Oracle® Applications

Maintaining Oracle Applications

Release 11*i*

October 2000

Part No. A87326-01

ORACLE®

Maintaining Oracle Applications, Release 11*i*

Part No. A87326-01

Release 11*i*

Author: Yun Shaw

Contributing Authors: Subash Chadalavada, Michael Cole, Carole Eubanks, Michael Fiore, Billy Greene, Rick Lotero, Kent Noble, Roger Sanders

Contents

Send Us Your Comments ix

Preface xi

Audience xi
Structure xii
Information for Windows NT Users xiii
Software Requirements xiii
Registry Variables and Environment Variables xiii
Conventions xiv
Getting Help xv
Oracle Consulting Services xv
Oracle Support Services xv
Related Documents xvi

1 About AD Utilities

What Are the AD Utilities? 1-1
AD Administration (adadmin or adadmin.exe) 1-1
AutoUpgrade (adaimgr or adaimgr.exe) 1-2
AutoPatch (adpatch or adpatch.exe) 1-3
AD Controller (adctrl or adctrl.exe) 1-3
AD Configuration (adutconf.sql) 1-3
AD File Identification (adident or adident.exe) 1-4
AD Splicer (adsplice or adsplice.exe) 1-4
File Character Set Conversion (adncnv or adncnv.exe) 1-4

ODF Comparison (adodfcmp or adodfcmp.exe) 1-4
AD Relink (adrelink.sh) 1-4
AD Merge Patch (admrgpch or admrgpch.exe) 1-4
DataMerge (addmimp or addmimp.exe) 1-5
AD Run SQL (adurs or adurs.exe) 1-5
AD Rebase (adrebase.exe) 1-5
Verify Apps (advrfapp.sql) 1-5
License Manager (LicenseMgr) 1-5
How They Work 1-5
Running AD Utilities 1-6
Getting Ready 1-6
Stopping and Restarting 1-8
Responding to Prompts 1-9
Configuration and Environment Files 1-15
Creating an Environment File 1-15
AD Timing Report 1-20

2 AD Administration

What is AD Administration? 2-1
Database Tasks 2-1
File System Tasks 2-2
Running AD Administration Interactively 2-2
Using the Main Menu 2-3
Running AD Administration in Non-Interactive Mode 2-16

3 AutoUpgrade

What is AutoUpgrade? 3-1
Using AutoUpgrade 3-2
How AutoUpgrade Works 3-2
Running AutoUpgrade in a Multi-server Environment 3-3
Running AutoUpgrade 3-3
Starting and Stopping AutoUpgrade 3-4
Responding to Prompts 3-4
Restarting AutoUpgrade 3-5
Using AutoUpgrade Menus and Screens 3-7

Main Menu 3-7
About the AutoUpgrade Screens 3-8
Select Products Screen 3-9
Product Detail Screens 3-14
Select Modules Screen 3-17
Database Parameters Screen 3-18
Tasks Screen 3-22
Running the Upgrade 3-27
Monitoring AutoUpgrade 3-29
Manager and Worker Messages 3-29
Log and Restart Files 3-30
Correcting AutoUpgrade Errors 3-32
Acceptable Errors 3-34
After Running AutoUpgrade 3-37

4 AutoPatch

What is AutoPatch? 4-1
How It Works 4-2
Patch Format 4-5
Running AutoPatch 4-7
Starting AutoPatch 4-8
Responding to Prompts 4-9
AutoPatch Messages 4-11
After Running AutoPatch 4-14
Running AutoPatch Again 4-17
Controlling AutoPatch Behavior 4-19
Test Mode 4-19
Pre-install Mode 4-20
Non-interactive Mode 4-21
Using the options Parameter 4-25
Using the AD Merge Patch Utility 4-27
Patch Driver Files 4-27
Patch History File 4-30
Patch Summary File 4-31
Adding Translations 4-32

Applying Java Patches 4-32
Java Release Infrastructure 4-35

5 Other AD Utilities

AD Controller (adctrl) 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 5-6
Shutting Down the Managers 5-6
Controlling the Number of Workers 5-7
AD Configuration (adutconf.sql) 5-7
AD File Identification (adident) 5-8
AD Splicer (adsplice) 5-9
Splicing New Products 5-9
Editing newprod.txt 5-11
File Character Set Conversion (adncnv) 5-13
ODF Comparison (adodfcmp) 5-14
Running ODF Comparison 5-14
AD Relink (adrelink.sh) 5-19
Running adrelink.sh 5-20
Relinking with Debug 5-23
Target File 5-23
Backup Options for adrelink.sh 5-25
AD Merge Patch (admrgpch) 5-26
DataMerge (addmimp) 5-27
AD Run SQL (adurs) 5-29
AD Rebase (adrebase.exe) 5-30
License Manager (LicenseMgr) 5-31

