate Oracle Reports PL/SQL library files
- Asks if you want to regenerate Oracle Reports executable files
- Asks for which products you want to generate Oracle Reports objects
- Asks if you want to generate specific reports objects for each selected product
- Creates a list of all Oracle Reports objects to generate
- Displays the list of Oracle Reports objects to be generated and allows you to select whether to regenerate specific objects or all objects. (This step is only done if you chose to generate specific Oracle Reports objects for each selected product.)
- Generates all selected Oracle Reports objects for all selected products

Finally, if any Oracle Reports objects did not generate successfully, it displays the list of objects that had warnings or errors, and asks if you want to continue as if successful. You may need to review the adadmin.log file to determine whether or not the problems require additional attention.

8. Return to Main Menu

Select this task to return to the AD Administration main menu.

AutoPatch Utility (adpatch)

You use AutoPatch to apply individual bug patches, patch sets, or release updates. A patch set is a collection of individual patches for a product, while a release update is a collection of patch sets for all Applications products. Release 11.1 would be a minor release, and would be installed using AutoInstall. Release 11.0.1 would be a release update, and would be installed with AutoPatch. AutoPatch may ask extra questions and do extra checking when you apply a release update. After you apply a release update, AutoPatch rewrites the information in your database to indicate the current release update.

Attention: Pay special attention to the following important information:

- If you run AutoInstall again to license new products, you should reapply the file and generation portions of all patches previously applied. When licensing new localizations, you should reapply *all* portions of all patches previously applied.
 - You may not use AutoPatch for custom development.
-
-

How AutoPatch Works

The AutoPatch utility replaces some of your existing product files with new versions of those files included with the patch and may also make changes to your Oracle Applications database objects. The utility resides in the \$AD_TOP/bin directory. The following steps summarize the way AutoPatch works:

1. You log in as applmgr and set up your environment.
2. You unload the new software into a patch directory and start the AutoPatch utility under a Bourne or Korn shell.
3. After you answer some questions, AutoPatch identifies the Oracle Applications products that need to be updated.
4. AutoPatch extracts the appropriate files from each product's C library.
5. AutoPatch compares the extracted object modules with their corresponding files in the patch directory. It also makes this type of comparison with files such as forms, reports, and SQL scripts.
6. If a file in the patch directory is a more recent version than the product's current file, AutoPatch backs up the product's current file by appending "O" to its filename.

7. AutoPatch then replaces each product's outdated files with newer files from the patch directory.
8. AutoPatch loads the new object modules into the C libraries.
9. AutoPatch backs up any files which you listed in adlinkbk.txt and which will be relinked.
10. AutoPatch relinks the Oracle Applications products with the Oracle8 Server.
11. AutoPatch runs sql scripts and exec commands, which change Oracle Applications database objects. By default, AutoPatch does this in parallel.
12. AutoPatch copies any specified Java, HTML, or media files to their respective destinations.
13. AutoPatch generates Oracle Forms files.
14. AutoPatch generates Oracle Reports files.
15. AutoPatch appends a record of how it changed your system to applptch.txt in the \$APPL_TOP/admin/<dbname> directory, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable.

Patch Format

Bug patches generally consist of a top-level directory, several files in the top level directory, and one or more subdirectories. The top-level directory is usually named <bugnum>, where <bugnum> corresponds to the Oracle Bug Database bug number for the patch.

The important files in the top level directory are: readme.txt, c<bugnum>.drv, and d<bugnum>.drv, and g<bugnum>.drv.

The file readme.txt contains general information about the bug patch. It may describe manual steps that you must perform as part of fixing the bug, and usually indicates on what servers you must run this patch. For example, if the patch updates Applications forms, you must run it on all forms servers.

Attention: Always read the readme.txt in the top level directory before running AutoPatch to apply a patch. It may contain steps that must be performed manually before and after AutoPatch is run.

The c<bugnum>.drv file contains commands for copying files. You use AutoPatch to apply this file portion of the bug patch (unless the readme.txt says otherwise). The file portion of the bug patch changes your Oracle Applications files. You may need to run this patch driver on more than one tier in a multi-tier environment.

You run the file d<bugnum>.drv using AutoPatch to apply the database portion of the bug patch. The database portion of the bug patch changes your Oracle Applications database objects. A d<bugnum>.drv file is only included if the bug patch requires changes to your Oracle Applications database objects, and if these changes can be easily automated. You must successfully run c<bugnum>.drv using AutoPatch before running d<bugnum>.drv. This driver file should only be applied from your administration server.

The g<bugnum>.drv file contains generation steps, and must be run after the file and database portions of the patch have been run. A g<bugnum>.drv file is only included if the bug patch requires new forms, reports, or message files to be generated. This driver file may need to be applied on your forms servers and/or concurrent processing servers.

The subdirectories under the top-level directory contain files that will be copied into your Oracle Applications directory structure.

WARNING: Patches must always be applied in their entirety. If you apply a patch to update your filesystem, you must also apply the corresponding database and generation portions of the patch if they are included. When updating the filesystem in a multi-tier environment, you must apply the patch on all tiers listed in the readme.txt file.

readme.txt

The readme.txt entry for a single bug lists:

- Bug number

Each bug entry starts with a line giving the bug number and product. For example:

```
oooooooooooooooooooooooooooo
o                                     o
o  BUG  fnd  691976                 o
o                                     o
oooooooooooooooooooooooooooo
```

- Bug description

A description of what the bug patch does. It may list the files included in the bug patch, and on what servers the patch is to be applied.

- Special Instructions

A description of manual steps you must perform. Most patches do not require any manual steps.

If the bug patch contains a d<bugnum>.drv, there is a reminder in this section of the readme.txt file telling you to run d<bugnum>.drv using AutoPatch after running c<bugnum>.drv using AutoPatch.

Integrated Patches

If the bug patch you are applying has prerequisite bug fixes, Oracle ships you an integrated patch containing both the specific bug patch you requested and all prerequisite bug patches. In this case, you may find multiple entries in your readme.txt—one entry for each bug included in the patch. You may also find more than one d<bugnum>.drv file. The readme.txt file tells you which d<bugnum>.drv files you should run, and the order in which to run them, if the order is important.

Running AutoPatch

The following sections explain how to run AutoPatch to apply a patch to an Oracle Applications product group.

Before Running AutoPatch

Perform the following steps when you are ready to run AutoPatch:

1. Log in as applmgr, and start a Bourne or Korn shell if you are not currently running one.

Note: Depending on your platform, AutoPatch may be run under a Bourne or Korn shell. Consult the *Installation Update* for your platform for further information.

2. Run the environment file for the Oracle Applications product group that you want to update.
3. Check that \$ORACLE_HOME is set to the proper database directory, and that \$ORACLE_SID or \$TWO_TASK identifies the correct database.

If you are updating multiple product groups in separate databases, set the variables to identify any of the databases. You can then reset the variables and run AutoPatch again if necessary to update the remaining groups.

Additional Information: [Running AutoPatch Again](#) in this chapter.

4. Ensure that the full pathname of the \$ORACLE_HOME/bin directory is defined in your PATH variable. You can add this directory to PATH by giving this command:


```
$ PATH=$PATH:$ORACLE_HOME/bin; export PATH
```
5. Backup any previously patched files that you want to save. The filenames of previously patched files end with O, such as xxxO. Copy these files to a different location if you need backups of these files. The original O file will be overwritten if AutoPatch patches the file.
6. Ensure that you have at least 50 Mb of space available in the operating system's temporary directory (typically /tmp or /usr/tmp) for use by AutoPatch. This is necessary because some programs called by AutoPatch, such as adrelink, use these default temporary directories.
7. If you are applying a patch that updates or relinks files on a concurrent processing server, shut down the concurrent managers. If you are relinking files on a forms server, have all Oracle Applications users log off before proceeding.

Attention: A patch is not completely applied until all portions of the patch have been run. Accessing Applications forms while a patch is being applied may result in sporadic errors or messages until the patch is completed.

Starting and Stopping AutoPatch

You run AutoPatch from the patch directory, which is the directory where you unloaded the AutoPatch files. Start AutoPatch with this command:

```
$ adpatch
```

You can exit AutoPatch by entering `abort` at any prompt. You can then restart AutoPatch from where the session ended or from the beginning.

WARNING: Once AutoPatch begins to update products, do not attempt to stop it without first contacting Oracle Support Services.

Additional Information: [Running AutoPatch Again](#) in this chapter.

Running AutoPatch in a Multi-Tier Environment

If you are installing Oracle Applications in a multi-tier environment, you must run AutoPatch on all relevant machines to install the necessary files.

AutoPatch Prompts

When AutoPatch prompts for information, it displays a default answer in square brackets. Here is an example:

```
Filename [adpatch.log] :
```

You accept the default value by pressing [Return]. To specify a different value, type the new value and press [Return].

AutoPatch Questions

When you start AutoPatch, it displays an introduction screen and asks you questions about the following aspects of your installation.

Applications directory

AutoPatch displays the value of the environment variable `APPL_TOP`. Press [Return] to confirm that this is the top directory for the Oracle Applications products.

If this is not the correct directory, exit AutoPatch by entering `No`. Restart AutoPatch and continue your session after you change the value of `$APPL_TOP`.

AutoPatch log file

When you start or restart AutoPatch, it prompts for a name for the AutoPatch log file. The default log file name is `adpatch.log`. The AutoPatch log file resides in `$APPL_TOP/admin/<dbname>/log`, where `<dbname>` is the value of your `ORACLE_SID` or `TWO_TASK` variable. This and other log files record the entire AutoPatch session.

Additional Information: [After Running AutoPatch](#) in this chapter.

Note: When applying a bug fix, Oracle recommends you name your log file in a manner similar to the driver file you are applying; e.g.: `d123456.log`.

ORACLE Database

AutoPatch lists the database and database directory you are set to use. It asks you to confirm that you want to apply a patch to Oracle Applications in this database with the files in this directory. Here is an example:

```
You are about to apply a patch to the installation of Oracle Applications
in your ORACLE database 'apptest'
using ORACLE executables in '/d01/app/oracle/prod/8.0.4'.
```

```
Is this the correct database [Yes] ?
```

If AutoPatch does not list the correct database and directory, answer No at the prompt to exit. Resume running AutoPatch after you change the values of `$ORACLE_SID` or `$TWO_TASK`, and `$ORACLE_HOME` to identify the correct database and directory.

Additional Information: [Verify \\$ORACLE_HOME, \\$ORACLE_SID, and \\$TWO_TASK](#) in Chapter 2.

SYSTEM and AOL user passwords

AutoPatch next asks for the server's SYSTEM password. It then connects to the database and determines the ORACLE username for the AOL schema that uniquely identifies your existing product group. AutoPatch prompts for the password of this schema, then verifies the connection.

AutoPatch obtains the rest of the information it needs directly from the database, including the installed languages, module information, and product dependencies. It then validates the password information for all Oracle Applications schemas.

Patch directory

AutoPatch next asks you to specify the directory that holds the patch driver file. The default is the directory from which you started AutoPatch. If the default is not correct, enter the full path name of the directory where you unloaded the patch files.

Patch driver file

AutoPatch next prompts for the patch driver file, which is typically the file `c<bugnum>.drv` in the patch directory. If you are using a different driver file, enter the filename at the prompt. Enter the full path name and filename if the file is not in the patch directory. AutoPatch then checks the integrity of the patch driver to ensure it is valid, and determines what bugs in the driver file need to be applied on your site.

If you are applying a release update, AutoPatch compares the release number of the patch driver file to the release number of the Oracle Applications products. If the numbers are the same, it prints the following message, which helps to prevent you from needlessly reapplying a release update:

```
On-site and patch driver versions are exactly equal
Do you want to run AutoPatch anyway [No] :
```

Start the AutoPatch update

After AutoPatch reads and validates the patch driver file, it displays a message like the following:

```
aru 12345 contains:
  Fix 1000 for product "alr" Apply=Y: "Initial setting"
```

This message lists the short name of each product that requires updating. AutoPatch then asks if you want to continue:

```
Do you want to continue with AutoPatch [Yes] ?
```

Number of parallel workers

Unless the patch driver file contains the line “compatible parallel no”, AutoPatch runs `exec` and `sql` commands in parallel mode by default. AutoPatch prompts:

```
Enter the number of parallel workers [3] :
```

After you specify the number of parallel workers, AutoPatch displays messages like the following as it begins to update the Oracle Applications products:

```
Performing version checking for driver files...
Copying driver files into installation area...
Determining valid on-site files...
Screening out files not valid for this installation...
Extracting object modules from product libraries...
Performing version checking...
Determining what executables to link...
Determining what Oracle Forms files to generate...
Determining what Oracle Reports libraries to generate...
Determining what Oracle Reports files to generate...
...
```

AutoPatch parallel mode works essentially the same as AutoInstall parallel mode: AutoPatch starts up workers; the workers run tasks as assigned by AutoPatch; the workers eventually finish; then AutoPatch starts running again, does a few more tasks, and exits. You can use the AD Controller utility (adctrl) to monitor and alter AutoPatch parallel worker status just as you would with AutoInstall.

Note: AutoPatch runs all database actions based upon phase order, which is not necessarily the order in which the commands are listed in the database patch driver.

AutoPatch Messages

You need to monitor AutoPatch as it runs to check for error messages. The following sections explain the types of messages you may see.

Informational Messages

Informational messages, such as files that were not applied, are written to an informational log file. This file resides in the same location as your AutoPatch log file, and has the same base filename, but has the extension .lgi instead of .log. For example, if your AutoPatch log file is named d123456.log, then your AutoPatch informational log file will be named d123456.lgi.

This log file contains output such as the following:

```
Will not apply POXPOPAA.rdf: Files are identical.
Patch   : /d01/appl/patch/po/reports/POXPOPAA.rdf, v110.3
On-Site: /d01/appl/110/po/11.0.28/reports/POXPOPAA.rdf, v110.3
```

Not running file 'wip patch/110/sql wipmlprb.pls' against schema 'apptest'
because the corresponding PL/SQL object in the database is up to date.
Revision in File = 110.8
Revision in Database = 110.8

These messages, which list the actions that AutoPatch will not perform, are normal. Typically these messages are produced if you run AutoPatch more than once because AutoPatch does not update files that it has updated already. You may also see these messages if you have previously applied other patches to update the products.

Impact on Customizations

If you have registered your customized files in \$APPL_TOP/admin/applcust.txt, AutoPatch will review the contents of this file before applying the contents of a patch to determine if any of your registered customized files will be replaced by the patch. If so, AutoPatch displays a message listing the customized files it will replace. For example:

Reading customized files list (if any)...

This patch replaces the following customized files:

The following files are registered as customized by modification,
and will be replaced by this patch. After this patch has been applied,
these files will contain the changes required for this bug fix,
but will no longer contain your customizations.

You may need to re-apply your customizations to these files.

Product	Subdirectory / Library	File Name
per	forms/US	PERWFDWF.fmb
per	reports	PERRPREQ.rdf

A similar message is displayed for files that are registered as customized by extension, and whose source files will be replaced by the patch.

Additional Information: Customization Standards, *Oracle Applications Developer's Guide*

Errors

If you encounter an error while running AutoPatch, check the AutoPatch log files, including worker log files, for information on the problem. Contact Oracle Support Services if you cannot determine or resolve the problem.

Additional Information: [After Running AutoPatch](#) in this chapter.

The information in the file c<bugnum>.drv may help you to finish the update successfully.

Additional Information: [Patch Driver Files](#) in this chapter.

Successful Completion Message

AutoPatch displays a message like the following once it runs to completion:

```
AutoPatch is complete.
```

```
AutoPatch may have written informational messages to the file  
/d01/appl/110/admin/apptest/log/adpatch.lgi
```

```
You should check the file  
/d01/appl/110/admin/apptest/log/adpatch.log
```

```
for errors.
```

Review the log files when AutoPatch has finished successfully.

After Running AutoPatch

Follow the instructions in this section after AutoPatch has run to completion successfully.

Check AutoPatch Log Files

After AutoPatch runs to completion, search for the words *Error*, *error*, *ERROR*, *Warning*, *warning*, and *WARNING* in the AutoPatch log files even if AutoPatch appeared to encounter no errors. Some warnings are diagnostic messages that do not indicate an error and can be ignored. Contact Oracle Support Services if you do not understand the meaning of an error message.

Additional Information: [Acceptable Errors](#) in Chapter 3

There are several AutoPatch log files, each of which records certain types of update action. If AutoPatch does not perform a certain type of action, it does not generate the log file that records that type of action. The directory \$APPL_TOP/admin/<dbname>/log, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable, may therefore contain only some of the following AutoPatch log files:

Log File	Description
adpatch.log	main AutoPatch log file (default name)
adrelink.log	for relinking
admvcode.log	for moving updated files to product directories
adlibin.log	for moving C object files into a product's C library
adlibout.log	for moving C object files out of a product's C library
adworkxxx.log	for database operations run in parallel mode
adpatch.lgi	for AutoPatch informational messages (default name)

WARNING: AutoPatch log files and restart files (adpprod.*) record the passwords to your Oracle Applications products. Restrict access to these files if the administration server or \$APPL_TOP/admin directory is not already restricted.

Additional Information: [Set Up AutoInstall Environment](#) in Chapter 2.

Review Customizations

If AutoPatch displayed a message indicating that you had registered customized files that would be replaced by the patch, you should review those files now to determine if your customizations need to be re-applied or merged.

Attention: If you had files that were registered as customized by modification, you should rename or move the backup file (your modified version before the patch was applied) immediately so it is not overwritten by a later patch.

Remove Obsolete Files

After you update all product groups successfully, you can delete the backup copies of files (files that have O appended to their filenames) from the product directory trees to recover disk space. You should check for customized files before deleting these backups. You register customized files in \$APPL_TOP/admin/applcust.txt. If these backups contain customizations, you may want to reapply your customizations before deleting the backup copies.

Additional Information: [Recovering Disk Space](#) in Appendix A.

The file applptch.txt in the \$APPL_TOP/admin/<dbname> directory contains information on which files have changed.

Additional Information: [Patch History File](#) in this chapter.

Maintain MultiLingual or Multiple Reporting Currencies Schema(s)

If you are using Multiple Language Support or Multiple Reporting Currencies functionality, you must re-run the options to maintain your MLS or MRC schema(s) in the AD Administration Utility after running any patch driver file which updates database objects. AutoPatch displays a reminder message when the patch is completed if it detects either functionality in your database. For example:

```
-----
You have installed the Multiple Reporting Currencies feature,
and AutoPatch has run at least one command that may have changed
the structure of your Oracle Applications database objects.

You should run the "Maintain Multiple Reporting Currencies schema(s)"
menu option from the "Maintain Applications Database Objects" menu
in the AD Administration Utility.

This will ensure that each of your your Multiple Reporting Currencies
schemas is correctly synchronized with the corresponding APPS schema.
-----
```

Additional Information: [AD Administration Utility \(adadmin\)](#) in this chapter.

Pin SGA Packages

If AutoPatch modified your Oracle Applications database objects, run the ADXGNPIN.sql and ADXGNPNS.sql scripts again to pin new packages and sequences in your ORACLE System Global Area.

Additional Information: [Pinning Packages and Sequences](#) in Appendix A.

Running AutoPatch Again

Make sure that you run AutoPatch as many times as necessary until it runs to completion. You can then run AutoPatch again to update other servers in your environment or another Oracle Applications product group.

Running a Session to Completion

If you aborted your AutoPatch session or it did not run to completion, restart AutoPatch with this command:

```
$ adpatch
```

AutoPatch first prompts for the name of the log file. If you specify the log file from the previous session, AutoPatch adds the message

```
****Start of AutoPatch Session****
```

to the end of that file and appends the messages from the new session as they are generated.

If you specify a new filename, AutoPatch creates a new main log file for this session. However, it does not create new versions of the other log files, such as admvcode.log or adrelink.log. It appends new messages to the existing versions of these files.

AutoPatch then asks if you want to complete the previous session. If you respond with No, AutoPatch asks you to confirm your choice and then restarts from the beginning. If you respond with Yes, AutoPatch restarts where the previous session stopped.

If you get an error when running AutoPatch indicating that the FND_INSTALL_PROCESSES table already exists, then you must determine if AutoPatch is currently being run in another session, or whether a previous patch session did not run to completion. If AutoPatch is running in another session or on another server, you must wait until that AutoPatch session completes before resuming in the current environment. If a previous patch session did not complete, you should resume applying that patch before continuing with a new one.

Attention: Applications patches must *always* be applied in their entirety.

Running AutoPatch in a Multi-Tier Environment

If you are installing Oracle Applications in a multi-tier environment, you must run AutoPatch on every machine to install the necessary files. You only need to run AutoPatch once, on the administration server, to update database objects, but you must run the file and generation portions on all machines that require those changed files. Specifically:

- you run the file portion of the patch (c<bugno>.drv) on every machine containing one or more of the files being replaced by the patch,
- you run the database portion of the patch (d<bugno>.drv) once, from the administration server only, and
- you run the generation portion of the patch on every machine containing one or more of the files to be generated.

For example, if the patch replaces forms files, you must run the file portion and the generation portion on every forms server. If the same patch also contains a database script, you must run the file and database portions on your administration server.

You may choose to apply the file and database portions of the patch on the administration server first, then run AutoPatch on every other server to apply the file and generation portions. Or you may choose to do apply the file portion of the patch on each server in turn, then the database driver on the administration server, and then the generation tasks on all other servers.

Attention: You can not run multiple sessions of AutoPatch simultaneously against a product group. You may run AutoPatch concurrently on separate machines, but only one session may be performing database or generation tasks.

Updating Another Product Group

In most cases, if you have multiple Oracle Applications product groups, you need to run AutoPatch against each product group.

Note: Multiple Sets of Books Architecture is not the same as multiple product groups. You need to run AutoPatch only once to update a Multiple Sets of Books configuration that uses the same Oracle Application Object Library tables.

However, if all of the following criteria apply, you do not need to run AutoPatch for each one of your product groups:

- The product groups have identical configurations.
Product groups with identical configurations have the exact same set of fully installed and shared products.
- The product groups share the same set of files.
All of the product groups share the same \$APPL_TOP.
- The patch you are applying does not run executable programs or SQL scripts (your patch has no enabled exec or sql actions).
If the patch runs executable programs or SQL scripts, these scripts will change the contents or structure of your product group within your database. This database change must be applied to each product group separately.

Follow these guidelines when you update another product group:

- Log out of applmgr. This clears your PATH environment variable. Then log into applmgr, execute the correct environment file, and start a new AutoPatch session.

Additional Information: [Running AutoPatch](#) in this chapter.

- Specify a different AutoPatch log file for each product group that you update.

Controlling AutoPatch Behavior

Running AutoPatch in Test Mode

AutoPatch provides a test mode in which it tells you everything it would have done in applying a patch, but doesn't actually apply the patch.

To run AutoPatch in test mode, you must include `apply=no` on the AutoPatch command line. For example:

```
$ adpatch apply=no
```

Instead of performing an action, AutoPatch indicates that it is not performing the action because `apply=no`. In general, AutoPatch lists each file it would have copied, generated, relinked, or executed. This shows you exactly what actions it would have performed.

AutoPatch test mode works the same as normal mode, with the following exceptions:

- It does not copy any files from your patch directory into your installation area.
- It does not archive any object modules into your product libraries.
- It does not relink any executables.
- It does not generate any forms, reports, PL/SQL libraries, or menu files.
- It does not run any sql or exec commands (commands that change the database).
- It does not update the Patch History File (`applptch.txt`).
- It does not update your release version in the database.

AutoPatch asks you the same initial questions in test mode as in normal mode. It performs the following actions to determine what it would have done if run in normal mode:

- Reads and validates the patch driver file.
- Reads product file driver files.
- Extracts object modules from your product libraries (so it can perform version checking on the object modules it extracts).
- Performs version checking.
- Looks in the database to determine what sql and exec commands it would have run.

You can also use AutoPatch test mode when running AutoPatch in pre-AutoInstall mode.

Running AutoPatch in Pre-AutoInstall Mode

AutoPatch allows you to apply patches after running adsetup and before running AutoInstall to install or upgrade Oracle Applications. This is useful for applying patches to AutoInstall itself, to AutoPatch, or to other installation utilities before running AutoInstall. Be sure to run the AutoInstall environment file, adsetenv, to set up the environment before running AutoPatch.

To run AutoPatch at this point in your installation or upgrade procedure, you must invoke AutoPatch with a special command line argument `preinstall=y`. This flag tells AutoPatch to run in a special mode called Pre-AutoInstall mode—a method that allows you to run AutoPatch before you run AutoInstall. For example, to invoke AutoPatch in Pre-AutoInstall mode:

```
$ adpatch preinstall=y
```

When you run AutoPatch in Pre-AutoInstall mode, it asks all of the normal AutoPatch startup questions, except those relating to the database. You should use Pre-AutoInstall mode only when running AutoPatch on your administration server.

There are several restrictions to running in Pre-AutoInstall mode:

- AutoPatch will not run SQL scripts, run `exec` commands, or generate any files, such as forms or reports, since these actions require AutoPatch to connect to a database.
- AutoPatch does not read the Oracle Applications product driver files to determine valid on-site files.
- AutoPatch only relinks AD and AOL executables.
- AutoPatch will not apply Release Updates, it will only apply individual bug patches.

Other than the above restrictions, AutoPatch runs the same in Pre-AutoInstall mode as it does normally.

Note: Because AutoPatch does not read driver files in Pre-AutoInstall mode, it copies all product files in the patch to your `$APPL_TOP` directory, even if they should not exist on the current tier. You should therefore only run AutoPatch in pre-install mode if the patch you are applying requires it.

Using the options Parameter

AutoPatch supports a generic command-line argument called `options`. The `options` argument consists of a comma-separated list of keywords instructing AutoPatch to enable or disable certain actions in a patch. If the keyword is preceded by "no", all actions in the patch corresponding to that keyword are disabled.

For example, to disable the generation of forms and reports files, you can invoke AutoPatch with the following command:

```
$ adpatch options=nogenrep,nogenform
```

Here are the valid keywords and what they do:

Option Name	What It Toggles
copy	copy commands in the driver file
exec	exec commands in the driver file
forcecopy	forcecopy commands in the driver file
genform	genform commands in the driver file
genfppl	genfppl commands in the driver file
genmenu	genmenu commands in the driver file
genrep	genrep commands in the driver file
genrppl	genrppl commands in the driver file
libin	libin commands in the driver file
libout	libout commands in the driver file
link	link commands in the driver file
sql	sql commands in the driver file
validate	validating the password for each ORACLE schema

Patch Merge Utility

The `admrgpch` program merges multiple AutoPatch-compatible patches into a single integrated patch. The `admrgpch` utility is a Bourne shell script located in the `$AD_TOP/bin` directory.

To merge two or more patches into a single integrated patch, run the `admrgpch` utility with the following arguments:

```
$ admrgpch <source directory> <destination directory>
```

<source directory> is the directory in which the patches to merge have been unloaded. The patches to merge must be formatted exactly as described in the Patch Format section earlier.

<destination directory> is the directory in which admrgpch will create the merged patch.

The admrgpch utility reads the c<bugnum>.drv, readme.txt, and d<bugnum>.drv for each bug patch in the source directory and merges them together to create c<bugnum>.drv, readme.txt, and dbmerged.drv files in the destination directory. It also merges the set of files contained in the individual bug patches under the source directory according to file revision, and copies them to the destination directory. If a file is contained in more than one source patch, only the highest revision of the file is copied over to the destination directory.

Actions in the merged c<bugnum>.drv are grouped by product, with one bug per product. Bugs in dbmerged.drv are ordered by bug number. Comments in the merged readme.txt are also ordered by bug number.

After admrgpch runs, you should check the admrgpch.log file for errors. The file is located in \$APPL_TOP/admin/log. If you do not find any errors, you can read the readme.txt file in the destination directory for instructions on applying the merged patch using AutoPatch.

Patch Driver Files

The patch driver file lists the update actions that AutoPatch performs on each Oracle Applications product. Reviewing the AutoPatch driver file may help you to resolve any problems with the update. However, do not attempt to use the information in this file to complete the update without first contacting Oracle Support Services.

The following sample shows part of a c<bugnum>.drv file and part of a g<bugnum>.drv file. Your driver files will have the same form as the following example, but will contain different information:

❑ Excerpt from c11235.drv:

```
begin aru c11235
  character set us7ascii
  compatible release 11.0.0
  compatible parallel yes
  begin bug po 10123
    ...
  end bug po 10123
  begin bug po 11235
    begin actions
      copy po reports POXPOPAA.rdf 110.3
      copy po forms/US POXSCVAR.fmb 110.6
      copy po resource POXPOPOS.pll 110.16
    end actions
  end bug po 11235
end aru c11235
```

❑ Excerpt from g11235.drv:

```
begin aru g11235
  character set us7ascii
  compatible release 11.0.0
  compatible parallel yes
  begin bug po 10123
    ...
  end bug po 10123
  begin bug po 11235
    begin actions
      genrep po reports POXPOPAA.rdf
      genform po forms/US POXSCVAR.fmb
      genfp11 po resource POXPOPOS.pll
    end actions
  end bug po 11235
end aru g11235
```

The “begin aru 11235” and “end aru 11235” statements in both files indicate that this patch corresponds to bug patch 11235. If you are applying a release update using AutoPatch, you will see “begin release xxxx” and “end release xxxx” instead. All actions for this patch must be contained within the “begin aru” and “end aru” statements. AutoPatch only supports one ARU or release update per patch driver file.

The “character set us7ascii” statement indicates that the files in this patch have character set us7ascii. If your on-site character set is different, AutoPatch converts text files to your on-site character set as it copies the file from the patch area to your installation area. If the character set statement is missing from your patch driver file, the patch character set defaults to us7ascii.

The “compatible release 11.0.0” statement indicates that this patch is compatible with any Oracle Applications 11.0.x release. For example, AutoPatch will not apply this patch to any Release 10.7 or 11.1 Applications directory. If the patch driver file does not contain a “compatible release” statement, AutoPatch tries to apply the patch against your release of Oracle Applications without verifying that your on-site release is compatible with the patch. Note that AutoPatch uses a similar mechanism to determine whether to apply a release update. This separate feature is always active when you apply a release update.

The “compatible parallel yes” statement indicates that AutoPatch should always run exec and sql commands in parallel mode, even if you choose only one parallel worker. This is the default when running exec and sql commands.

All statements between “begin bug po 11235” and “end bug po 11235” are part of the fix for bug 11235 of Oracle Purchasing.

The lines between “begin actions” and “end actions” list the actions that AutoPatch performs. Each line lists the action, the product, the product subdirectory, the filename and file version, and any additional arguments. For example, in the c11235.drv file, the first action line tells AutoPatch to compare POXPOPAA.rdf version 110.3, from the patch directory, with the version on-site in \$PO_TOP/reports. If the on-site version is older or missing, AutoPatch copies the newer file to \$PO_TOP/reports. Similarly, the second line instructs AutoPatch to compare POXSCVAR.fmb version 110.6 with the version on-site in \$AU_TOP/forms/US, where all Applications forms are located. If the on-site version is older or missing, AutoPatch copies the file to \$AU_TOP/forms/US. Note that AutoPatch automatically knows when certain file types, such as forms and PL/SQL libraries, are to be copied directly to a centralized location, like \$AU_TOP.

The following are the actions that AutoPatch may perform:

- | | |
|------|--|
| copy | Compare version of file in patch with on-site and copy from patch directory to target subdirectory if target is older or missing. The file admvcode.log records these actions. |
| exec | Run executable program. The main AutoPatch log file and worker log files record these actions. |

forcecopy	Copy object file into target subdirectory but do not check version number against existing file. The file admvcode.log records these actions.
genform	Generate Oracle Forms form.
genrppl	Generate the specified Oracle Reports PL/SQL library file.
genrep	Generate Oracle Reports report.
genfppl	Generate Oracle Forms PL/SQL library file.
genmenu	Generate Oracle Forms menu file.
libin	Copy file into product library. The file adlibin.log records these actions.
libout	Extract file from product library. The file adlibout.log records these actions.
link	Relink executable with Oracle8 Server. The file adrelink.log records these actions.
sql	Run SQL script. The main AutoPatch log file and worker log files record these actions.

Patch History File

The first time it runs, AutoPatch generates a file named applptch.txt in the \$APPL_TOP/admin/<dbname> directory, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable. It updates the header and appends information to this file each subsequent time it runs. This file records the following information pertaining to your AutoPatch sessions:

1. Bugs not applied and why.
2. Bugs applied. For each bug applied the following is listed:
 - All actions that were executed for the patch.
 - All actions that were not executed for the patch.

Here is an example of the applptch.txt header, which lists the last time AutoPatch was run:

```
%% applptch file format 11.0.A
CURRENT_RELEASE: 11.0.28
RELEASE_CHANGE_DATE: Thu Mar 12 1998 16:45:37
FILE_CHANGE_DATE: Fri Mar 13 1998 13:10:46
WHY_FILE_CHANGED: ARU 11235
```

Here is an example of an AutoPatch session record after the first of the two sample driver files above is run:

```
#
# Run of AutoPatch, Thu Mar 12 1998 18:44:09
# ORACLE_SID/TWO_TASK: apptest
# ORACLE_HOME: /oracle/db/8.0.4
begin aru c11235
  charset us7ascii
# The following bugs were not applied because:
# No active actions
# bug po 10123
  begin bug po 11235
    begin actions
#
# AutoPatch executed the following actions:
#
      copy      po  forms/US  POXSCVAR.fmb  110.6
      copy      po  resource  POXPOPOS.pll  110.16
#
# AutoPatch did not execute the following actions:
#
      copy      po  reports   POXPOPAA.rdf  110.3
    end actions
  end bug po 11235
end aru c11235
#
# End run of AutoPatch, Thu Mar 12 1998 18:44:10
#
```

Unload Utility (adunload)

This section explains how to use the unload utility to unload Oracle Applications product files from CD-ROM. The unload utility is provided in case you need to unload a small number of files after you have installed or upgraded your system.

Attention: You should always use AutoInstall to unload product files, unless you are only retrieving individual files. AutoInstall unloads only the necessary files depending on configuration, license status, and what files already exist from patches applied in pre-AutoInstall mode. In contrast, the adunload utility does not copy files to alternate destinations and may overwrite newer files.

Additional Information: [Installing Product Files \(File Tasks\)](#) in Chapter 3.

Log File

If an error occurred or if you are unsure of messages returned by the unload utility, review the file adunload.log. If adunload was run from AutoInstall, the file will be in \$APPL_TOP/admin/<dbname>/log, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable. If you run adunload from the command line, the file will be in \$APPL_TOP/admin/log. As it runs, the unload utility appends information to the end of this file.

You can delete the adunload.log file if it is large and contains no necessary information. In this case, the unload utility creates a new adunload.log file the next time it runs.

Running the Unload Utility

Do the following before you run the unload utility:

- ❑ Log in as applmgr.
- ❑ Verify that the environment variable \$APPL_TOP is set to the Applications top directory.

Additional Information: [Verify \\$ORACLE_HOME, \\$ORACLE_SID, and \\$TWO_TASK](#) in Chapter 2.

- ❑ If you have not yet run AutoInstall to install or upgrade your products, unload the AutoInstall files and run the adsetenv script to add the directory that holds the unload utility (\$AD_TOP/bin) to your PATH.

Additional Information: [Unload AutoInstall Files](#) and [Set Up AutoInstall Environment](#) in Chapter 2.

Syntax

Run the unload utility with the following command syntax:

```
$ adunload product=<product abbreviation> \
directory=<directory name> \
filename=<filename> \
[appltape=<tape description file>] \
[character set=<character set name>] \
[industry=<industry name>] \
[topfile=<filename>]
```

or

```
$ adunload filelist=<filename> \
[appltape=<tape description file>] \
[character set=<character set name>] \
[industry=<industry name>] \
[topfile=<filename>]
```

You must specify *either* the product, directory, and filename parameters, *or* the filelist parameter. All other parameters are optional.

The utility prompts for the drive name and instructs you to mount the appropriate CD after you give an unload command.

Type the following to see online instructions about the unload utility:

```
$ adunload
```

Parameters

The following entries describe the unload utility parameters. You can specify parameters in any order. There are no default values for parameters.

product Use the product's short name, as in product=gl for Oracle General Ledger.

Additional Information: Table 2-1: Release 11 Products in *Oracle Applications Concepts*.

directory Include this parameter to unload only the files in a certain directory. Use the directory name, as in `directory=forms`.

If you are unloading translated reports files, for example, you need to specify the NLS language code subdirectory, as in `directory=reports/F` for European French reports. For forms, specify the Oracle Applications language short name subdirectory, as in `directory=forms/F` for European French forms. You also need to use the `apptape` parameter.

Additional Information: Language Directory Structure in *Oracle Applications Concepts*, [Language Codes and Character Sets](#) in Appendix A.

filename Use this parameter with a filename as its value to unload a single file. You must also use the `directory` parameter.

filelist Use this parameter to specify a file that contains a list of the product files to unload. Each line in this filelist file contains the name of one product file to unload, in the format:

```
<product> <directory> <filename>
```

The product, directory, and filename on each line are separated by one or more spaces or tabs. For example:

```
wip      forms/US      WIPMRQAW.fmb
wip      forms/US      WIPSTMSC.fmb
```

apptape Use this parameter to unload a CD that contains language or localization files. Specify the file in the `$APPL_TOP/admin` directory that contains information about the language or localization CD, such as `alotape.txt` for localizations and `awe8tape.txt` for French.

character set Use this parameter to specify your system character set if the character set of the files on the CD does not match the system character set. This parameter takes as its value the name of a supported character set, as in `character set=we8iso8859p1`.

Additional Information: Character Set of Shipped Files, *Oracle Applications Installation Update* for your platform

industry Use the parameter `industry=GV` to unload files for Oracle Public Sector Financials products.

topfile The utility unloads product files under the \$APPL_TOP directory unless you specify different top directories with the topfile parameter. The parameter value is a text file in your working directory. This file lists any number of products by short name. An absolute pathname of the directory that will hold the product files must follow the product short name.

Examples

The following table contains examples of the way you can use adunload:

To...	Do this...
Unload Files	<p>This command unloads the Oracle General Ledger file GLXSTUOM.fmb, which is in the forms directory:</p> <pre>\$ adunload product=gl directory=forms/US \ filename=GLXSTUOM.fmb</pre>
Unload Files and Change Character Set	<p>This command unloads the Oracle Application Object Library file FNDNLINS.sql, in the sql directory, and changes the character set to we8hp:</p> <pre>\$ adunload product=fnd directory=sql \ filename=FNDNLINS.sql characteraset=we8hp</pre>
Unload Oracle Public Sector Financials Files	<p>This command unloads files listed in polist.txt for Oracle Public Sector Payables:</p> <pre>\$ adunload filelist=polist.txt industry=GV</pre> <p>The polist.txt file would contain the following lines:</p> <pre>ap reports APXINUPD.rdf ap reports APXVDTIN.rdf ap reports APYRLIVR.rdf</pre> <p>Use this type of command if you have licensed only some of the Oracle Public Sector Financials products.</p>
Unload Product Files into Directories Other than \$APPL_TOP	<p>This command performs the same action as the previous example, except the files are unloaded to a different location:</p> <pre>\$ adunload filelist=polist.txt topfile=potop.txt</pre> <p>The potop.txt file would contain the following line:</p> <pre>po /d03/appl/110</pre> <p>This causes Oracle Purchasing files to be unloaded to /d03/appl/110 whenever adunload is invoked with topfile=potop.txt. If a product top directory (such as /d03/appl/110/po/11.0.28) does not exist under the top directory already, the utility creates the directory.</p>

Relink Utility (adrelink)

The adrelink utility lets you relink Oracle Applications executable programs with the Oracle8 Server product libraries.

Attention: In Release 11, you should only use adrelink to relink AD executables. For all other relinking tasks, use the AD Administration utility.

Additional Information: [AD Administration Utility \(adadmin\)](#) in this chapter.

Log File

If an error occurred during relinking or if you are unsure that the relinking completed successfully, review the file adrelink.log. If adrelink was run from AutoInstall or the AD Administration Utility, the file will be in \$APPL_TOP/admin/<dbname>/log, where <dbname> is the value of your ORACLE_SID or TWO_TASK variable. If you run adrelink from the command line, the file will be in \$APPL_TOP/admin/log. As it runs, the adrelink utility appends information about the latest relink action to the end of this file.

You can delete the adrelink.log file if it is large and contains no necessary information. In this case, adrelink creates a new adrelink.log file the next time it runs.