6 Maintaining the Applications Database

Analyzing the Database 6-1
Pinning Packages and Sequences 6-2
Changing Oracle Applications Passwords 6-4

Migrating the Oracle Applications Database 6-5
Preparing to Migrate the Database 6-7
Export the Oracle Database 6-12
Import the Oracle Database 6-14
Finishing the Migration 6-16
Database Initialization Parameters 6-20
General init.ora Parameters 6-20
Fixed SGA init.ora Parameters 6-23
Cost-Based Optimization (CBO) init.ora Parameters 6-26
Special Features 6-30
Rollback Segments 6-31
Product Tablespace Requirements 6-31

7 Maintaining the Applications File System

Migrating the Oracle Applications File System 7-1
Moving Files on the Same Machine 7-1
Preparing to Migrate the File System to Another Machine 7-4
Unload Oracle Applications Product Files 7-4
Completing the Migration 7-4
Recovering Disk Space 7-5
Server Processes 7-7
Starting Server Processes 7-7
Reserved Words 7-11

Index

Send Us Your Comments

Maintaining Oracle Applications, Release 11*i*

Part No. A87326-01

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

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

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

- FAX - 650.506.1113 Attn: Oracle Applications Release Group
- email: apps_relgrp_us@oracle.com
- postal service:
  Oracle Corporation
  Oracle Applications Release Group
  M/S 3op4
  500 Oracle Parkway
  Redwood Shores, CA 94065 U.S.A

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

Preface

Maintaining Oracle Applications provides instructions for maintaining the Applications file system and database, and directions on using the Applications DBA (AD) utilities, the main tools for these tasks. In addition to maintaining Applications, the AD utilities are also used for installing, patching, and upgrading Oracle Applications products.

Audience

If you are responsible for maintaining the file system or database of Oracle Applications or for installing or upgrading Oracle Applications, it is important that you read and understand the information in *Oracle Applications Concepts, Installing Oracle Applications,* and *Upgrading Oracle Applications*. *Concepts* explains the technology, architecture, and terminology used in *Installing* and in *Upgrading.* The following people typically use these books:

- Database Administrator

  Installs, upgrades and configures the ORACLE database and maintains database access controls. This 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. The system administrator also installs packaged applications environments and converts 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

Maintaining Oracle Applications contains these chapters. It also contains an index.

Chapter 1	Discusses the suite of Oracle Applications utilities, commonly called AD utilities. Briefly describes each utility, suggests when to use it, and lists common elements.
Chapter 2	Discusses the AD Administration utility and describes how to use it.
Chapter 3	Discusses the AutoUpgrade utility and describes how to use it.
Chapter 4	Discusses the AutoPatch utility and describes how to use it.
Chapter 5	Discusses the other AD utilities and describes how to use them.
Chapter 6	Contains information about maintaining various aspects of the Oracle Applications database.
Chapter 7	Contains information about maintaining various aspects of the Oracle Applications file system.

Information for Windows NT Users

While UNIX and Windows NT users will find little difference (aside from the obvious command line syntax) in the way the instructions in this manual affect their use of the AD utilities, some of the information bears further explanation. This section describes some general differences that affect the way the AD utilities work on the Windows NT platform.

Software Requirements

In Release 11i and beyond, additional tools are required to maintain Applications on Windows NT. In the new "UNIX-like" model, relinking (required by patches or upgrades) is done at the customer site, thus allowing a higher level of granularity of patching.

> **Note:** Prior to 11i, all NT executable programs were built by Oracle and shipped to the customer.

The following software must be installed at the customer site:

- Microsoft Visual C++ version 6.0 + Service Pack 3 or higher (http://www.microsoft.com)
- MKS Toolkit version 6.1a or higher (http://www.mks.com)
- gnu make (shareware) version 3.77 or higher (http://www.gnu.org)

Registry Variables and Environment Variables

The NT registry is the primary data store for the NT system environment. Applications programs (including the AD utilities) look in the registry to determine the value of Applications variables. However, for the sake of convenience, you can use environment variables, and registry variables are superseded by environment variables of the same name.

The examples shown in this manual assume a command prompt syntax and that environment variables are used (since registry variables cannot be typed directly on the command line). If an example makes reference to an environment variable %<var_name>%, first check to see if it is set in the environment:

```
C:\> echo %<var_name>%
```

For example, if your <var_name> is GL_TOP, and echo %GL_TOP% returns a valid pathname, you can use GL_TOP on a command line:

```
C:\> cd %GL_TOP%\admin\sql
```

If <var_name> is not defined in the environment, you have two options:

Run adregenv.exe:

1. Run adregenv.exe to copy all the registry variables under the <APPL_CONFIG> key into a .cmd file (called apps.cmd in the APPL_TOP directory).
2. Run apps.cmd to create environment variables that correspond to the registry variables.