Running adrelink

Perform the following steps before you use adrelink:

- ❑ Verify that you have C development tools compatible with Berkeley Version 1.15 or higher of make. The UNIX online manual pages may list the version number of make. Note that a C compiler is not needed unless you are doing custom development in C.
- ❑ Log in as applmgr and run the appropriate environment file.
- ❑ If you are relinking files on a concurrent processing server, shut down the concurrent managers. If you are relinking files on a forms server, have all Oracle Applications users log off before proceeding. After relinking, restart the concurrent managers and allow Oracle Applications users to log on.

Additional Information: Administer Concurrent Managers in *Oracle Applications System Administrator's Guide*; [Concurrent Managers](#) in Appendix A.

Syntax

To relink one or a few executable programs, run adrelink with one of these two commands:

```
$ adrelink force={y | n} [<optional args>] <targets>
```

or

```
$ adrelink force={y | n} [<optional args>] filelist=<file>
```

where

```
<targets>={<product module> | [<product module>]...}
```

and

```
<product module>="<product> <executable to link>"
```

Valid <optional args> are:

envfile=adsetenv	Used only by the adsetup script before an environment file is created.
link_debug={y n}	Link executables with debugging information intact. The default is "n".
backup_mode={none all file}	Indicate which executables to back up when linking with force=y. Options are: <i>none</i> (do not back up any executables) <i>all</i> (back up all executables) <i>file</i> (back up files in adlinkbk.txt) The default is "file".

You can type the following to see online instructions about adrelink syntax:

```
$ adrelink
```

You can type the following to see examples of running adrelink:

```
$ adrelink examples
```

Parameters

The following entries describe adrelink parameters. These parameters are required. There are no default values for these parameters.

force (Required) The parameter force=n makes the utility relink the executable program only if the dependent libraries or object files are more recent than the current executable program. With force=y, the utility relinks regardless of the status of the libraries or object files.

The **backup_mode** argument is only relevant when force=y. If force=n, adrelink neither backs up nor removes executables.

product module (Required) This parameter identifies the executable programs to relink. Enclose a product abbreviation and program name in quotation marks, as in “gl GLPPOS”.

Additional Information: Table 2-1: Release 11 Products in *Oracle Applications Concepts*.

Examples

The following table contains examples of the way you can use adrelink:

To...	Do this...
Relink a Single Executable Program	<p>This command relinks a specific executable program:</p> <pre>\$ adrelink force=y "<product> <module name>"</pre> <p>To relink the Oracle Application Object Library program f45webm:</p> <pre>\$ adrelink force=y "fnd f45webm"</pre>
Relink Selected Executable Programs for Multiple Products	<p>This command relinks the Oracle General Ledger module GLPPOS and the Oracle Payables module APXRTB:</p> <pre>\$ adrelink force=y "gl GLPPOS" "ap APXRTB"</pre>

Relinking with Debug

In some cases, you may want to link an executable with debugging information intact, so that it contains the maximum amount of debugging information available. Usually, this is not very useful, since the maximum amount of debugging information available is just the name of each routine. However, if you receive some object modules that were compiled in debug mode, linking in debug mode may be very useful.

To link an executable in debug mode with adrelink, you must specify the argument `link_debug=y` on the adrelink command line after the force argument and before any other arguments.

The following example shows the command to relink the Oracle General Ledger module GLPPOS with debug:

```
$ adrelink force=y link_debug=y "gl GLPPOS"
```

Target File

To relink a number of executable programs at once, list the executable programs in a file and relink them with a single adrelink command.

Create a Target File

Use the editor of your choice to create an ASCII text file in your current working directory. Each line should have this format:

```
<product>  bin  <module name>
```

Separate the entries with tabs. The following table explains the parameters:

If the parameter is...	Do this...
product	Use the product's short name, such as fnd or gl.
module name	Use the name of the executable program, such as GLPPOS. Enter <i>all</i> to relink all executable programs for the product.

For example, you might create a file that lists several executable programs for different products:

```
gl      bin      GLPPOS
gl      bin      GLCCON
fnd     bin      f45webm
fnd     bin      FNDLIBR
```

Run adrelink

Run the adrelink utility with the following command:

```
$ adrelink force=<y|n> filelist=<filename>
```

Here is an example:

```
$ adrelink force=y filelist=/home/applmgr/exec.lst
```

Backup Options for adrelink

The adrelink utility provides three basic options for backing up your existing executables when force=y: backup_mode=none, backup_mode=all, and backup_mode=file (the default). When force=n, adrelink simply relinks out-of-date executables, and does not back them up or remove them prior to relinking.

When adrelink runs with force=y and backup_mode=none, it removes existing executables prior to relinking them. When adrelink runs with force=y and backup_mode=all, it renames each existing executable prior to relinking the executable.

When adrelink runs with force=y and backup_mode=file (the default), it does whatever the instructions in \$APPL_TOP/admin/adlinkbk.txt tell it to do. As shipped, adlinkbk.txt tells adrelink to back up a specific set of executables prior to relinking them, and to remove all other executables. It is possible to configure adlinkbk.txt to tell adrelink to back up all executables for a specific product or set of products, to back up all executables, or not to back up any executables. Comments inside adlinkbk.txt explain how to change the set of executables that adrelink backs up.

Because AD programs call adrelink with force=y in most cases, and backup_mode=file is the default, the set of executables listed in adlinkbk.txt will, by default, be backed up every time you relink executables through AutoPatch or the AD Administration utility.

When adrelink backs up an executable prior to relinking it, it renames the existing executable. If there is already a backup copy of the executable, adrelink first renames the backup copy, then renames the executable. For example, if you are relinking FNDLIBR, adrelink will rename it to FNDLIBR.sav prior to relinking. If both FNDLIBR and FNDLIBR.sav exist, adrelink first renames FNDLIBR.sav to FNDLIBR.<pid> (where <pid> is the process ID of the current adrelink session), then renames the existing FNDLIBR to FNDLIBR.sav.

This backup strategy provides for a virtually unlimited number of backup copies for executables that adrelink backs up. You should therefore make sure you have disk space available for backup executables, and you should purge old backup executables periodically.

Default adlinkbk.txt File

Many of the critical Applications programs are listed in a default adlinkbk.txt file located in the \$APPL_TOP/admin directory. The files listed in this file will be backed up unless you modify the file.

Character Set Conversion Utility (adncnv)

This section explains how to convert the character set of unloaded files. You may need to do this for any text files shipped by Applications, including SQL*Plus scripts, PL/SQL scripts, loader files, driver files, ODF files, header files, and HTML files. If you have not unloaded files already, convert the character set during the unload. If you specify the character set argument to adunload, it automatically calls adncnv to convert the character set.

Additional Information: [Unload Utility \(adunload\)](#) in this chapter.

Syntax

You can convert one file at a time with this command:

```
$ adncnv <source file> <source char set> <destination file> <dest char set>
```

Parameters

All parameters are required. The parameters and their meanings are:

source file	Path and filename of the file to convert
source char set	Current character set
destination file	Path and filename for converted file
dest char set	New character set for converted file

The path and filename for the source file and the destination file can be the same as long as the source file's directory and the \$APPLTMP directory are on the same file system.

Suggestion: If you cannot convert to the same filename, convert to a different filename or change \$APPLTMP to a directory on the same file system as the source file directory.

Example

This example shows how to convert the file afcmstat.sql from the character set we8dec to the character set we8hp:

```
$ adncnv $FND_TOP/sql/afcmstat.sql we8dec $FND_TOP/sql/afcmstat.sql we8hp
```

The appltape.txt file lists the character set of the files on your CD. Other files, such as aloctape.txt and awe8tape.txt, will list the character set for your localizations or language CDs.

ODF Comparison Utility (adodfcmp)

This section explains how to run the ODF comparison utility. You may need this information if you contact Oracle Support Services about problems with objects, grants, or synonyms in your Oracle Applications database.

Attention: You may not use the ODF comparison utility for custom development.

Each Oracle Applications product is composed of building blocks. For example, Journal Entry is one building block of Oracle General Ledger. Each building block has an object description file (ODF) that describes its tables, views, indexes, sequences, and privilege sets. Privilege sets are grants that other schemas needed in earlier releases, before the introduction of the APPS schema.

The ODF comparison utility compares a building block to its description file. A log file records any missing, extra, or incorrectly defined objects in the database. Here is an example of a log file entry:

The database is missing the sequence GL_JE_CATEGORIES_S.
Create it with the statement:

```
CREATE SEQUENCE GL_JE_CATEGORIES_S MINVALUE 1 MAXVALUE
2147483647 INCREMENT BY 1 START WITH 1 CACHE 20 NOCYCLE
ORDER
```

You can set the changedb parameter to have the utility create missing objects, grants, and synonyms automatically. You will then see the following line in the log file, after the entry above:

Statement Executed.

Running the ODF Comparison Utility

To run the ODF comparison utility, move to the directory that contains the appropriate object description file:

```
$ cd $<PROD>_TOP/admin/odf
```

To run the utility on a building block for Oracle General Ledger, for example, move to this directory:

```
$ cd $GL_TOP/admin/odf
```

Attention: If you have applied any database patches, the ODF files may have been superseded with newer versions. If newer ODF files exist, these will be located in the \$<PROD>_TOP/patch/110/odf directory.

Syntax

Run the utility by entering the following command string:

```
$ adodfcmp <parameter>=<value> [<parameter>=<value> . . .]
```

You can see instructions about the ODF comparison utility syntax by typing the following:

```
$ adodfcmp
```

Parameters

The following entries explain the ODF comparison utility parameters. You can specify parameters in any order on the command line. You must specify values for mode, touser, priv_schema, odffile, and userid. The default value applies if you do not specify an optional parameter.

Parameter Name	Default Value	Description
mode (Required)	(no default)	Determines the types of database objects to compare against the description file. The following table lists the modes and the corresponding comparisons the ODF utility will make in Release 11. For example, use changedb=yes and mode=tables to create tables, indexes, and grants. Note that the comparisons ODF makes in Release 11 differ from Release 10.6.

Table 5–2 11.0 ODF Modes and Comparisons

Mode	Compare tables?	Compare indexes?	Compare sequences?	Compare views?	Compare grants?
baseonly	Yes	Yes	Yes	No	Yes
tables	Yes	Yes	No	No	Yes
indexes	No	Yes	No	No	Yes
noindexes	Yes	No	No	No	Yes
sequences	No	No	Yes	No	Yes
views	No	No	No	Yes	No
grants	No	No	No	No	Yes

Parameter Name	Default Value	Description
rel106mode (Optional)	No	Run in Release 10.6.0-compatible mode. Use rel106mode=Yes only if there is an error during an upgrade to Release 11 from Release 10.4 or 10.5. (That is, use this parameter when the Context column in the adctrl Show Worker Status screen begins with “Upgrading....”) The default mode, rel106mode=No, should be used in all other cases.This parameter affects the behavior of the mode parameter.

Table 5–3 10.6 ODF Modes and Comparisons

Mode	Compare tables?	Compare indexes?	Compare sequences?	Compare views?	Compare grants?
baseonly	Yes	Yes	Yes	No	No
tables	Yes	Yes	No	No	No
indexes	No	Yes	No	No	No
noindexes	Yes	No	Yes	Yes	No
sequences	No	No	Yes	No	No
views	No	No	No	Yes	No
grants	No	No	No	No	Yes

Parameter Name	Default Value	Description
touser (required)	(no default)	<p>This parameter specifies the ORACLE username and password of the Oracle Applications product to grant to. In Release 11, this is usually the list of APPS schemas.</p> <p>Attention: The touser parameter is required whenever rel106mode=No (the default), or when both rel106mode=Yes and mode=grants. When rel106mode=Yes and mode is not set to grants, the touser parameter is not required.</p> <p>The touser parameter is a list of comma-separated pairs of usernames and passwords, with no spaces between them. You can supply just one username/password pair. You only need to provide multiple APPS schema usernames and passwords if you have a Multiple Sets of Books installation.</p>
priv_schema (Required)	(no default)	Specify the schema that has DBA privileges, such as SYSTEM/MANAGER.
odffile (Required)	(no default)	Specify the name of the object description file (extension .odf) to compare against the database. Contact Oracle Support Services to find out which files correspond to which database building blocks.
userid (Required)	(no default)	For this parameter, supply the ORACLE username and password for the Oracle Applications product. For example, use userid=GL/GL when comparing a building block of Oracle General Ledger. If mode=views, you should use the APPS schema username/password; e.g.: userid=APPS/APPS.

Parameter Name	Default Value	Description
changedb (Optional)	No	<p>Set changedb=Yes to change database objects to match the definitions in their object description files. This does not affect customizations made by extension because the utility does not delete objects that are not defined in the description file.</p> <p>Attention: The adodfcmp utility creates views using the FORCE option if rel106mode=No.</p> <p>Additional Information: CREATE VIEW command in <i>Oracle8 Server SQL Reference Manual</i>.</p>
alternext (Optional)	No	This parameter specifies whether to alter the next extent (No, Yes, Force). Yes changes the next extent size of a table if current next extent size is less than that specified in the odf file. Force changes the next extent size regardless of the current value.
defer (Optional)	(no default)	Provide for this parameter a list of indexes to change later. This parameter is used to delay any index creation when mode=tables. You must run adodfcmp again later with mode=indexes to create the deferred indexes.
logfile (Optional)	adodfcmp.log	This parameter specifies the name of a log file that records the comparison results. If you reuse a log file, the utility appends information to the file instead of overwriting the file. The log file resides in the directory where you start the utility if an explicit directory path is not specified.
sizingfactor (Optional)	100	<p>This parameter sets the sizing factor at which the utility creates missing database objects. The utility will create missing database objects if changedb=Yes.</p> <p>Additional Information: Sizing Factor in <i>Oracle Applications Concepts</i>, Chapter 3.</p>
tspac (Optional)	SYSTEM	This parameter sets the tablespace for tables created by the utility.
indextspace (Optional)	SYSTEM	This parameter sets the tablespace for indexes created by the utility.
listextra (Optional)	No	Set listextra=Yes to report extra objects.
listmatch (Optional)	Yes	Specify listmatch=No to prevent the utility from reporting objects that match their descriptions.
listmissing (Optional)	Yes	Specify listmissing=No to make the utility ignore missing objects.
batchsize (Optional)	1000	This parameter determines the number of rows to update at a time when populating new columns.

Parameter Name	Default Value	Description
oldviews (Optional)	replace	<p>This parameter lets you rename, drop, or replace views that do not match their descriptions. Set oldviews=rename to change the names of non-matching views to AD11_<view name> and create new views from the description file.</p> <p>Choose oldviews=drop to delete non-matching views and recreate them according to the view descriptions. The parameter oldviews=replace works the same as oldviews=drop, except the utility preserves existing grants and synonyms when recreating the views.</p>

Examples

Note the following examples:

Compare Building Block and Create Missing Objects Use a command like this to compare and create database objects:

```
$ adodfcmp odffile=<filename> userid=<username>/<password> \
changedb=Yes tspace=<tablespace> indextspace=<tablespace> \
priv_schema=<DBA schema username>/<DBA schema password> \
mode=<mode name> touser=<APPS schema name>/<APPS schema password> \
[,<second APPS schema name>/<second APPS schema password>,...]
```

For example, when rel106mode=No (which is the default):

```
$ adodfcmp odffile=glje.odf userid=GL/GL \
changedb=Yes tspace=GLD indextspace=GLX \
priv_schema=SYSTEM/MANAGER mode=baseonly touser=APPS/APPS
```

Below is the same example shown with rel106mode=Yes. As noted earlier, use this only when upgrading to Release 11. The touser parameter is not required because the mode is not grants.

```
$ adodfcmp odffile=glje.odf userid=GL/GL \
changedb=Yes tspace=GLD indextspace=GLX \
priv_schema=SYSTEM/MANAGER mode=baseonly \
rel106mode=Yes
```

Create Grants and Synonyms Use a command like this to create missing grants and synonyms when rel106mode=No (the default):

AD File Identification Utility (adident)

```
$ adodfcmp odffile=<filename> userid=<username>/<password> \  
changedb=Yes mode=grants \  
priv_schema=<DBA schema username>/<DBA schema password> \  
touser=<APPS schema name>/<APPS schema password> \  
[,<second APPS schema name>/<second APPS schema password>,...]
```

For example, the following command creates grants from Oracle General Ledger (GL) to the APPS and APPS2 schemas and creates synonyms in APPS and APPS2. Note the usernames and passwords are the list of APPS schemas.

```
$ adodfcmp odffile=glje.odf userid=GL/GL \  
changedb=Yes mode=grants \  
touser=APPS/APPS,APPS2/APPS \  
priv_schema=SYSTEM/MANAGER
```

Use a command like this to create grants and synonyms with rel106mode=yes:

```
$ adodfcmp odffile=<filename> userid=<username>/<password> \  
changedb=Yes mode=grants \  
touser=<ORACLE username>/<ORACLE password> \  
priv_schema=<DBA schema username>/<DBA schema password> \  
rel106mode=Yes
```

For example, the following command creates grants from Oracle General Ledger (GL) to Oracle Purchasing (PO) and creates synonyms in PO. Note the usernames and passwords are to the product schemas. This command line should not be used after upgrading to Release 11.

```
$ adodfcmp odffile=glje.odf userid=GL/GL \  
changedb=Yes mode=grants \  
touser=PO/PO priv_schema=SYSTEM/MANAGER \  
rel106mode=Yes
```

AD File Identification Utility (adident)

The adident program can be used to identify the version of one or more Oracle Applications files. This information is used by AutoPatch to determine whether a file in a patch is newer than the on-site version. The program is also useful for collecting information about your site when contacting Oracle Support Services.

Use the following command to run the program:

```
$ adident Header <file 1> [ <file 2> <file 3> ... ]
```

The <file *n*> arguments should be the name of any Applications text file, binary object file (extension .o), library file (extension .a), or executable program. You may provide any number of filenames as arguments. When you run give adident the name of a library file or executable, it lists all of the files that comprise the library or executable and their respective versions.

For example:

```
$ adident Header $FND_TOP/lib/wfload.o $FND_TOP/lib/libfnd.a
wfload.o:
  wfload.oc          110.5.1100.3
libfnd.a:
  fdacon.lc          110.0
  fdatat.lc          110.0
  fdastr.lc          110.0
  fdaupd.lc          110.0.1100.2
  fdahmi.lc          110.0.1100.2
  fdacv.lc           110.0
  fdfutl.lc          110.4
  ...
```

AD Configuration Utility (adutconf.sql)

This utility is a SQL script, adutconf.sql, that reports standard information about the installed configuration of Oracle Applications. This script generates a file called adutconf.lst that contains the following:

- information about the Product Group
- whether Multi-Org is installed
- whether Multi-Lingual is installed
- information about all installed products, including shared and dependent products
- information on all registered schemas
- the base language and other installed language

Use the following command to run this script:

```
$ cd $APPL_TOP/admin
$ sqlplus <APPS schema username>/<APPS schema password> @adutconf
```

You may need this information for debugging, when contacting Oracle Support Services, or to document the status of your installation.

AD Configuration Utility (adutconf.sql)

System Reference Material

This appendix contains Oracle Applications reference material that is specific to the operating system. It includes these sections:

- [Install Multiple Reporting Currencies](#)
- [Integrating Custom Objects and Schemas](#)
- [Concurrent Managers](#)
- [Printing](#)
- [Analyzing the Database](#)
- [Pinning Packages and Sequences](#)
- [Changing Oracle Applications Passwords](#)
- [Migrating the Oracle Applications Database](#)
- [Migrating the Oracle Applications Filesystem](#)
- [Recovering Disk Space](#)
- [Product Directory Structure and File Types](#)
- [Environment Variables](#)
- [Territory Names](#)
- [Language Codes and Character Sets](#)
- [Reserved Words](#)

Install Multiple Reporting Currencies

Installing Multiple Reporting Currencies is optional. This section explains the steps in the following checklist. Follow these steps to install MRC.

- ☐ 1: Log off Oracle Applications and shut down concurrent managers
- ☐ 2: Set NLS_LANG
- ☐ 3: Set rollback segments
- ☐ 5: Verify tablespace requirements
- ☐ 6: Run adadmin to validate APPS schemas
- ☐ 7: Run adadmin to compile APPS schemas
- ☐ 8: Run adadmin to convert to multiple reporting currencies
- ☐ 9: Review the adadmin.log file and worker log files for errors
- ☐ 10: Verify number of PACKAGE and PACKAGE BODY object types
- ☐ 11: Verify number of synonyms and views
- ☐ 12: Verify that there are no database triggers in the APPS_MRC schema
- ☐ 13: Verify additional information about the APPS_MRC schema
- ☐ 14: Verify the MRC data group
- ☐ 15: Create FastFormula packages in the APPS_MRC schema

1. Log off Oracle Applications and shut down concurrent managers

Ensure that all users are logged off Oracle Applications and that all concurrent managers are shut down.

2. Set NLS_LANG

Ensure that your NLS_LANG variable is set to AMERICAN_AMERICA.<codeset> before you run adadmin to convert to MRC. After converting to MRC, you can change your NLS_LANG setting.

3. Set rollback segments

Set up rollback segments so that each rollback segment has a minimum extent size of 1 Megabyte and a minimum of 60 extents. You need to set up a minimum of two rollback segments.

4. Apply the MRC patch

Check the *Oracle MetaLink* web site to be sure you have the latest version of the patch.

Additional Information: [Getting Help](#) in the [Preface](#)

5. Verify tablespace requirements

If you have a single product installation group or Multiple Organization installation, ensure you have at least 600 Megabytes of free space in your system tablespace. If you have a multiple product installation group database, ensure that you have at least 600 Megabytes of free space in your SYSTEM tablespace for each set of books in the multiple product installation group architecture. You may need additional free SYSTEM tablespace, if you have custom objects installed or if you have third party applications products installed in the same instance.

Note: For your reporting sets of books in General Ledger, ensure you have the same amount of free space reserved for each reporting set of books as is used by your primary set of books. For any of the subledger products you plan to use with MRC, ensure you have approximately 50 percent additional free space reserved for each reporting set of books. For example, if you will be using Payables with MRC and are planning to define one reporting set of books for Payables, you need approximately 50 percent more space than what Payables uses without MRC.

6. Run adadmin to validate APPS schemas

Choose the Maintain Applications Database Objects menu. Run the Validate APPS schema(s) option. Running this option checks for potential problems in your APPS schema. You must correct all problems listed in the log file before continuing on to the next step.

7. Run adadmin to compile APPS schemas

Run the Compile APPS schema(s) option. You must correct all problems listed in the log file before continuing on to the next step.

8. Run adadmin to convert to multiple reporting currencies

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

<name of your APPS schema>_MRC

Note: The text and examples in this book use the default name APPS_MRC when referring to the MRC schema.

When all problems identified in the previous step have been corrected, run the Convert to Multiple Reporting Currencies option. Running this option does the following:

- Creates the APPS_MRC schema.
- Registers the APPS_MRC schema in Oracle Applications.
- Creates database objects in the APPS_MRC schema.
- Creates MRC database triggers in the APPS schema.

Attention: Running the Convert to Multiple Reporting Currencies option can take from 5 to 15 hours for a single product installation group. For a multiple product installation group database, running this option can take from 5 to 15 hours per product installation group.

To verify that additional objects are being created in your APPS_MRC schema while adadmin is running the Convert to Multiple Reporting Currencies option, run the following SQL statements in the SYSTEM schema, where the APPS schema is named APPS and the MRC schema is named APPS_MRC:

```
SQL> select object_type, count(*) from dba_objects
2> where owner= 'APPS' group by object_type;
```

```
SQL> select object_type, count(*) from dba_objects
2> where owner= 'APPS_MRC' group by object_type;
```

The first SQL statement shows how many objects of each type are present in the APPS schema. The values returned by the first SQL statement will increase when the MRC triggers are created. The second SQL statement shows how many objects

are present in the APPS_MRC schema. The values returned by the second SQL statement should increase each time you check the progress of the conversion.

9. Review the adadmin.log file and worker log files for errors

Review the adadmin.log and adworkxx.log files for any errors that may have occurred during the installation process. The log files are located in \$APPL_TOP/admin/<dbname>/log, where <dbname> is the name of the database in which the current product group is installed.

Additional Information: [Review AutoInstall Log Files](#) in Chapter 4.

10. Verify number of PACKAGE and PACKAGE BODY object types

When you have finished running the Convert to Multiple Reporting Currencies option, verify that you have identical numbers of PACKAGE and PACKAGE BODY object types in the APPS_MRC schema.

Note that packages with names starting with FFP will not be created in the APPS_MRC schema, that is

$$\text{MRC_P} = \text{APPS_P} - \text{APPS_FFP}$$

where:

MRC_P = total number of packages in the MRC schema

APPS_P = total number of packages in the APPS schema

APPS_FFP = total number of packages in the APPS schema with names starting with FFP

You can determine the number of packages in your APPS schema that start with FFP by running the following SQL statement:

```
SQL> select owner, object_type, count(*) from dba_objects
2 where owner = 'APPS' and object_name like 'FFP%'
3 group by owner, object_type order by 2, 1;
```

11. Verify number of synonyms and views

Verify that there are more synonyms and fewer views in the APPS_MRC schema than in the APPS schema.

12. Verify that there are no database triggers in the APPS_MRC schema

13. Verify additional information about the APPS_MRC schema

Verify the APPS_MRC schema by navigating to the Oracle Users window in the System Administrator responsibility, and verifying the following information

Field	Value
Database Username	APPS_MRC
Privilege	Multi-currency
Install Group	1

If you have a multiple product installation group database, the format should be as follows:

Field	Value
Product Installation Group 1:	
Database Username	APPS_MRC
Privilege	Multi-currency
Install Group	1
Product Installation Group 2:	
Database Username	APPS2_MRC
Privilege	Multi-currency
Install Group	2
Product Installation Group 3:	
Database Username	APPS3_MRC
Privilege	Multi-currency
Install Group	3

If you have a multiple product installation group database, the data group names have the following format:

Product Installation Group	Data Group Name
1	Multiple Reporting Currencies
2	MRC Data Group 2
3	MRC Data Group 3

14. Verify the MRC data group

Navigate to the Data Groups window in the System Administrator responsibility, and confirm that the Multiple Reporting Currencies data group has been defined.

15. Create FastFormula packages in the APPS_MRC schema

The FastFormula packages, which always begin with FFP, cannot be copied from the APPS schema to the APPS_MRC schema, because they contain lines larger than 255 characters. To create all FastFormula packages in your MRC schema, run the following command:

```
$ $FF_TOP/bin/FFXBCP <MRC schema>/<MRC schema password> 0 Y %% %%
```

If you have not implemented FastFormula and/or do not have any FastFormula packages in your APPS schema, you do not need to perform this step.

Integrating Custom Objects and Schemas

If you have custom objects or custom schemas that need to be tightly integrated with Oracle Applications, follow the instructions in this section.

Suggestion: We recommend that you consult with an Oracle Applications consultant when integrating custom objects or schemas with Oracle Applications.

1. Create custom schemas

If you have custom objects in Oracle Applications schemas, you must move them to custom schemas before you integrate with an APPS schema.

Create one new schema to hold your custom data objects for each Oracle Applications schema in which your objects currently reside. Export your custom

tables, indexes, and sequences from these schemas and then import them into the new custom schemas.

Your data objects will be integrated with an APPS schema and your code objects will be created in Step 5.

Attention: Make sure your custom objects follow Oracle Applications naming conventions.

Additional Information: Naming Standards, *Oracle Applications Developer's Guide*

2. Register custom schemas

If you have not done so already, register your custom schema by using the System Administrator responsibility in Oracle Applications. Use the navigator to select Security:ORACLE:Register.

3. Determine and set install group number

If you have not done so already, set the install group number for each custom schema. You can do this by using the System Administrator responsibility in Oracle Applications. Use the navigator to select Security:ORACLE:Register.

Use install group number 0 to represent your custom schemas that need only single installations.

Therefore, if you use Multi-Org or have only one product installation group, enter 0 for the install group number for your custom schemas and skip to the next step.

For the remaining custom schemas, you must choose an install group number that matches the install group number of the Oracle Applications product it customizes. For instance, if the schema PO2 lists an install group number of 2 and your custom schema CUST_PO2 is based upon it, then you set 2 as the install group number for CUST_PO2 also.

4. Change your data groups to use the APPS schema

Using the System Administrator responsibility in Oracle Applications. Use the navigator to select Security:ORACLE:Register. Change the name in the ORACLE schema column to be the appropriate APPS schema for each data group that previously used your custom schema.

5. Create custom objects and grant access to APPS schema

If you use Multi-Org or have only one product installation group, you must:

- grant ALL privileges from each custom data object to APPS.
- create a synonym in APPS to each custom data object in every custom schema.
- create custom code objects in APPS.

Otherwise, you must create synonyms for each table and sequence in the appropriate APPS schema for the related custom schema. To do this:

- grant ALL privileges from each custom data object to each APPS schema.
- create a synonym in the APPS schema with the same install group number as your custom schema for every custom data object. For instance, create synonyms in APPS2 for CUST_PO2 objects.
- create custom code objects in each APPS schema.

Older Oracle Applications releases used the program FNDSCSGO to create cross-product grants and synonyms. This script is obsolete in Release 11 since integration is done through the APPS schema.

6. Drop duplicate code objects

After upgrading from Release 10.4 or Release 10.5, and once you determine that your customizations work correctly from each APPS schema, you can drop any duplicate code objects from your custom schemas.

Concurrent Managers

Keep the following in mind when you start the concurrent managers:

- Concurrent managers inherit directory privileges from the user who starts them. If you plan to start the managers from a login other than the main applications login, applmgr, ensure that the login has the appropriate directory privileges.

Additional Information: [Directory Privileges](#) in this appendix

- Always run the product group's environment file before you start the group's concurrent managers. This ensures that the managers inherit the correct values for environment variables such as \$APPLLOG and \$APPLOUT.
- Ensure that you are pointing to the correct ORACLE instance (i.e. \$ORACLE_HOME and \$ORACLE_SID or \$TWO_TASK are properly set).

- Startup parameter values apply in this order:
 - Values set in the startmgr script
 - Values entered on the command line
 - Values passed through your environment
 - Default values

This means that parameter values set in startmgr override any other values. Command line values override environment and default values, and so on. Another section in this manual contains more information on editing the startmgr script.

Additional Information: [Set Up Concurrent Managers](#) in Chapter 4

You cannot change directory privileges, environment variables, or startup parameters while the managers are running. To put changes into effect, shut down the managers, make the necessary modifications, and restart the managers.

Additional Information: [Restarting the Concurrent Managers](#) in this appendix

Starting the Concurrent Managers

You can start the concurrent managers from the Administer Concurrent Managers form or by running the script startmgr directly from the operating system command line. To start the concurrent manager from the operating system prompt, use the following syntax:

```
$ startmgr \  
sysmgr="<APPS username>/<APPS password>" \  
mgrname="<name>" \  
PRINTER="<printer>" \  
mailto="<userid1 userid2...>" \  
restart="N|<minutes>" \  
logfile="<filename>" \  
sleep="<seconds>" \  
pmon="<cycles>" \  
quesiz="<cycles>" \  
diag="Y|N"
```

All parameters are optional. You can pass parameters to the script in any order.

Parameters

The following entries describe the concurrent manager startup parameters. The default values apply if you do not specify different values in the startmgr script, on the command line when you run startmgr, or in your environment.

Parameter Name	Default Value	Description
sysmgr	\$FNDNAM	APPS schema name should be set to the APPS schema user ID and password. You will be prompted for the password if you omit the parameter and use the default value.
mgrname	std	Name of the internal concurrent manager (alphanumeric characters only).
PRINTER	(none)	Name of the printer to which all concurrent managers send request output if a user does not specify a printer.
mailto	user who starts managers	List of users who receive mail when the internal concurrent manager stops running.
restart	N	Number of minutes (integer) the internal concurrent manager waits before attempting to restart after abnormal termination. The default value prevents the manager from restarting after abnormal termination.
logfile	<mgrame>.mgr	The name of the internal concurrent manager's log file.
sleep	60	Number of seconds (integer) the internal concurrent manager waits between times it looks for new concurrent requests.
pmon	20	Number of sleep cycles (integer) the internal concurrent manager waits between times it checks for concurrent managers that have failed while running a concurrent request.
quesiz	1	Number of pmon cycles (integer) the internal concurrent manager waits between times it checks for normal changes in concurrent manager operation. Normal changes include the start or end of a work shift and changes to concurrent manager definitions entered in the Concurrent Managers form.

Parameter Name	Default Value	Description
diag	N	diag=Y tells all concurrent managers to produce diagnostic output regularly. The default value diag=N prevents large log files.

Example

Here is an example of running startmgr and specifying values for most parameters:

```
$ startmgr sysmgr="apps/apps" \  
mgrname="std" \  
PRINTER="hqseq1" \  
mailto="jsmith" \  
restart="N" \  
logfile="mgrlog" \  
sleep="90" \  
pmon="5" \  
quesiz="2"
```

Restarting the Concurrent Managers

You must restart the concurrent managers whenever you start the Oracle8 Server database or change the concurrent manager startup parameters. You may have to delete some files before or after you restart the concurrent managers.

Additional Information: Controlling Concurrent Managers, *Oracle Applications System Administrator's Guide*

Concurrent managers append to their own log file if the log files exist when they restart. Therefore, the user who restarts the concurrent managers must either own the existing files, have write privilege for them, or delete them before restarting.

The concurrent managers delete temporary files when each concurrent process finishes. If the concurrent managers stop abnormally, however, they may not delete these files. In this case, delete temporary files once the managers no longer need them.

Suggestion: Delete temporary files only if they have not been accessed more recently than a few days ago. This helps to prevent the loss of files required by the operating system or the concurrent managers.

File Conventions

The following table lists the locations and file naming conventions for log, output, and temporary files. The location of product log and output files depends on whether you have set up a common directory.

Table A–1 Concurrent Manager File Conventions

File Type	Location	Filename
Internal Concurrent Manager Log	\$FND_TOP/\$APPLLOG \$<PROD>_TOP/\$APPLOUT with Common Directory: \$APPLCSF/\$APPLOUT	<mgrname>.mgr
Concurrent Manager Log	\$FND_TOP/\$APPLLOG \$<PROD>_TOP/\$APPLOUT with Common Directory: \$APPLCSF/\$APPLOUT	W<nnn>.mgr
Product Log	Default: \$<PROD>_TOP/\$APPLLOG with Common Directory: \$APPLCSF/\$APPLLOG	l<request ID>.req
Product Output	Default: \$<PROD>_TOP/\$APPLOUT with Common Directory: \$APPLCSF/\$APPLOUT	<USERNAME>.<request ID>
Temporary	\$APPLTMP or \$REPORTS25_TMP	OF<abcd12345>.t

The variable parameters have the following values:

<mgrname>	The name specified with the <i>mgrname</i> parameter in the <i>startmgr</i> command. If no name is specified, the filename is <i>std.mgr</i> .
<nnn>	A sequence number between 1 and 999 is generated by the concurrent processing facility.
<PROD>_TOP	The product's top environment variable, such as <i>GL_TOP</i> .
<request ID>	The number that identifies the concurrent request.
<USERNAME>	Up to eight characters (uppercase) of the application username of the user that requested the concurrent process.

<abcd12345> Naming convention in which <abcd> are random letters and <12345> designate the operating system process ID of the concurrent process that generated the file.

Directory Privileges

Any user who runs an environment file and has access to the startmgr script can start the concurrent managers. This user then owns the log and output files that the concurrent managers create. Oracle recommends that you start the managers from the applmgr login to ensure that they inherit the correct directory privileges.

WARNING: Always start the concurrent managers from the applmgr login if you are using parallel concurrent processing on multiple nodes.

Setting the startmgr User ID with setuid

To ensure that startmgr inherits the applmgr directory privileges, you can use the UNIX setuid facility to set startmgr to the applmgr login's UNIX user ID. The concurrent managers then inherit the applmgr privileges no matter which login runs startmgr. This allows you to start or restart concurrent managers using the Administer Concurrent Managers form, regardless of the originating UNIX login. Note that you must reset the user ID with setuid if you modify or copy startmgr. Refer to your online UNIX documentation for information on setuid.

Note: The use of the setuid command may cause unexpected behavior on certain platforms that employ dynamic linking of libraries. Please refer to the *Oracle Applications Installation Update* for your platform for any information regarding this problem.

Directory Privileges for Logins other than applmgr

If you do not set the startmgr script to the applmgr user ID and you start the managers from a login other than applmgr, that login needs to have these privileges:

- Read and execute privileges on all Oracle Applications directories
- Write privilege for all directories defined by the following variables:
 - APPLLOG (typically log directory or directories)
 - APPOUT (typically out directory or directories)

- APPLCSF (common directory for log and output files)
- APPLTMP (temporary directory)
- APPLPTMP (temporary directory for PL/SQL output files)
- REPORTS25_TMP (temporary directory for Oracle Reports files)
- Write privilege for these directories: /tmp and /usr/tmp

You can verify that a login has the necessary privileges on a certain directory with this command:

```
$ ls -ld <directory>
```

Here is an example:

```
$ ls -ld /usr/tmp
drwxrwxrwx 3 root 22880 Mar 10 11:05 /usr/tmp
    ^^^
```

The three letters marked in the sample response indicate that all users have read, write, and execute privilege for the directory.

Printing

This section contains printer reference material specific to the UNIX operating system, including information on how to create and register executable printing programs.

To register printers in the Printers form of Oracle Applications, your Oracle Applications System Administrator needs to know each printer's operating system name. Your installation update tells you where to find the printer names for your platform.

Additional Information: Operating System Names for Printers,
Oracle Applications Installation Update for your platform.

Standard Print Subroutine

The standard printing subroutine that you can select in the Printer Drivers form uses Oracle Application Object Library routines to print reports. This requires fewer machine resources than printing through a customized executable program or a shell command such as lp or lpr.

When you use the subroutine, there may be options available through the descriptive flexfield at the bottom of the form. These options vary by platform and may include the following:

mail	Notify user by electronic mail when report finishes printing.
priority	Set the priority for reports in the print queue.

Check your installation update for any additional options available on your platform.

Additional Information: Standard Print Subroutine, *Oracle Applications Installation Update* for your platform

Executable Printing Programs

Oracle Applications supports the use of executable programs for printing. However, we recommend that you use executable programs only to provide features unavailable through Oracle Applications printer drivers, such as:

- Interpreting special characters in the text passed to the printer. For example, you need a program to interpret 8-bit characters sent to a 7-bit compatible printer.
- Interpreting arguments passed by Oracle Applications. For example, you need a program to perform different actions based on different output filenames.

If you do not need to support special features such as these, print through the standard printing subroutine and printer drivers defined in the Oracle Applications database. This makes the most efficient use of machine resources.

Additional Information: Overview of Printers and Printing, *Oracle Applications System Administrator's Guide*

Upgrading Existing Executable Programs

Because printing through the standard printer subroutine uses machine resources more effectively than printing through executable programs, we recommend the following if you used executable printing programs in the previous release of Oracle Applications:

- If predefined printer drivers can replace the executable program, simply register the drivers along with the printer types in the Printer Types form.
For example, Oracle Applications provides a print style Landscape and the printer driver LANDSCAPESUB. They perform the same function as the

program land, which enables DEC LN03 printers to print 132 characters per line.

Additional Information: Assign Printer Driver, *Oracle Applications System Administrator's Guide*

- If no predefined drivers will work, you may be able to create a customized driver that can replace the executable. You create drivers with the Printer Drivers form.

Additional Information: Printer Drivers, *Oracle Applications System Administrator's Guide*

- If you cannot replace the executable with a simple printer driver definition, you can continue to use the executable by registering it or the shell script that calls it with Oracle Applications.

Additional Information: [Registering Executable Programs](#) in this appendix.

Writing an Executable Program

Executable printing programs can format report output through escape sequences or a printer programming language. Creating them requires a thorough knowledge of both printer operation and a computer programming language. Follow the guidelines in this section if you need to create an executable printing program.

Printer Styles An executable program should be able to format report output for various print styles, including these:

Portrait	80 columns wide, 66 lines per page
Landscape	132 characters wide, 66 lines per page (62 lines per page on A4 style paper)
Landwide	180 characters wide, 66 lines per page (62 lines per page on A4 style paper)

Formatting Arguments If the program handles formatting for various print styles internally, you can pass arguments from the printer drivers to the program to determine which print style to use.

If the program does not contain print style formatting commands, you can define the commands in a shell script that calls the program. You then define the shell

script as the printing program in a printer driver and pass arguments that determine the print style from the driver to the script.

The printer driver that calls the executable program or shell script must be able to pass the following arguments:

- Name of the destination printer
- Number of copies to print
- Banner on title page
- Filename

Initialization and Reset You do not have to add printer initialization and reset strings to your program if you can define these strings in the Printer Drivers form.

Character Mode Oracle Reports Commands We recommend that you design your executable programs to work with the standard Oracle Reports print drivers. The following standard drivers are located in the `$FND_TOP/$APPLREP` directory:

P.prt	Portrait style
L.prt	Landscape style
W.prt	Landwide style
A.prt	A4 style

The program should not misinterpret the commands for bold on, bold off, and page size that the standard drivers imbed in Oracle Applications reports. If necessary, you can create customized Oracle Reports drivers as described below.

Location of Program When you have compiled and linked the source code or written a shell script, move the program to the `$APPLBIN` subdirectory under the top directory of your custom development area. Keep copies of the source file in your custom development area as a backup.

Creating Customized Character Mode Oracle Reports Print Drivers

The Oracle Reports print drivers set the font styles for italics, underlining, and bolding. If your executable printing program cannot use the standard Oracle Reports drivers, create a customized driver for each print style you will use with the program.

To create a customized driver, copy L.prt, P.prt, A.prt, or W.prt from `$FND_TOP/$APPLREP` to your custom development area. Modify a standard

driver as needed for your executable printing program. Give the customized driver a new filename but keep the .prt extension. Then copy the customized driver to \$FND_TOP/\$APPLREP.

Additional Information: *Oracle Reports Developer/2000 Reports 2.5 Reference Manual*

Suggestion: Use the executable program name and print style letter as the driver name. For example, use HPLJ3P.prt for portrait style printing with the executable program HPLJ3. When you print in portrait style with this program, the concurrent managers pass the HPLJ3P.prt driver to Oracle Reports as DESFORMAT=HPLJ3P.

Registering Executable Programs

When you have created your executable programs and, optionally, your shell scripts and Oracle Reports drivers, register them in the Printer Drivers form.

Additional Information: *Printer Drivers, Oracle Applications System Administrator's Guide*

Define a printer driver and corresponding print style for each print style that your executable program supports. Complete the following steps to register an executable program for a printer driver:

1. Navigate to the Printer Drivers form and create a new printer driver name. Also add the user name, description, and platform.
2. In the SRW Driver field, enter the name (without the .prt extension) of a standard or customized Oracle Reports driver. All drivers must be in the directory \$FND_TOP/\$APPLREP.
3. Enter Program as the driver method.
4. Enter No in the Spool File field.
5. Enter No in the Standard Input field.
6. In the program name field, enter the name of the executable program or the shell script that calls it. Include the full path name if this file is not in the \$FND_TOP/bin directory.
7. Add the arguments that Oracle Applications passes to the program or shell script. The driver must pass the following to the executable program:

- Name of the destination printer
 - Number of copies to print
 - Banner on title page
 - Filename
8. Add the initialization and reset strings to the appropriate fields if the program does not send these strings to the printer.

Analyzing the Database

The Oracle Applications initialization parameters set the Oracle8 Server to use rule-based optimization by default when executing a SQL statement. Some Oracle Applications SQL statements contain optimization cost-based hints, however. In executing these statements, the Oracle8 Server uses cost-based optimization.

To enable the Oracle8 Server to use cost-based optimization when the SQL or PL/SQL statements contain hints, you must gather statistics about Oracle Applications database objects by analyzing the database:

- Analyze a new system after it has a significant amount of transaction data stored.
- Analyze your system soon after finishing an upgrade.
- As part of system maintenance, analyze an Oracle Applications database approximately once a month.

Your Oracle8 Server DBA may want to write a SQL*Plus script that analyzes the Oracle Applications database. Or your DBA can run the script `ADXANLYZ.sql`, which analyzes all tables and indexes for an ORACLE user ID. Run this script for each fully installed product when the products are not in use:

```
$ cd $AD_TOP/sql
$ sqlplus <username>/<password>
SQL> @ADXANLYZ.sql
```

The script `ADXANLYZ.sql` writes a script called `Z<username>.sql` that lists the database objects owned by the username. This second script then runs the SQL*Plus `ANALYZE` command to estimate (rather than compute) statistics at a 20% sampling rate for these objects. The text file `Z<username>.lst` lists any errors that may have occurred.

Additional Information: `ANALYZE` command, *Oracle8 SQL Reference*

Here is an example of running the script for an installation of Oracle General Ledger that uses the default ORACLE user ID GL/GL:

```
$ cd $AD_TOP/sql
$ sqlplus GL/GL
SQL> @ADXANLYZ.sql
```

This creates the SQL*Plus script ZGL.sql and the output file ZGL.lst. As long as the database structure does not change, you could run ZGL.sql the next time you analyze Oracle General Ledger. Always run ADXANLYZ.sql, however, after you upgrade the system or change the database structure.

Additional Information: The Optimizer, *Oracle8 Concepts*

Pinning Packages and Sequences

Oracle Applications requires space in the ORACLE System Global Area (SGA) to execute stored packages and functions. If SGA space is fragmented, there may not be enough space for a package or function. You should pre-allocate space in the SGA shared pool for packages, functions, and sequences by "pinning" them.

Attention: You will need to run these scripts every time objects are patched using AutoPatch, or any time after objects are invalidated, either because of patching or customizations.

The ADXGNPIN.sql script pins packages and functions, while the ADXGNPNS.sql script pins sequences. Both scripts take the name of a schema from which to pin objects as an argument. This is usually the APPS schema.

```
$ cd $AD_TOP/sql
$ sqlplus SYS/<SYS password>
SQL> @ADXGNPIN.sql <schema name>
```

You must run the ADXGNPIN.sql and ADXGNPNS.sql scripts each time you start an ORACLE database instance.

Suggestion: Your database administrator must run the ADXGNPIN.sql pinning scripts automatically at startup by adding the scripts to the database startup script, which is typically dbstart in the \$ORACLE_HOME/bin directory.

Attention: Pinning all packages installed in the APPS schema will require a substantially large SGA. You may wish to make a copy of the ADXSPPIN.sql script produced by ADXGNPIN.sql and customize it for your own needs.

We also provide a script to query for objects stored in the shared pool. It shows the objects known to the SGA and the size that they consume in the SGA. You run the script with these commands:

```
$ cd $AD_TOP/sql
$ sqlplus system/<SYSTEM password> @ADXCKPIN.sql
```

Changing Oracle Applications Passwords

This section explains how to change the ORACLE password for an Oracle Applications products. Changing passwords frequently helps to ensure database security. You cannot change a username, such as APPLSYS or GL, after you install a product.

WARNING: Running the following steps out of order may lock you out of Oracle Applications.

Attention: When changing the APPS password, ensure all concurrent managers have been shut down first.

Additional Information: Administer Concurrent Managers, *Oracle Applications System Administrator's Guide*

Change Password in Oracle Applications

Sign on to Oracle Applications as the System Administrator. On a new installation, the username and password are both SYSADMIN. The System Administrator's Navigator is displayed if you have only one responsibility. If you have more than one responsibility, select System Administrator from the Responsibilities window.

Then perform the following steps, which are listed here with the default menu paths:

1. Navigate to the Register ORACLE Users form. (Security:ORACLE:Register)
2. Start a query. (From the menubar, Query -> Enter)
3. Enter the Database User Name for which you want to change the password.
4. Run the query. (From the menubar, Query -> Run)
5. Enter a new password. (The change is not displayed.) Enter the password a second time to verify it.
6. Commit the changes. (From the menubar, Action -> Save)

Note: The passwords for APPLSYS and each APPS schema — including MRC and MLS schemas — must be the same. When altering the user in SQL*Plus you must do this for APPLSYS and every APPS schema if you changed the password for one.

Change Password in SQL*Plus

Log on to SQL*Plus using the SYSTEM DBA account:

```
$ sqlplus system/<SYSTEM password>
```

Change the ORACLE password with the following command, replacing <username> with the product's ORACLE username and <new password> with the new password you created in the previous step:

```
SQL> alter user <username> identified by <new password>;  
SQL> commit;
```

Verify Password

If you changed the password for APPS, then log on to Oracle Applications to ensure your users can.

Migrating the Oracle Applications Database

This section lists the tasks you need to perform to transfer an Oracle Applications database to another database instance or to another machine. It assumes that you have already installed Release 11 of Oracle Applications, and want to transfer the entire database—migration of individual schemas or parts of an Applications database is not supported.

Attention: You can also use the instructions in this section to export from and import to the same database. In this case, complete only the tasks under the [Export the Oracle Database](#) and [Import the Oracle Database](#) headings, making sure to purge all data from the database in between.

There are many reasons for which you may want to move your Applications database. One reason is because you are upgrading the version of the Oracle8 Server you are using with Applications, either because you are moving to a server-partitioned configuration, or because you are in the process of upgrading Applications and require a higher version of the Oracle8 Server.

If you are upgrading your database to a higher version, the Oracle8 Server comes with a Migration Utility that automates the process of converting data dictionary objects and database information from an earlier release to the current one. This is an optimal way to upgrade the data in your Applications database.

Additional Information: Migration Overview in *Oracle8 Server Migration*.

You may also be performing a migration to improve scalability or performance. For example, you may have originally installed the database on the same machine as your concurrent processing server, and now want to get better scalability by dividing the two functions across different machines. Or, your database may have become significantly fragmented over time, and you want to rebuild it to improve performance. In these cases, it is typically best to perform an export and import of the database to move or rebuild it. The steps in this section detail this procedure.

You may have to modify the commands in the tasks according to the operating system used with your source or target machine. As you complete the tasks, note the following terms:

Source machine	The machine you are migrating <i>from</i> .
Target machine	The machine you are migrating <i>to</i> .

Preparing to Migrate the Database

Complete the following tasks before you export the database:

Create an Empty Database on the Target Machine

You will need to create an empty ORACLE database on the target machine before you can begin the migration. If you have not done so already, make sure you install the Oracle8 Server software on the target machine.

Attention: Always be sure you are using a version of the Oracle8 Server that is certified for use with Oracle Applications Release 11.

When creating your database, it should have the same basic structure (tablespaces, rollback segments, and initialization parameters) as the source machine's database. The import procedure creates the ORACLE user IDs and database objects.

Additional Information: [Oracle8 Server and Oracle Tools Requirements](#) in Chapter 1.

You may want to reorganize the database during the migration process. For example, you may want to create different tablespaces or resize and move database objects. We recommend that the Oracle8 Server database administrator do this on the source machine *before* exporting the database, or on the target machine *after* importing the database.

If you plan to reorganize the database *during* the migration, you need to ensure that the import processes described in the steps below will run successfully. This requires that you plan the reorganization carefully and understand export/import behavior thoroughly. To reorganize the database during the migration, you may need to manually create the tables, indexes, grants, and other objects in the target database so that you can resize them or move them to different tablespaces.

Additional Information: [Import/Export Tips, Oracle8 Utilities](#)

Run the Preparatory Scripts for the Target Database

Copy the database preparatory scripts to the \$ORACLE_HOME/dbs directory on the target machine. Follow the steps in Step 34 of Chapter 2 to run these scripts against the target database.

Attention: Make sure you obtain the scripts for the version of the target database if it is different from the source database.

Export the Oracle Database

Before you export the database, have all users sign off of Oracle Applications and shut down the concurrent managers on the source machine. Until the export finishes, users must not access the database through Oracle Applications forms or reports, or ORACLE tools such as SQL*Plus.

Additional Information: Administer Concurrent Managers, *Oracle Applications System Administrator's Guide*

Use the Oracle8 Server Export utility to export the entire database from the source machine. You can export the database as a whole or in pieces following the steps below.

Additional Information: Export, *Oracle8 Utilities*

Exporting the Entire Database

Exporting the entire database is the simpler export option. It creates an export file that is at least as large as the amount of data in the database. Do not use this method if the export file cannot fit on one disk on the source machine.

To export the entire database, move to the directory where you want to create the export file and execute the following command. Here, <SYSTEM password> is the password for the SYSTEM ORACLE user:

```
$ exp system/<SYSTEM password> file=full.dmp \
full=y buffer=1000000 grants=y log=xxxxx.log
```

When the export is finished, examine the log file to ensure there were no errors with the export.

Exporting the Database By Schema

If a full database export file will not fit on one disk, export the database to multiple export files. You first export the database structure, then export data by each ORACLE schema:

1. Export database structure with no data

Move to the directory where you want to create the export file and execute the following command to export the database structure (table, index, view, sequence, grant, and synonym definitions) without any data. Here, <SYSTEM password> is the password for the SYSTEM ORACLE user:

```
$ exp system/<SYSTEM password> file=fullnorows.dmp \
buffer=1000000 grants=y rows=n log=xxxxx.log full=y
```

2. Export data for each ORACLE schema

Export the table data for each Oracle Applications ORACLE schema used in the source machine database. This includes the schemas for all installed and dependent products, as well as the Oracle Applications Public Access schema. DO NOT export the SYSTEM or SYS schemas.

To export data for a single schema, move to the directory where you want to create the export file and execute the following command. Here, <ORACLE username>/<ORACLE password> is the username and password for the schema to be exported:

```
$ exp <ORACLE username>/<ORACLE password> \  
file=<ORACLE username>.dmp buffer=1000000 log=<xxxx>.log
```

Suggestion: If you have enough disk space available, export several schemas at the same time by running simultaneous export sessions.

3. Examine the log files to ensure there were no errors with the exports.

Attention: Do not proceed with the migration if the export contained errors and you are not certain how to resolve them.

Import the Oracle Database

Complete the following tasks to import your database on the target machine.

Move the export files from the source machine to the target machine. Then import the database on the target machine in the same way you exported the database on the source machine. You can import the database as a whole or in pieces.

Additional Information: Import, *Oracle8 Utilities*

Importing the Entire Database

Move to the directory that contains the full export file and give the following command to import the file into the target database:

```
$ imp system/<SYSTEM password> file=full.dmp buffer=200000 \  
commit=y full=y ignore=y log=<xxxx>.log
```

When the import finishes, examine the log file to ensure there were no errors with the import.

Importing the Database By Schema

If you exported the database in pieces, you first recreate the database structure from the full database export with no data. You then restore data with the exports of each ORACLE schema.

1. Import database structure with no data.

Enter the following command to import the full database export with no data:

```
$ imp system/<SYSTEM password> file=fullnorows.dmp \  
buffer=1000000 commit=y full=y indexes=n ignore=y log=<xxxx>.log
```

Note: You do not create indexes with this command because the table data imports run faster without them.

2. Import data from ORACLE schemas.

Next, import data by moving to the directory that holds the ORACLE schema export files and entering the following command for each file:

```
$ imp <ORACLE username>/<ORACLE password> file=<ORACLE username>.dmp \  
buffer=1000000 commit=y full=y ignore=y log=<xxxx>.log
```

Suggestion: You can import several user IDs at the same time by running simultaneous import sessions. Ensure that you have sufficient rollback segment space and that the imports do not result in disk contention.

3. Examine the log files to ensure there were no errors with the imports.

Attention: Do not proceed with the migration if the export contained errors and you are not certain how to resolve them.

Finishing the Migration

Complete the following tasks to finish your database migration.

Create Missing SYS Objects and Grants

A full database export does not export objects and grants from the SYS schema. After an import, certain SYS objects and grants from SYS must therefore be recreated. The database preparatory scripts you ran when preparing the migration

should have created most of these objects beforehand, but objects and grants owned by Oracle Applications will still be missing.

Run the afsys.sql script, located in the \$FND_TOP/admin/sql directory, to grant each of your APPS schemas access to additional data dictionary objects. Run the script in SQL*Plus, connected to the SYSTEM user. The script takes three arguments: the SYSTEM username, the SYSTEM password, and the name of an APPS schema to which the grant should be given. For example, to issue grants against the APPS_APPDEMO schema, you would type:

```
$ sqlplus SYSTEM/MANAGER @afsys SYSTEM MANAGER APPS_APPDEMO
```

Run this script once for each APPS schema in your database.

Note: If you have products other than Oracle Applications installed in your database that access objects in SYS and that are not listed above, you may need to recreate these grants and synonyms after you import.

Update Oracle Applications Tablespace Information

The FND_PRODUCT_INSTALLATIONS and GL_STORAGE_PARAMETERS tables store information about tablespaces used by Applications objects. If you imported objects into tablespaces with different names, you must update these tables to reflect these changes.

FND_PRODUCT_INSTALLATIONS In the FND_PRODUCT_INSTALLATIONS table, you want to update are the TABLESPACE and INDEX_TABLESPACE columns. These indicate the tablespaces that will be used for each product when creating new tables and indexes. There are two cases where you will need to update this information.

■ If you have renamed a tablespace, or relocated all objects in one tablespace to another.

If you imported the objects for a schema into a different tablespace and the originally-named tablespace no longer exists, you must update the value of these columns to an existing tablespace. Perform the following SQL statement in SQL*Plus, connected to the APPS schema:

```
$ sqlplus <APPS username>/<APPS password>
SQL> update FND_PRODUCT_INSTALLATIONS
      2> set TABLESPACE='<new ablespace>', INDEX_TABLESPACE='<new index tablespace>'
      3> where ORACLE_ID='<schema whose objects were moved>';
```

For example, if you imported all tables and indexes owned by the PO2 schema into the PO2DNEW and PO2XNEW tablespaces respectively, you would enter the following SQL to update the table:

```
SQL> update FND_PRODUCT_INSTALLATIONS
2> set TABLESPACE='PO2DNEW', INDEX_TABLESPACE='PO2XNEW'
3> where ORACLE_ID='PO2';
```

▮▮ If you want to change the tablespace where new tables or indexes are created for a product.

If you want to separate out objects created by a specific Application, rather than a specific schema, you can also update the table based on Application ID. For example, to have all new tables for MRP created in MRPD2, you would issue the following update statement:

```
SQL> update FND_PRODUCT_INSTALLATIONS
2> set TABLESPACE='MRPD2' where APPLICATION_ID=704;
```

Application IDs can be found in the table FND_APPLICATION.

Additional Information: *Oracle Application Object Library Technical Reference Manual*

GL_STORAGE_PARAMETERS In the GL_STORAGE_PARAMETERS table, you want to update the TABLESPACE_NAME column. This table stores storage information for specific objects owned by GL, particularly interim objects that are repeatedly dropped and recreated. Update the rows for objects you imported into a different tablespace from your source database, or for objects that you want to be created in a different tablespace. For example, to have the GL_SUMMARY_INTERIM table created in tablespace INTM, enter the following SQL statement in SQL*Plus, connected to the APPS user:

```
SQL> update GL_STORAGE_PARAMETERS
2> set TABLESPACE_NAME='INTM' where OBJECT_NAME='GL_SUMMARY_INTERIM';
```

The GL_STORAGE_PARAMETERS table also has storage information for individual objects, such as initial and next extent sizes. If you find it necessary to resize these objects, either during the import or while using the product, you can update these columns as well.

Additional Information: *Oracle General Ledger Technical Reference Manual*

Run AD Administration Against the Imported Database

Run the AD Administration utility on your administration server or on the target machine and connect to the *new* database you just finished importing to perform the following tasks *in the order shown* here:

- ☐ Check SYS.DUAL table
- ☐ Compile APPS schema(s)

If the export or import processes failed to recreate some Applications objects or grants properly, you may want to perform these additional steps:

- ☐ Recreate grants and synonyms for APPS schema(s)
- ☐ Compile flexfield data in AOL tables
- ☐ Maintain MultiLingual schema(s)
- ☐ Maintain Multiple Reporting Currency schema(s)

Additional Information: [AD Administration Utility \(adadmin\)](#) in Chapter 5.

Perform Post-Installation Steps on the Target Machine

If the database name or Net8 database alias for your target database differs from your source database, you must perform some database-specific post-installation steps. Perform each of the following steps, detailed in Chapter 4.

- ☐ [Create an Oracle Web Application Server Database Access Descriptor \(DAD\)](#)
- ☐ [Define Oracle Forms Cartridge parameters](#)
- ☐ [Configure Developer/2000 Metrics Server](#)
- ☐ [Modify the initial HTML signon file](#)

Additionally, if you are specifying the ORACLE_SID or TWO_TASK environment variable in any environment files or login scripts, you will need to edit those as well.

Finally, you may wish to pin the SGA packages and sequences in your target database.

Additional Information: [Pinning Packages and Sequences](#) in this Appendix.

Verify Oracle Applications Products on the Target Machine

Verify that the migration is successful by using Oracle Applications with the database on the target machine. You can then remove the database on the source machine.

Migrating the Oracle Applications Filesystem

This section lists the tasks you need to perform to relocate all or part of the Oracle Applications filesystem. These steps assume you are moving all product files to another machine, or moving all files for one or more Applications products to another filesystem on the same machine.

Moving Files On the Same Machine

Typically, when moving files on the same machine, you create a distributed installation by relocating all files for a particular product. You can also move only the files in a product's subdirectory. The Applications environment structure does not directly support this, so in order to relocate files in a subdirectory only, you must copy the files to the new location, then create a symbolic link from the original location to the new location. In this scenario, no other steps are needed.

Additional Information: The `ln` for UNIX man page for the `ln` command explains how to create symbolic links on your filesystem.

To move all files for one or more Applications products on the same machine, perform the following steps:

- 1. Make sure all users log off of Oracle Applications.**

Have all users log off of Oracle Applications and shut down any processes running on the current machine. For example, if you are moving files on a concurrent processing server, shut down all concurrent managers before continuing. Or, if you are moving files on a forms server, shut down the Oracle Forms Server Listeners before moving files.

- 2. Copy product files.**

Copy the directory trees to the new filesystems. Use the `cp -r` command to move an entire directory tree at once—for example:

```
$ cp -r /d02/app1/110/gl /d05/app12/110/gl
```

Do not use the move command or delete the old files until you are sure the migration is successful.

Attention: You cannot move the AD, FND, or admin directories. These must always be located directly under \$APPL_TOP.

3. Change environment variables.

Set the _TOP environment variables for the product files you moved to the new location. For the above example, you would set the variable GL_TOP to the new location by doing the following:

```
$ GL_TOP=/d05/appl2/110/gl; export GL_TOP
```

Do this only at the operating system prompt. Do not edit any environment files yet.

4. Recreate the main Applications environment file.

Run the AD Administration Utility and choose the option Create Applications Environment File from the Maintain Applications Files menu. After answering several questions, the AD Administration Utility will recreate your main environment file to reflect the new filesystem arrangement.

Attention: If you previously customized your main Applications environment file, make a backup before running adadmin so you can copy your customizations over to the new file.

Additional Information: [AD Administration Utility \(adadmin\)](#) in Chapter 5.

5. Read new environment file and restart background processes.

Re-read the environment file and restart any forms server or concurrent processing server processes that you shut down earlier. You may now log on to Oracle Applications to test the new filesystem arrangement. Once you are satisfied that everything is working properly, you may allow other users to access the system, and remove the files from the old directory to free up space.

Preparing to Migrate the Filesystem to Another Machine

If you are moving your product files to a different machine, complete the following tasks before you move the filesystem.

Obtain Oracle Software for the Target Machine

If the target machine is running a different operating system than the source machine, you must obtain the software for Oracle Applications and all products in the underlying Applications technology stack, including the Oracle8 Server and related tools. You should use the same release of Oracle Applications software that is on the source machine. For example, if you are running Oracle Applications Release 11 on the source machine, obtain Release 11 for the target machine.

Attention: Ensure that the target machine is running the versions of the Oracle8 Server and Oracle Tools certified for that release of Oracle Applications.

Complete Preparation Steps for the Target Machine

Perform all steps listed in Chapter 2 of this manual, *up to and including* Step 33, [Apply pre-installation patches](#). If the target machine has already been set up to use Oracle Applications, most of these steps will already be done.

Unload Oracle Applications Product Files

You must use AutoInstall to unload the necessary files on the target machine. Follow these steps to set up and run AutoInstall:

1. Set up your environment.

Define your APPL_TOP variable and set up your initial environment using the adsetenv script.

Additional Information: [Set Up AutoInstall Environment](#) in Chapter 2.

2. Run AutoInstall.

Type "adaimg" to start up AutoInstall. After answering the initial questions, you will reach the Main Menu. Since you already have an Applications database, AutoInstall connects to your database to determine what products are installed, so you do not need to select the products on the Select Products screen. However, if you want to install product files to a different filesystem, you need to go to the Product Detail screen for that particular product and enter the new location you want the files for that product to be unloaded.

Additional Information: Chapter 3, Using AutoInstall

3. Choose file-related tasks.

Select "Choose overall tasks and their parameters" from the Main Menu and select all file-related tasks. Additionally, if you are setting up a forms server or concurrent processing server, choose the tasks to generate forms or reports, as necessary.

4. Run selected tasks.

Return to the Main Menu and choose "Run Selected Tasks" to unload and relink Applications files, etc.

Completing the Migration

Follow these steps to complete the file migration process from one machine to another.

Perform Post-Installation Steps

You must perform several of the steps in Chapter 4. The exact steps you perform will depend on what installation configuration you have chosen for the target machine. Use the table below to determine what steps you should run. Do all steps in order they are listed in the table and in Chapter 4.

Perform the steps in these sections	If you are currently running on...
<input type="checkbox"/> Review AutoInstall Log Files (except Step 4), Set Oracle8 Server Parameters	any Applications server machine.
<input type="checkbox"/> Configure the Web Server	a web server.
<input type="checkbox"/> Configure the Forms Server, Set Up the Report Review Agent (Step 24 only)	a forms server.
<input type="checkbox"/> Set Up Concurrent Managers, Install Parallel Concurrent Processing (optional), Set Up the Report Review Agent (Step 24 only)	a concurrent processing server.
<input type="checkbox"/> Tasks To Do Any Time After Installing or Upgrading (Steps 37 through 40 only) *	any Applications server machine.
* For Step 40, apply file and generation portions of patches only.	

Verify Oracle Applications Products on the Target Machine

Verify that the migration is successful by using Oracle Applications with the files on the target machine. You can then remove the product files on the source machine.

Recovering Disk Space

If you need to recover disk space, you can compress or delete certain Oracle Applications product files. Oracle recommends that you do this only if you have no other way to increase your available disk space. Always back up files before you delete them and keep the backup readily available in case you need to restore files.

Additional Information: [Product Directory Structure and File Types](#) in this appendix.

Installation and Upgrade Files

After you have completed your installation or upgrade of the product group successfully, you can compress, archive, or delete the following files:

- Files in \$APPL_TOP/admin/<dbname>/log and \$APPL_TOP/admin/<dbname>/out, where <dbname> is the name of the database in which the current product group is installed.

These directories contain log and output files, respectively, for AutoInstall and other Oracle Applications utilities. You can back up and remove the files after you run AutoInstall or another utility successfully. Do not delete the directories, however.

If you are not performing any upgrades or all product groups have been upgraded to the current release, then you can compress, archive, or delete the following files:

- Files in \$APPL_TOP/admin/preupg

This directory contains SQL*Plus and other files used to prepare products for an upgrade to the current release of software.

- Files in \$<PROD>_TOP/install and \$<PROD>_TOP/upgrade (such as \$GL_TOP/install and \$GL_TOP/upgrade)

You can remove all the files under the /install and /upgrade directories of each product directory tree as these are only used during an upgrade.

Attention: Do not remove any files under the /admin directory as these are used by AD utilities such as AutoPatch and AD Administration.

Form Definition Files

Do not delete Oracle Forms source files (.fmb) or PL/SQL libraries (.pll) under \$AU_TOP. These are necessary for generating Applications forms after patching them using AutoPatch.

Library Files

Oracle recommends that you do NOT compress or delete the library files (extensions .a and .o) in the \$<PROD>_TOP/lib directories. These are needed to relink programs if, for example, you update them using AutoPatch or upgrade the Oracle8 Server. These files take approximately 40 MB of disk space on a system that has all products installed.

Obsolete Files

After you run AutoPatch to update Oracle Applications, you can delete old files, which end in O. Once you have verified the patch was successful, you do not need to back up these files before deleting them. The amount of space recovered depends on the number of files that were replaced.

Additional Information: [After Running AutoPatch](#) in Chapter 5

Executable Program Compression Utilities

Oracle Applications automatically uses certain utilities to reduce the size of executable programs. Therefore, you do not need to run these utilities:

- | | |
|-------|---|
| mcs | On platforms that support the mcs utility, Oracle Applications runs mcs automatically when relinking product files. This utility reduces the size of executable programs by removing all entries in the executable program's comment section. |
| strip | Oracle Applications uses a linking option to remove symbol information from executable programs. The strip command performs the same function, and does not need to be run manually. |

Product Directory Structure and File Types

All Oracle Applications product directories have similar subdirectory structures. This section lists the main subdirectories under each product directory and describes by file extension the types of files held in each directory. Files that end with capital O may appear in any subdirectory. These are backed-up versions of files that were replaced by AutoPatch.

Table A–2 Applications Directory and File Types

Subdirectory Name	Description
/admin, /install, + /upgrade	Contain files used during an installation and upgrade. Contains these file types:
▪ /import	▪ .ctl (DataMerge file) ▪ .dat (DataMerge or import parameter file) ▪ .dmp (export file) ▪ .exp (DataMerge file)
▪ /driver	Contains .drv files (driver files).
▪ /odf	Contains .odf files (object description files).
▪ /sql	Contains the following file types: ▪ .sql (SQL*Plus scripts used during an installation or upgrade) ▪ .pkh, .pkb, .pls (scripts to create PL/SQL stored procedures)
/bin	Contains concurrent programs and other C language programs.
/forms	Contains .fmx files (Oracle Forms form files). The form description files (.fmb extension) are unloaded here, then moved to \$AU_TOP.
/html	Contains .html files (HTML files), which are copied to \$OAH_TOP.
/java	Contains .class files (Java class files) and .jar files (Java ARchives), which are copied to \$JAVA_TOP.
/lib	Contains files used to relink concurrent programs with the Oracle8 Server: ▪ .a (library file that contains compiled code common to the product's programs) ▪ .mk (makefile that specifies how to create linked programs) ▪ .o (object file that contains compiled code specific to one of the product's programs)

Table A-2 Applications Directory and File Types

Subdirectory Name	Description
/log and /out	Contains output files for concurrent programs: <ul style="list-style-type: none"> ■ .mgr (master log file for concurrent manager) ■ .req (log file for a concurrent process) <p>Note: /log and /out subdirectories under a product directory are not used if you set up a common directory for log and output files except in \$FND_TOP. See Concurrent Manager File Conventions in this appendix for more information.</p>
/mesg	Contains .msb files (binary message files).
/plsql	Location where .pll files (PL/SQL library files for Oracle Reports) are unloaded, after when they are moved to the \$AU_TOP/plsql directory.
/reports	Contains Oracle Reports .rdf files, which are platform-specific binary report files.
/resource	Contains .pll files (PL/SQL library files for Oracle Forms), which are copied to \$AU_TOP.
/sql	Contains .sql files (SQL*Plus scripts).

Environment Variables

The table in this section lists the environment variables defined for Oracle Applications. The table includes the following headings:

Variable	The name of the environment variable.
Defines	What the environment variable defines.
Default	Default value. "Set on install" means that AutoInstall sets the variable according to your installation criteria.
Change	Whether you can change the value of the variable.
File	The environment file that defines the variable. The file APPLSYS.env, which may have a different name on your system, is in the \$APPL_TOP directory. The file fndenv is in the \$FND_TOP directory.

You can find the value of an environment variable with this command:

```
$ echo ${<variable>}
```

Table A-3 Environment Variables

Variable	Defines	Default	Change	File
<PROD>_TOP	Product top applications directory, as in GL_TOP	(set on install)	yes	APPLSYS.env
APPCPNAM	Controls whether concurrent manager log/output filenames are in DOS-compatible 8.3 format.	(set on install)	yes	APPLSYS.env
APPLBIN	Subdirectory for binary executable files	bin	no	fndenv
APPLCSF	Optional common directory for log and output files	(none)	yes	APPLSYS.env
APPLDCP	Whether parallel concurrent processing is off, on, or on with load-balancing (OSQ)	(set on install)	yes	APPLSYS.env
APPLDOC	Location of on-line documentation files (copied to SOAD_TOP)	docs	no	fndenv
APPLFENV	Name of main environment file (<dbname>.env, where <dbname> is the name of the database in which the current product group is installed.	(set on install)	no	<dbname>.env
APPLFRM	Subdirectory for form files	forms	no	fndenv
APPLFULL	Short names of fully installed products	(set on install)	no	<dbname>.env
APPLINC	Subdirectory for include files of custom applications	include	yes	fndenv
APPLLIB	Subdirectory for C library files	lib	no	fndenv
APPLLOG	Subdirectory for log files	log	yes	<dbname>.env
APPLMAIL	Mail product linked to Oracle Alert	(set on install)	yes	<dbname>.env
APPLMSG	Subdirectory for message files	mesg	no	fndenv
APPLMSK	File creation mode mask for files written by concurrent managers	022	yes	fndenv
APPLORB	Name of bitmap mode Oracle Report executable	none	no	fndenv
APPLORC	Name of character mode Oracle Reports executable	none	no	fndenv
APPLOUT	Subdirectory for output files	out	yes	<dbname>.env
APPLPLS	Subdirectory for PL/SQL libraries	plsql	no	fndenv

Table A-3 Environment Variables

Variable	Defines	Default	Change	File
APPLPTMP	Tells Concurrent Manager which directory on Database server temporary PL/SQL output files are written to	plsql	no	fnenv
APPLREG	Subdirectory for regression tests defined for custom applications	regress	yes	fnenv
APPLREP	Subdirectory for Oracle Reports report files	reports	no	fnenv
APPLRGT	Subdirectory for files generated while running regression tests in compare mode	regress	yes	fnenv
APPLRSC	(not used presently)	resource	no	fnenv
APPLSAV	Subdirectory for old regression tests	save	yes	fnenv
APPLSHAR	Short names of shared products needed for relinking	(set on install)	no	<dbname>.env
APPLSQL	Subdirectory for SQL*Plus files	sql	no	fnenv
APPLTMP	Directory for Oracle Applications temporary files	/tmp or /usr/tmp	yes	<dbname>.env
APPLUSR	Subdirectory for user exits	usrxit	no	fnenv
APPL_TOP	Applications top directory	(set on install)	yes	<dbname>.env
CLASSPATH	List of directories and zip files to be scanned for Java class files needed at runtime	(varies)	yes	adovars.env
FNDNAM	Name of main APPS schema	(set on install)	no	<dbname>.env
FORMS45_%, CNTL_BREAK	Variables used by Oracle Forms 4.5	(set on install)	no	<dbname>.env
GWYUID	ORACLE user ID for applications sign-on security form	APPLSYSPUB/ PUB	no	<dbname>.env
JAVA_TOP	Directory where all Applications Java class files reside	(none)	yes	adovars.env
LD_LIBRARY_ PATH	List of directories to be scanned for dynamic library files needed at runtime	(varies by platform)	yes	adovars.env
NLS_DATE_ FORMAT	Format for dates	DD-MM-RR	no	<dbname>.env
NLS_LANG	Language, territory, and character set values	(set on install)	yes	<dbname>.env
NLS_NUMERIC_ CHARACTERS	Characters used as numeric separators	. ,	no	<dbname>.env
NLS_SORT	Defines collating sequence for ORDER BY queries	BINARY	no	<dbname>.env

Table A–3 Environment Variables

Variable	Defines	Default	Change	File
OAD_TOP	Directory for context-sensitive documentation (in HTML)	(none)	yes	adovars.env
OAH_TOP	Directory for HTML files and CGI scripts (/bin)	(none)	yes	adovars.env
OAM_TOP	Directory for Media files (GIF files, etc.)	\$JAVA_TOP/ oracle/apps	no	adovars.env
OA_DOC, OA_HTML, OA_MEDIA	Equivalent to virtual directory mappings for HTML, Media, and context-sensitive documentation files	\$OAD_TOP/doc, \$OAH_TOP/html, \$OAM_TOP/media	no	adovars.env
OA_JDK_TOP	Location of Java Development Kit or Java Runtime Environment on your system.	(none)	yes	adovars.env
PATH	Path for executable files	(set on install)	yes	<dbname>.env
PLATFORM	Computer platform	(varies)	no	<dbname>.env
REPORTS25_PATH	Path for Oracle Reports version 2.5 files	(set on install)	yes	<dbname>.env
REPORTS25_TMP	Directory for Oracle Reports version 2.5 temporary files	/tmp or /usr/tmp	yes	<dbname>.env

Territory Names

The table in this section lists the territories and the equivalent NLS Territory Names, and indicates which territories have specific localizations in the Oracle Applications Localization products. Your NLS_LANG environment variable setting does not affect which localizations you install or have access to.

Additional Information: [Set NLS environment variables](#) in Chapter 2.

Table A–4 Territory Names

Territory Name	NLS Territory Name	Localizations
Argentina	ARGENTINA	Y
Australia	AUSTRALIA	Y
Austria	AUSTRIA	Y
Belgium	BELGIUM	Y
Bolivia	BOLIVIA	Y
Brazil	BRAZIL	Y

Table A-4 Territory Names

Territory Name	NLS Territory Name	Localizations
Bulgaria	BULGARIA	N
Canada	CANADA	Y
Chile	CHILE	Y
China	CHINA	Y
CIS (formerly USSR)	CIS	N
Colombia	COLOMBIA	Y
Costa Rica	COSTA_RICA	Y
Croatia	CROATIA	N
Czech Republic	CZECH_REPUBLIC	Y
Denmark	DENMARK	Y
Dominican Republic	DOMINICAN_REPUBLIC	Y
Ecuador	ECUADOR	Y
Egypt	EGYPT	N
El Salvador	EL_SALVADOR	Y
Finland	FINLAND	Y
France	FRANCE	Y
Germany	GERMANY	Y
Greece	GREECE	N
Guatemala	GUATEMALA	Y
Honduras	HONDURAS	Y
Hong Kong	HONG KONG	N
Hungary	HUNGARY	Y
Iceland	ICELAND	N
Israel	ISRAEL	Y
Italy	ITALY	Y
Jamaica	JAMAICA	Y
Japan	JAPAN	Y

Table A–4 Territory Names

Territory Name	NLS Territory Name	Localizations
Korea, Republic of	KOREA	Y
Lithuania	LITHUANIA	N
Mexico	MEXICO	Y
Netherlands	THE NETHERLANDS	Y
Nicaragua	NICARAGUA	Y
Norway	NORWAY	Y
Panama	PANAMA	Y
Paraguay	PARAGUAY	Y
Peru	PERU	Y
Poland	POLAND	Y
Portugal	PORTUGAL	Y
Puerto Rico	PUERTO_RICO	Y
Romania	ROMANIA	N
Singapore	SINGAPORE	Y
Slovenia	SLOVENIA	N
Spain	SPAIN	Y
Sweden	SWEDEN	Y
Switzerland	SWITZERLAND	Y
Taiwan, Republic of China	TAIWAN	Y
Thailand	THAILAND	Y
Trinidad and Tobago	TRINIDAD_AND_TOBAGO	Y
Turkey	TURKEY	Y
United Arab Emirates	UNITED ARAB EMIRATES	N
United Kingdom	UNITED KINGDOM	N
United States	AMERICA	N
Uruguay	URUGUAY	Y
Venezuela	VENEZUELA	Y

Language Codes and Character Sets

The following table lists the languages supported. Language translations are not necessarily available on your platform. Check with your Oracle representative for more information on the availability of translations. The table contains the following headings:

Language	Name of the language.
Code	NLS language code you use to designate the language when you run AutoInstall. Language translation directories use this code, as in \$GL_TOP/forms/F.
Character Set	The character set is used for the installation media (that is the "tape character set") for each language, such as KO16KSC5601 for Korean. You set the character set portion of NLS_LANG to the character set your admin platform supports. The character set of the installation media, your admin platform, and the database do not have to be identical, but there must be a character translation between those that are different such that all characters can be translated.

Additional Information: Language Directory Structure, *Oracle Applications Concepts*

Table A-5 Language Codes and Character Sets

Language	Code	Character Set
American English	US	WE8ISO8859P1
Arabic	AR	AR8ISO8859P6
Brazilian Portuguese	PTB	WE8ISO8859P1
Canadian French	FRC	WE8ISO8859P1
Czech	CS	EE8ISO8859P2
Danish	DK	WE8ISO8859P1
Dutch	NL	WE8ISO8859P1
Finnish	SF	WE8ISO8859P1
French	F	WE8ISO8859P1
German	D	WE8ISO8859P1
Greek	EL	EL8ISO8859P7

Table A-5 *Language Codes and Character Sets*

Language	Code	Character Set
Hebrew	IW	IW8ISO8859P8
Hungarian	HU	EE8ISO8859P2
Italian	I	WE8ISO8859P1
Japanese	JA	JA16EUC
Korean	KO	KO16KSC5601
Latin American Spanish	ESA	WE8ISO8859P1
Norwegian	N	WE8ISO8859P1
Polish	PL	EE8ISO8859P2
Portuguese	PT	WE8ISO8859P1
Romanian	RO	EE8ISO8859P2
Russian	RU	CL8ISO8859P5
Simplified Chinese	ZHS	ZHS16CGB231280
Slovak	SK	EE8ISO8859P2
Spanish	E	WE8ISO8859P1
Swedish	S	WE8ISO8859P1
Thai	TH	TH8TISASCII
Traditional Chinese	ZHT	ZHT16BIG5
Turkish	TR	WE8ISO8859P9

Reserved Words

Do not use the following reserved words as environment or registry variables. In general, do not use any name that starts with APPL, NLS, FORMS45, or an Oracle Applications product short name, such as FND, AD, or GL.

Additional Information: Release 11 Products and Product Families,
Oracle Applications Concepts

AD	CHMOD	MFG_TOP
AD_TOP	CHV_TOP	MRP_TOP
AK_TOP	CLIBS	MSC_TOP
ALR_TOP	CN_TOP	MV

AP_TOP	CP	NETLIBS
APPCPNAM	CRP_TOP	OA_DOC
APPL_CONFIG	CS_TOP	OA_HTML
APPL_ROOT	CZ_TOP	OA_JAVA
APPL_TOP	DPFBUG	OA_JDK_TOP
APPLBIN	DT_TOP	OA_MEDIA
APPLCFRM	EC_TOP	OAD_TOP
APPLCSF	ENG_TOP	OAH_TOP
APPLFENV	FA_TOP	OAM_TOP
APPLFRM	FDBDMCHK	OE_TOP
APPLFULL	FDCPCHK	ORACLE_LINK
APPLHLP	FDMSGCHK	ORALIBS
APPLIMP	FDOPGOAL	OTA_TOP
APPLINC	FDSQLCHK	OTHERLIBS
APPLINP	FF_TOP	PA_TOP
APPLINST	FILE	PATH
APPLLDR	FLM_TOP	PAY_TOP
APPLLIB	FND_CRT	PCC
APPLLOG	FND_RT	PCCFLAGS
APPLMAIL	FND_RTCP	PCCINC
APPLMSG	FND_TOP	PER_TOP
APPLMSK	FND_ZOOM	PJM_TOP
APPLNQS	FNDFAPP	PLATFORM
APPLORB	FNDLANG	PO_TOP
APPLORC	FNDMAPP	PRCOPIES
APPLOUT	FNDMENU	PRGLIB
APPLPLS	FNDNAM	PROLIBS
APPLPLUS	FNDOTHERLIBS	PRSAVOUT
APPLPTMP	FORMS45_APPS_LIBS	QA_TOP
APPLREG	FORMS45_MAPPING	RANLIB
APPLREP	GHR_TOP	RELINK_LIBS
APPLRGT	GL_TOP	RELINKM
APPLRMF	GWYUID	REPORTS25_PATH
APPLRSC	HXT_TOP	REPORTS25_TMP
APPLSAV	ICX_TOP	REPORTS_PATH
APPLSHAR	INCLUDE_FLAGS	REPORTS_TMP
APPLSQL	INV_TOP	RG_TOP
APPLSUB	JA_TOP	RLA_TOP
APPLTMP	JAVA_TOP	RM
APPLUSR	JE_TOP	SCP
APPLMLANG	JG_TOP	SPCCHANL
AR_TOP	JL_TOP	SQLLIB
AS_TOP	LDFLAGS	SSP_TOP
AU	LFNDLIB	VEH_TOP
AU_TOP	LN	VNDARPL

AX_TOP	LUSRIAP	VNDARSL
AZ_TOP	LUSRLIB	VNDPAYPL
BOM_TOP	LUSRPRG	VNDPAYSL
CC	LUSRSRW	WIP_TOP
CE_TOP	MAILLIB	XITOBJ
CFLAGS	MCS	XITSRW

Oracle Application Object Library Reference Material

This appendix provides operating system-specific information about developing applications using Oracle Application Object Library. This information is specific to the UNIX operating system and serves as a supplement to the *Oracle Applications Developer's Guide*. This appendix includes specific information on these topics:

- Development Environment
- Message Dictionary
- Concurrent Processing

Development Environment

This section explains how to set up the directory structure and TOP environment variable for a custom application.

Create Directory Structure

You must create an operating system directory structure for each custom application that you build. We recommend that you use the application short name for the custom application's top directory. For example, the top directory for Oracle Application Object Library uses the application short name `fnd`. Beneath the top directory, create the subdirectories listed in the following table:

Directory	Contents	fnndev Variable
bin	Executable code of concurrent programs written in a programming language such as C, as well as any SQL*Loader control (.ctl) files or any operating system scripts.	\$APPLBIN
forms	Oracle Forms .fmb and .fmx files (separated into subdirectories by language codes).	\$APPLFRM
lib	Compiled object code and linking makefile for concurrent programs.	\$APPLLIB
log	Log files from concurrent processes.	\$APPLLOG
mesg	Base language and translated message files (see below).	\$APPLMSG
out	Concurrent process output such as reports.	\$APPLOUT
plsqli	PL/SQL libraries used with Oracle Reports.	\$APPLPLS
sql	Concurrent programs written with SQL*Plus (.sql files) or PL/SQL.	\$APPLSQL
reports	Concurrent programs written with Oracle Reports (.rex and .rdf files).	\$APPLREP

You may also have additional subdirectories such as include, admin (added in Release 11), and src for your application. If you have install and upgrade directories for your custom application, you should move their contents to admin for Release 11.

Additional Information: File System, *Oracle Applications Concepts*

The directory names in the table match the environment variable values defined in the fnndev file in the directory \$FND_TOP. For example, fnndev sets the variable \$APPLBIN to "bin" and the variable \$APPLLIB to "lib". These particular values enable Oracle Application Object Library to find executable code and to write output files, respectively.

Attention: Oracle Applications installation and patch routines use the default subdirectory names, regardless of how the values of the APPL variables are set in `findenv`.

Language Directories

Set up a language directory under your custom application's forms, sql, and reports directories for each language you plan to use, including American English.

Use the appropriate NLS language code to name the language subdirectories. For example, to run an American English installation, create a US directory under the sql, reports, and forms directories. To run a German installation, create a D directory under the sql, reports, and forms directories.

Additional Information: Language Directory Structure, *Oracle Applications Concepts*

TOP Environment Variable

Oracle Application Object Library uses a TOP environment variable (also known as `PROD_TOP` or `PRODUCT_TOP`) to find your application's directory tree. Define this variable in the environment file (extension `.env`) in the `$APPL_TOP` directory. Add the following lines to the end of the file:

```
<PROD>_TOP="<directory path>"
export <PROD>_TOP
```

where the value `<PROD>` is your application short name. Here is an example:

```
XXFAC_TOP="/mis/applbase/xxfac"
export XXFAC_TOP
```

Reread the modified environment file to add the new TOP variable to your environment:

```
$ . <filename>.env
```

Here is an example:

```
$ . APPLSYS.env
```

Before you run concurrent programs associated with the application, update the environment used by the concurrent managers by shutting down and restarting the managers.

Additional Information: Starting the Concurrent Managers in Appendix A

Users of the custom application should execute the environment file automatically at login. The environment file executes `findenv`, which in turn executes `devenv` in `$FND_TOP/$APPLUSR`.

Other Environment Variables in `devenv`

These environment variables are defined in `devenv`. You should not change them except where noted.

CC The name of the C compiler on this platform. Usually 'cc'.

CFLAGS The arguments you must specify on the C compiler command line to compile a custom C program for linking with Oracle Applications code. **CFLAGS** includes information on where to look for C header files (**INCLUDE_FLAGS**) and defines the platform name.

CFLAGS should always include the symbol 'AFSTUBS'.

INCLUDE_FLAGS Specifies a list of directories to check when trying to find an include file.

You may add directories to this list, but do not remove anything from it.

PCC The name of the Pro*C pre-compiler for this RDBMS release. This is usually `$ORACLE_HOME/bin/proc`. You may change this name if necessary.

PCCFLAGS The arguments you must specify on the Pro*C compiler on the command line to correctly pre-compile a Pro*C program.

If you use PL/SQL in your Pro*C programs, you should change 'sqlcheck=syntax' to 'sqlcheck=full'. You then add

```
userid=<username>/<password>
```

where `username` and `password` are the Oracle Server schema name and password to which Pro*C connects while compiling your programs.

We recommend against changing any arguments in PCCFLAGS that are not mentioned here. You may add more arguments to PCCFLAGS.

PCCINC The list of directories to check when trying to 'EXEC ORACLE INCLUDE' a Pro*C header file.

You may add directories to this list, but do not remove anything.

RM The Unix remove file command. This is usually 'rm' or 'rm -f'.

CP The Unix copy file command. This is usually 'cp'.

MCS The Unix MCS command. This is usually 'echo mcs' or '/usr/bin/mcs'.

CHMOD The Unix change file protections command. This is usually 'chmod'.

Message Dictionary

After you define and save messages with the Messages window, you submit a concurrent request to create your application message files using the Generate Messages program. Message Dictionary stores the messages in the database and creates message files in your application's \$APPLMSG subdirectory (typically mesg).

The files are created for the language you choose when you submit your request. For example, if the Language Code you choose is US (American English), and you use the DB_TO_RUNTIME mode, the concurrent request creates the file US.msb. The human-readable file would be US.msg (created using the DB_TO_SCRIPT mode).

Additional Information: Create Your Message Files, *Oracle Applications Developer's Guide*

Printing Lists of Application Messages

You can print the .msg file from your operating system. Use any print command or routine that prints 80-character wide output. The printout shows you the following:

- message names
- message numbers
- message text
- message tokens, if any

Calling Message Dictionary from a User Exit or Concurrent Program

To call Message Dictionary routines from any C routine (user exit or concurrent program), you must include the header file `fddutl.h`. You can find this file in the directory `$FND_TOP/$APPLINC`.

Concurrent Processing

This section explains certain procedures and conventions for creating customized concurrent programs:

- Log and Output Filenames
- Oracle Tool Concurrent Programs
- Pro*C Concurrent Programs
- Submitting Concurrent Requests (CONCSUB)

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

Log and Output Filenames

Log and output files must have specific names and locations for users to review the files online with the Requests window.

The `fdpwrt()` Routine

If you use the Oracle Application Object Library routine `fdpwrt()` to write to files, the concurrent managers automatically name the files according to the operating system's naming conventions. This method of writing to files is completely portable. You do not have to rewrite your programs to name your log and output files differently if you port your application to another platform.

Standard Names

Oracle Application Object Library permits the use of standard C functions such as `fopen()`. These functions are supported for upward compatibility of programs written with older versions of Oracle Application Object Library. If you do not use `fdpwrt()` to write to log and output files, you must follow the standard naming and directory conventions for log and output files listed on page 2.

However, Oracle Application Object Library reserves the right to change the method of handling log and output files, for example in the implementation of a distributed environment. In this case you may need to change your code to handle the new implementation.

File Type	Location	Filename
Log	<i>Default:</i> \$<PROD>_TOP/\$APPLLOG with Common Directory: \$APPLCSF/\$APPLLOG	l<request ID>.req
Output	<i>Default:</i> \$<PROD>_TOP/\$APPLOUT with Common Directory: \$APPLCSF/\$APPLOUT	<USERNAME>.<request ID> or O<request ID>.out based on value of APPCPNAM

The variable parameters shown in this have the following values:

<PROD>_TOP The application's top environment variable.

<Request ID> The number that identifies the concurrent request.

<USERNAME> Up to eight characters (uppercase) of the application username of the person who requested the concurrent process.

Oracle Tool Concurrent Programs

If you write concurrent programs in PL/SQL, SQL*Plus, or Oracle Reports, name the program exactly as you identified it in the Execution File field of the Concurrent Program Executable window, plus an extension if necessary.

The following table lists the file extensions used for these programs and the directories where the programs should reside. (This does not apply to PL/SQL stored procedures, which are stored in the database.) The directories are under your custom application's TOP directory, \$<PROD>_TOP.

If you use shared PL/SQL libraries with your Oracle Reports programs, and you want to include the libraries you write for your custom application, place the libraries in the \$APPLPLS directory under your custom application's TOP directory.

Tool	Ext.	Directory	Notes
SQL*Plus and PL/SQL	.sql	\$APPLSQL	The program name is case sensitive and must exactly match the Execution file you defined with Oracle Application Object Library.
Oracle Reports:	.rdf	\$APPLREP	Oracle Application Object Library looks for the .rdf file first. It uses the .rex file if it does not find the .rdf file. The program name is case sensitive and must exactly match the Execution file you defined with Oracle Application Object Library.

Pro*C Concurrent Programs

When you write a concurrent program in Pro*C, copy the skeleton programs EXMAIN.c and EXPROG.c from the directory \$FND_TOP/\$APPLUSR. Rename the files and globally replace SUBROUTINE_NAME with the name of your subroutine.

EXMAIN.c is the skeleton used for your spawned programs. EXPROG.c is the skeleton used for your program's logic. This module can be used to create a spawned or an immediate program. For immediate programs, you must include your copy of EXPROG.c in a program library. See below for information on building a program library.

You can use programs written with these skeleton programs as spawned or immediate concurrent programs. Spawned programs run as a separate process while immediate programs run linked in with a concurrent manager.

Attention: Oracle Corporation provides information on immediate concurrent programs for backwards compatibility only. We strongly recommend that you do not create any new immediate concurrent programs. You should define your new Pro*C concurrent program executables as spawned.

Naming your Executable File

Name your program's executable file exactly as you identified it in the Execution File field of the Concurrent Program Executable window. Put your executable file in the \$APPLBIN directory under your application's TOP directory.

Building your Program Library

Register a new program library with the Register Concurrent Program Library form and register all the programs you want to include in this library. Then enter "Yes" in the Rebuild field and commit. This creates a request to build a new catalog file called <Library Name>.c under \$<PROD>_TOP/\$APPLLIB\$. You should compile the <Library Name>.c file after the request completes.

Sample program libraries such as prgcat.c and prglib.c are located under \$FND_TOP/\$APPLUSR.

Suggestion: For ease of maintenance, define your concurrent program executables as spawned.

Compiling C and Pro*C Programs

Your environment for compiling custom code depends on the file \$FND_TOP/usrxit/devenv. If you change this file, you should reread it by logging in again so that the changes take effect.

You compile your C or Pro*C programs into object modules using \$FND_TOP/usrxit/Makefile. You then link your programs using adrelink. We do not support both compiling and linking executables using a single makefile or utility.

To compile the C program example.c, use the following syntax. In all the examples, you should run the commands from the directory in which your files are located.

```
$ make -f $FND_TOP/usrxit/Makefile example.o
```

To compile the Pro*C program proexamp.pc, use the following syntax:

```
$ make -f $FND_TOP/usrxit/Makefile proexamp.o
```

To compile the four C and Pro*C programs a.c, b.c, c.pc, d.pc (all of which are in the current directory), use the following syntax:

```
$ make -f $FND_TOP/usrxit/Makefile a.o b.o c.o d.o
```

Linking Spawned Concurrent Programs as Stand-alone Programs

If you want your spawned concurrent program to run as a stand-alone program, perform the following steps before compiling your stand-alone executable.

For custom concurrent programs you define under your custom application (as recommended), you should copy the sample.mk file from \$FND_TOP/usrxit to your \$<PROD>_TOP/\$APPLLIB directory. Modify your copy according to the instructions contained in the file. This is the file adrelink uses to link your stand-alone executables.

Then enter the following commands.

```
$ . $FND_TOP/fndenv
```

Move to the directory in which your source files are kept.

```
$ cd <source_directory>
$ make -f $FND_TOP/$APPLBIN/Makefile <source file>.o
```

Here, <source file> is the name of the file containing your program and <directory> is the directory where the source file is located.

You can then link your stand-alone executable and place the executable in the \$APPLBIN directory under the TOP directory for your custom application:

```
$ adrelink force=y "<appl_short_name> <program name>"
```

In this relink command, <appl_short_name> is the application short name of the application your program belongs to, and <program name> is the program name.

Linking your Immediate Concurrent Program

To create a program library, you link your compiled library catalog with your program object files using an Oracle Application Object Library link procedure.

Attention: Oracle Corporation provides information on immediate concurrent programs for backwards compatibility only. We strongly recommend that you do not create any new immediate concurrent programs. You should define your new Pro*C concurrent program executables as spawned.

Make sure the environment variable \$LUSRLIB includes the modules that define the functions for the immediate concurrent programs and your program library. Set the \$LUSRPRG variable to include the object modules of your library catalog. The

file `devenv` in the directory `$FND_TOP/$APPLUSR` defines the variables `$LUSRLIB` and `$LUSRPRG`. The file `fnenv` executes `devenv`.

The files `devenv` and `fnenv` are UNIX shell scripts that set up the necessary environment variables.

We recommend that you make a copy of the working program library before linking your new immediate concurrent program library in case your new program library does not function as expected. To link your program library, execute this command from the operating system:

```
$ adrelink force=y "fnd UFNDLIBR"
```

This creates your new program library as `UFNDLIBR`. You can rename it, but the name of your new program library must be eight characters or less.

Testing Pro*C Concurrent Programs

You can use the following method to test your program. You must pass each argument needed by your program. To pass parameters, enter the following at the operating system prompt:

```
$ <program name> <ORACLE username>/<ORACLE password> 0 Y \  
[<parameter 1> <parameter 2>... ]
```

The program name must be uppercase and the same name that you entered in the Execution File field of the Concurrent Program Executable window. The 0 and Y arguments are required.

If any of your program-specific parameters includes spaces, enclose that parameter in double quotes. If a parameter contains a literal double quote, precede that mark with a backslash [`\`].

Host Language Concurrent Programs

Name your program `<name>.prog`, where `<name>` is the value you enter in the Execution File field of the Concurrent Executable window. Then make a symbolic link using your execution file name (without an extension) to `fnpcpsr`, which is located in the `$FND_TOP/$APPLBIN` directory. Put your executable file and the linked file in the `$APPLBIN` directory under your application's TOP directory.

For example, name your custom shell script `CUSTOM.prog`. Create a symbolic link to `fnpcpsr` named `CUSTOM`. Place both files in your `$APPLBIN` directory. Create your concurrent program executable using the execution file `CUSTOM`.

Host Program Parameters

The concurrent manager running your program puts your program name in \$0, the four arguments orauser/pwd, userid, username, and request_id in \$1 to \$4, and your program specific parameters in \$5 and beyond. Each of these arguments can be at most 50 characters.

For example, if you pass two parameters into your program, you use \$5 to refer to the first parameter and \$6 to refer to the second parameter.

Protecting Your Oracle User Password

In some cases, there are security concerns with passing your Oracle username and password directly to your program. If you do not want the concurrent manager to pass your orauser/pwd to your program, you can have the manager pass it as an environment variable instead.

When defining a concurrent program using a concurrent program executable defined as a HOST program, enter the term "encrypt" in the Execution Options field of the Concurrent Programs window. This signals the concurrent manager to pass the orauser/pwd in the environment variable fcp_login. The argument \$1 is left blank.

Success Codes

By default, a shell script returns success (status code 0). If your script traps an error, use the UNIX exit command "exit 1" to return failure (status code 1) to the concurrent manager running the program.

Testing Your Program

You should test using the <name>.prog file to make sure your script behaves correctly.

Submitting Concurrent Requests (CONCSUB)

After defining a concurrent program executable, use the Concurrent Programs window to define an instance of your executable file. To protect your data during an upgrade, register the program with your custom application. Define a user-friendly, descriptive name as well as a short name for the concurrent manager to use with request submissions.

Additional Information: Concurrent Programs, *Oracle Applications Developer's Guide*

Syntax

You can submit a concurrent request to run any concurrent program by running the CONCSUB program with the following syntax:

```
$ CONCSUB <APPS username>/<APPS password> \
<responsibility application short name> \
<responsibility name> \
<username> \
[WAIT=N|Y|<n seconds>] \
CONCURRENT \
<program application short name> \
<program name> \
[PROGRAM_NAME="<description>"] \
[REPEAT_TIME=<resubmission time>] \
[REPEAT_INTERVAL=<number>] \
[REPEAT_INTERVAL_UNIT=<resubmission unit>] \
[REPEAT_INTERVAL_TYPE=<resubmission type>] \
[REPEAT_END=<resubmission end date and time>] \
[START=<date>] \
[IMPLICIT=<type of concurrent request> \
[<parameter 1> ... <parameter n>]
```

For parameters that follow the CONCURRENT parameter and include spaces, enclose the parameter argument in double quotes, then again in single quotes. Oracle Application Object Library requires this syntax because it parses the argument string twice. For example, to pass this argument to a program:

This is an example

pass this argument through CONCSUB:

```
'"This is an example"'
```

Example

Here is an example of the command to run CONCSUB:

```
$ CONCSUB APPS/APPS \
SYSADMIN \
"System Administrator" \
SYSADMIN \
WAIT=N \
CONCURRENT \
FND \
```

```
FNDFMRTC \
PROGRAM_NAME=' "Register Custom Tables Weekly" ' \
REPEAT_INTERVAL=7 \
REPEAT_INTERVAL_UNIT="DAYS" \
REPEAT_INTERVAL_TYPE="START" \
START=' "08-JUN-96 23:55:00" '
CGL
APPLSYS
ALL
CGL
```

Parameters

The following entries explain the required and optional parameters for submitting a concurrent program with CONCSUB. Default values are listed to the right.

username/password (required) (no default)

The ORACLE username and password that provides access to the data that your program uses.

responsibility application short name (required) (no default)

The application short name of the responsibility whose concurrent processing options you want to use.

responsibility name (required) (no default)

The name of your responsibility. If the name of your responsibility includes spaces, enclose that name in double quotes.

username (required) (no default)

The uppercase username of the application user whose concurrent processing options you want to use.

WAIT (optional) N

A flag that indicates whether to wait for the submitted request to complete. If you leave this parameter out, the default value of N makes CONCSUB return you to the operating system prompt without waiting for your request to complete.

Set WAIT=Y to have CONCSUB check the request status every 60 seconds and return you to the operating system prompt when your request is completed. You

can also enter an integer value for a number of seconds, as in WAIT=30, for CONCSUB to check for request completion every <number> seconds.

Attention: Using WAIT=Y or WAIT=<number> requires that your request completes before CONCSUB returns you to the operating system. If the concurrent manager is down, your CONCSUB process waits indefinitely until the concurrent manager is started and the request completes

CONCURRENT (required)

(n/a)

A flag that separates the program-specific parameters from the operating system parameters.

program application short name (required)

(no default)

The application short name of your concurrent program.

program name (required)

(no default)

The uppercase name of your program. It must be the short name that you enter in the Concurrent Programs window when defining a concurrent program.

PROGRAM_NAME (optional)

(no default)

A descriptive name for your program. The program field on the View Requests form displays this as the user-friendly program name. The concurrent program short name passed to CONCSUB is often hard for end users to understand, so the PROGRAM_NAME parameter allows you to pass a more easily remembered name for your concurrent program. If you do not specify a PROGRAM_NAME, the View Requests form displays the user-friendly program name specified in the Concurrent Programs window.

You may also use the PROGRAM_NAME parameter to indicate the batch that your request processes for programs that process a set of data, where there could be several requests for a given program that are active at the same time.

REPEAT_TIME (optional) (no default)

The time of day to resubmit the request. The format for the time is HH24:MI or HH24:MI:SS. For example, REPEAT_TIME=14:30 resubmits your request daily at 2:30 p.m.

Attention: Do not use REPEAT_TIME with other resubmission parameters except for the optional parameters REPEAT_END and START.

REPEAT_INTERVAL (optional) (no default)

The interval between resubmission (a positive integer or real number). Use this parameter along with REPEAT_INTERVAL_UNIT to specify the time between resubmissions.

REPEAT_INTERVAL_UNIT (optional) DAYS

The unit of time used for the interval between resubmissions. The available units are MINUTES, HOURS, DAYS or MONTHS. Use this parameter along with REPEAT_INTERVAL to specify the time between resubmissions. For example, setting REPEAT_INTERVAL=12 and REPEAT_INTERVAL_UNIT=HOURS resubmits your request every twelve hours.

Attention: Do not use REPEAT_INTERVAL and REPEAT_INTERVAL_UNIT with REPEAT_TIME.

REPEAT_INTERVAL_TYPE (optional) START

Whether to time the resubmission interval from the requested start time of the request or from its completion. Set this parameter either to START or END.

Attention: Use REPEAT_INTERVAL_TYPE only if you use REPEAT_INTERVAL.

REPEAT_END (optional) (no default)

The date and time to stop resubmitting the concurrent request. Use one of the following for the format of the end date:

```
'"DD-MON-RR HH24:MI:SS"' (as in '"07-APR-98 18:32:05"')
```

or

```
'"DD-MON-RRRR HH24:MI:SS"' (as in '"07-APR-1998 18:32:05"')
```

Note that because this date format includes a space, you must enclose the date in double quotation marks and single quotation marks. You can also specify just the date:

```
'DD-MON-RR'
```

or

```
'DD-MON-RRRR'
```

START (optional)

(current time)

A start date and time for your program in this format:

```
'"DD-MON-RR HH24:MI:SS"' (as in '"07-APR-98 18:32:05"')
```

Because this date format includes a space, you must enclose the date in double quotation marks and single quotation marks. If you do not specify a start time, your program submits immediately and is processed by the next available concurrent manager.

IMPLICIT (optional)

NO

Whether to show this concurrent request on the View Requests form. Specify NO, YES, ERROR or WARNING. The value IMPLICIT=NO allows the request to appear on the View Request form.

The value IMPLICIT=YES means that only the System Administrator's privileged View Concurrent Requests form displays this request. Use this value if the request is not interesting to the user.

Specify IMPLICIT=ERROR or IMPLICIT=WARNING, respectively, if you want the request to appear only if it fails or completes with warnings.

REPEAT_DAYS (optional)**(no default)**

The number of days after which to repeat the concurrent request, calculated from the last requested start date. The number can be a positive integer or real number. For example, REPEAT_DAYS=1.5 resubmits your request every 36 hours.

Attention: Do not use REPEAT_DAYS with other resubmission parameters except for the optional parameters REPEAT_END and START.

Suggestion: REPEAT_DAYS will become obsolete in a future release. You may therefore want to use REPEAT_INTERVAL, REPEAT_INTERVAL_TYPE and REPEAT_INTERVAL_UNIT instead of REPEAT_DAYS.

parameter 1... parameter *n* (optional)**(no default)**

Your program-specific parameters. If a parameter includes spaces, enclose that parameter in double quotes, then in single quotes. If a parameter contains a double quotation mark as part of the argument, precede that mark with a backslash [\\].

Symbols

\$APPL_TOP
 creating directory, 2-12
\$APPLCSF
 specifying directory, 3-38
\$APPLLOG
 specifying directory, 3-38
\$APPLMSK, 4-24
\$APPLOUT
 specifying directory, 3-38
\$APPLTMP
 setting, 3-38
\$REPORTS25_TMP
 setting, 3-38

A

AD administration utility (adadmin)
 running, 5-7
AD Configuration Utility, 5-61
adadmin.log, 5-8, 5-9
adaimgr.log, 3-6, 3-41, 4-6
adctrl.log, 5-2, 5-3
addbxxx.sql script, 2-22
adident, 5-60
Administration Utility
 log file, 5-8
Administration utility (adadmin)
 log file, 5-9
admrgpch utility, 5-38
adodfcmp utility, 5-54
adovars.env file
 editing, 2-20
adrelink.log, 4-6, 5-48
adsetenv, 2-32
adsetup, 2-21, 5-37
adsysapp.sql script, 2-22
adsyxxx.sql script, 2-22
adunload.log, 4-6, 5-44
adutconf.sql, 5-61
adworkxxx.log, 3-42, 4-6
ADXANLYZ.sql, A-20
ADXCKPIN.sql, A-22
ADXGNPIN.sql, 5-33, A-21
ADXGNPNS.sql, 5-33
afsamplerd.htm, 4-17
afsamples.htm, 4-17
airfinit.rf9, 3-41
airfmain.rf9, 3-42
airfprod.rf9, 3-41
airftask.rf9, 3-42
airftodo.rf9, 3-42
ANALYZE command, A-20
APPL_TOP variable
 defined, 4-18
 setting, 2-18
applcust.txt, 5-29
applmgr login
 creating, 2-9
 profile file, 2-10
 umask value, 2-11
applptch.txt, 5-32, 5-42
APPS schema
 and custom objects, 4-34
 and custom schemas, 4-34
APPS schema(s)
 compiling, 5-12

- validating, 5-12
- AutoInstall
 - copying products, 3-19
 - database parameters screen, 3-23
 - deleting products, 3-20
 - erasing screen information, 3-12
 - error correction, 3-43
 - error messages, 4-5
 - acceptable, 3-45
 - installation actions, 3-15
 - log file, 3-6, 3-42, 4-5
 - main menu, 3-10
 - monitoring, 3-40
 - parallel mode, 3-34
 - product detail screens, 3-18
 - questions, 3-5
 - restart files, 3-42
 - screens, using, 3-12
 - select modules screen, 3-22
 - select products screen, 3-13
 - setup script, 2-21
 - shutting down, 5-6
 - starting and stopping, 3-2
 - tasks screen, 3-27
- AutoInstall Controller
 - log file, 5-3
 - restarting a worker, 5-4, 5-5
 - restarting the manager, 5-5
 - running, 5-1
 - shutting down AutoInstall, 5-6
 - worker status, 5-3
- AutoPatch
 - driver file format, 5-39
 - error messages, 5-28
 - handling of customizations, 5-29
 - log files, 5-30
 - patch history file, 5-42
 - questions, 5-25
 - running, 5-23
 - test mode, 5-35

B

- Bourne shell, 2-9
- Bug patch

- format, 5-21
- integrated, 5-23
- Merge utility, 5-38
- readme.txt file, 5-21

C

- C shell, 2-9
- Certified Oracle products, 1-18
- character set conversion
 - when unloading, 5-46
- Codeset conversion
 - after unloading, 5-53
- CONCSUB
 - example, B-13
 - parameters, B-14
 - syntax, B-13
- Concurrent managers
 - directory privileges when starting, A-14
 - log files, A-13
 - restarting, A-12
 - starting, 4-25
 - startup parameters
 - list of, A-11
 - passing, 4-21
- Concurrent processing
 - altering file privileges, 4-24
 - custom applications, B-6
 - Host language programs, B-11
 - memory requirements, 1-4
 - Oracle Tools programs, B-7
 - Pro*C programs, B-8
 - setting up, 4-20
 - submitting a custom program, B-12
- Concurrent processing (parallel)
 - applmgr logins, 2-9
 - creating common directory, 2-13
 - environment variables, 4-28
 - memory requirements, 1-4
 - ping in \$PATH, 2-10
 - product file installation, 3-31
- Concurrent programs
 - host language, B-11
- CPU requirements, 1-1
- Custom APPS% schemas

- name conflicts, 2-14
- Customizations, 2-14
 - custom schemas, 2-27
 - integrating with APPS, 4-34
 - reapplying after upgrade, 4-34
 - triggers, 2-27
 - verifying compatibility, 1-20

D

- Database
 - preparation, 2-14
- Database Access Descriptor (DAD), 4-11
- database preparatory scripts, 2-22
- DB_BLOCK_SIZE parameter, 1-21
- dbmerged.drv file, 5-39
- dcptest, 4-28
- Demonstration product groups
 - unloading files, 3-35
- Dependent products
 - copying, 3-19
 - disk space for, 1-5
 - installing fully, 3-15
 - list of, 1-6
 - ORACLE user ID, 3-18
- Developer/2000
 - version, 1-19
- Directories
 - \$APPL_TOP, 2-12
 - custom applications, B-1
 - in \$PATH, 2-9
 - installation, 2-19
 - multiple product groups, 4-20
 - product, 3-18
 - subdirectory types, A-37
- Disk space
 - recovering, A-36
- Disk space requirements, 1-5
 - AutoInstall generated files, 1-16
 - deleting old files, 1-17
 - dependent products, list of, 1-6
 - installing products on separate disks, 1-17
 - language files, 1-14
 - log and output files, 1-15
 - temporary files, 1-15

- variations in file sizes, 1-13
- DUAL table
 - verifying
 - through adadmin, 5-12
 - through AutoInstall, 3-29

E

- .env environment file, A-39
- Environment file
 - regenerating, 5-16
 - running, 4-27
 - specifying name, 3-29
- Environment variables
 - list of, A-39
- Error messages
 - acceptable, 3-45
 - reviewing, 4-5
- Executable programs
 - compression utilities for, A-37
- Executable programs (printing), A-16
- EXMAIN.c, B-8
- Exporting the database, A-26
- EXPROG.c, B-8

F

- fcpl_login, B-12
- fcpwrt(), B-6
- file types, A-37
- Flexfield data
 - compiling
 - through adadmin, 5-12
- Flexfields data
 - compiling
 - through AutoInstall, 3-30
- FND_INSTALL_PROCESSES table, 5-2
- fndcpesr, B-11
- FNDIFS, 4-29, 4-31
- FNDNAM variable, 4-18
- fopen(), B-6
- Force argument (relinking), 5-50
- Form generation
 - errors or warnings during, 3-48
- Forms

generating, 3-30, 5-18

G

glogin.sql, 1-23

Grants

recreating

through adadmin, 5-12

through adodfcmp, 5-56

GWYUID variable, 4-18

H

Host language

concurrent programs, B-11

I

Importing the database, A-27

Initialization parameters

changing, 1-21

checking through AutoInstall, 3-28, 3-35

for installation, 1-21

language, 1-22, 4-6

Installation directory

unloading, 2-19

Installation Update documents, xi

Interim tablespace

recommended size, 1-29

J

Java archive (JAR) files

downloading, 4-17, 4-19

Java Runtime Environment

installing, 2-6

JDBC

installing, 2-6

version, 1-19

JDBC, configuring for Oracle Self-Service

Applications, 4-10

K

Korn shell, 2-9

L

Language installation

disk space requirements, 1-14

NLS language codes, A-45

Server language parameters, 1-22, 4-6

short names, A-45

specifying language, 3-10

territories, A-42

unloading language files, 5-46

listener.ora, 4-29

merging contents, 4-30

Localization installation

unloading localization files, 2-21

Localizations

database space, 1-29

installation screen, 3-22

unloading installation directory, 2-21

Log files, AutoInstall

designating, 3-6

manager file, 3-41

reviewing, 4-5

worker files, 3-42

Log files, product

custom applications, B-6

disk space requirements, 1-15

location of, A-13

multiple product groups, 2-13, 4-20

Log files, utilities

Administration (adadmin), 5-8

AutoInstall Control (adctrl), 5-2

control (adctrl), 5-3

ODF comparison (adodfcmp), 5-58

relink (adrelink), 5-48

unload (adunload), 5-44

login.sql, 1-23

M

make utility, 5-48

mcs utility, A-37

Memory requirements, 1-3

pinning packages and sequences, 4-36

SGA size, 1-3

Message dictionary, B-5

- Message files
 - generating, 3-30
 - printing, B-5
 - regenerating, 5-18
- Migrating Oracle Applications, A-24
- MRC
 - installing, A-2
- MultiOrg
 - converting a standard database to, 5-14
- Multiple product groups
 - log and out directories, 2-13, 4-20
- Multiple product installation groups
 - effect on database size, 1-28
- Multiple Reporting Currencies
 - installing, A-2
- Multiple sets of books
 - installing, 3-16
 - upgrading, 3-20

N

- Net8
 - configuring, 4-29
 - version, 1-19
- network configuration files, 4-29
- NLS territory name, 3-10
- NLS_DATE_FORMAT, 1-21, 1-22
- NLS_DATE_LANGUAGE, 4-8
- NLS_LANG variable, 2-18, A-42
- NLS_NUMERIC_CHARACTERS, 1-21, 1-22

O

- Object description file (ODF), 5-54
- ODF (object description file), 5-54
- ODF comparison
 - parameters, 5-56
 - running manually, 5-54
- Online report viewing
 - setting up, 4-29
- Optimizer mode, 1-23
 - cost-based optimization, A-20
 - parameter for, 1-21
- Oracle Alert
 - environment variable for linking, 3-39

- Oracle Application Object Library
 - changing password, 4-36
 - concurrent processing, B-6
 - development environment, B-1
 - message dictionary, B-5
 - reference material, B-1
- Oracle Applications System Administrator's Guide, xii
- Oracle Consulting Services, xii
- Oracle Financials Country-Specific Installation Supplement, xii
- Oracle Forms
 - and Java client applet, 4-19
 - Cartridge Handler, 4-14
 - Metrics Server, 4-16
 - version, 1-19
- Oracle Forms Server listener, 4-19
- Oracle Reports
 - printer drivers, A-18
 - regenerating, 5-19
 - regenerating files, 3-30
 - version, 1-19
- Oracle Server
 - version, 1-19
- Oracle Server requirements, 1-18
 - certified versions, 1-18
 - initialization parameters, 1-21
- Oracle Support Services, xiii
- Oracle Tools
 - concurrent programs, B-7
- ORACLE user ID
 - changing password, 3-25, A-22
 - dependent product, changing, 3-18
 - product, 3-14, 3-25
- Oracle Web Application Server
 - version, 1-20
- ORACLE_HOME variable
 - verifying, 2-10
- ORACLE_SID variable
 - verifying, 2-10
- Oracle8 ConText Cartridge
 - version, 1-19
- Output files
 - custom applications, B-6
 - disk space requirements, 1-15

location of, A-13

P

Patch

- driver files, 5-22
- format, 5-21
- integrated, 5-23
- Merge utility, 5-38
- readme.txt file, 5-21

PATH environment variable, 2-9

Pinning scripts, 4-36

PL/SQL

- version, 1-19

PL/SQL agent

- configuring, 4-12

Pre-AutoInstall mode, 5-37

Preparing the database, 2-14

Printing

- executable programs
 - creating, A-16
 - registering, A-19
- Oracle Reports drivers, A-18
- printer names, A-15
- reference material for, A-15
- setting up, 4-32, A-15
- subroutine options, A-15

Pro*C

- linking
 - immediate concurrent programs, B-10
 - spawned concurrent programs, B-10
- program library, B-9
- program testing, B-11
- version, 1-20

Product Detail Screens, 3-18

Product groups, 2-14

Product Update Notes, xii

Profile file

- for applmgr, 2-10

R

Release Notes, xi

Relinking

- force argument, 5-50

- log file, 5-48

- manually, 5-48

- target file, 5-51

- through adadmin, 5-17

- through AutoInstall, 3-29

- through AutoPatch, 3-46

Report Review Agent, 4-29, 4-31

- setting up, 4-29

Reports

- generating, 5-19

- regenerating, 3-30

Reserved words, A-46

Rollback segments

- recommended size, 1-25

S

sample.mk, B-10

Screens

- for AutoInstall, 3-12

Send us your comments, xv

Separating product groups, 2-14

serverHost cartridge parameter, 4-17

serverPort cartridge parameter, 4-17

SGA size, 1-3

Shared pool

- effect on SGA, 1-3

- parameter value, 1-21

- pinning packages and sequences, 4-36

Short names

- languages, A-45

Signon screen, 4-32

Sizing factor

- 100% size table, 1-26

- setting during installation, 3-26

SQL*Loader

- version, 1-19

SQL*Plus

- version, 1-19

SQL*Plus options, 1-23

Starting Applications, 4-31

startmgr script

- customizing, 4-23

- editing, 4-22

strip utility, A-37

- Synonyms
 - recreating
 - through adadmin, 5-12
 - through adodfcmp, 5-56
- SYS.DUAL, 2-15
- SYS.DUAL table
 - creating, 3-29
- SYSTEM tablespace
 - recommended size, 1-24

T

- Tablespaces
 - requirements, 1-24
 - specifying during installation, 3-26
- Technical Reference manuals, xii
- Temporary files
 - creating directories for, 2-13
 - disk space requirements, 1-15
 - location of, A-13
 - setting directory variables, 3-38
- Temporary tablespaces
 - recommended size, 1-25
 - specifying during installation, 3-26
- Territory names, A-42
- tnsnames.ora, 4-29
- TOP variables
 - custom applications, B-3
- TWO_TASK variable
 - defined, 4-18
 - verifying, 2-10

U

- umask command, 4-24
- UNIX shell
 - applmgr, 2-9
- Unload Utility
 - log file, 5-44
- Unloading
 - AutoInstall files, 2-19
 - character set conversion, 5-46
 - demonstration product groups, 3-28
 - language files, 5-46
 - product files, 3-28, 3-31, 3-35

- with unload utility, 5-44
- Upgrade manual, xi
- Upgrade steps
 - category 1, 2-17
 - category 2, 2-25
 - category 3, 2-27
- User's Guide, xii

V

- Virtual paths
 - configuring, 4-9