The environment variable <var_name> can now be expanded on the command line using %<var_name>%.

Run regedt32.exe:

Use regedt32.exe to examine the value of <var_name> and substitute this value on the command line.

Conventions

We recommend that you review the following conventions used in this manual.

Convention	Meaning
Monospace text	Represents command line text. Type this text exactly as shown.
< >	Text enclosed in angle brackets represents a variable. Substitute an appropriate value for the variable text. Do not type the brackets.
[ ]	Square brackets enclose optional items or indicate a function key. 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 or \directory	A slash before a directory name indicates that it is a subdirectory. The path name may be either uppercase or lowercase.
$ or C:\>	Represents the command prompt. Your prompt may differ.
\	In examples of commands you type online, a backward slash at the end of a line signifies that you must type the entire command on one line. *Do not type the backslash.*

Special notes alert you to particular information within the body of the manual.

Additional Information:	Refers you to portions of this manual, another manual, or the online documentation. All references to other manuals refer to the most recent version of that manual, unless otherwise noted.
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 of actions which, if not carried out properly, could be damaging or destructive to your operations.

Getting Help

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

Oracle Consulting Services

Oracle Consulting Services can help:

- determine machine size and database size required by Oracle Applications
- install or upgrade Oracle Applications
- implement Oracle Applications
- customize Oracle Applications
- 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://metalink.us.oracle.com/ offers registered Oracle *MetaLink* customers self-service support technologies, available 24 hours at day, 7 days a week. Oracle *MetaLink* provides information such as 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 of each server
- 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 or maintaining
- 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 AutoUpgrade, AutoPatch, or AD Administration parallel workers
- the output of the AD Configuration utility, contained in the adutconf.lst file
- the output of relevant .log files

Related Documents

All Release 11*i* documentation is included on the *Oracle Applications Document Library* CD, which is supplied in your CD Pack. You can download soft-copy documentation from http://docs.oracle.com. Or, you can purchase hard-copy documentation from the Oracle Store at http://oraclestore.oracle.com.

Note: All titles used throughout this manual refer to Release 11*i*, unless otherwise noted. Documentation for pre-upgrade steps generally refers to the manuals associated with the release you are upgrading from.

Note: This manual, and any other documentation associated with this release, was current as of the time it was published and released. However, we make enhancements to Oracle Applications products and respond to user needs on a continuing basis. It is important, therefore, that you always check Oracle*MetaLink* for the most up-to-date information about Oracle Applications before you begin your upgrade.

Specific documentation that you may need in addition to this manual includes:

If you are looking for...	See these documents...
New Application features	*Oracle Applications Product Update Notes* *Oracle Applications User's Guides* *Oracle Applications Implementation Manuals* *Multiple Organizations in Oracle Applications* *Multiple Reporting Currencies in Oracle Applications*
Database information	*Oracle8i Backup and Recovery* *Oracle8i Reference Guide* *Oracle8i Tuning Guide* *Oracle8i National Language Support Guide*
Installation or Upgrade information	*Installing Oracle Applications* *Upgrading Oracle Applications* *Oracle Applications Installation Updates* <for your platform>
Information about custom development	*Oracle Applications Coding Standards* *Oracle Applications Developers' Guide*
Other information	*Oracle Applications Concepts* *Oracle Applications System Administrator's Manual* *Oracle Self-Service Web Applications Implementation Manual* *Oracle Workflow Guide* *Oracle Applications Character Mode to GUI Menu Path Changes*

1

About AD Utilities

This chapter contains an overview of the Oracle Applications utilities, commonly known as the AD Utilities. It briefly describes each utility and explains how it is used and when to use it. It also contains information on common utility procedures and common configuration and environment files. This chapter contains the following topics:

- What Are the AD Utilities?
- How They Work
- Running AD Utilities
- Configuration and Environment Files

What Are the AD Utilities?

AD Utilities are a group of tools designed to install, upgrade, maintain, and patch a specific set of products contained in a given release of Oracle Applications.

AD Administration (adadmin or adadmin.exe)

AD Administration (adadmin) performs maintenance tasks on an installed Oracle Applications system to ensure that it runs smoothly. It can also be used to complete some runtime tasks during an installation or upgrade, or at any time thereafter.

The tasks performed with this utility fall into two categories: database and file system. Just like AutoUpgrade and AutoPatch, AD Administration can run parallel workers for most database tasks and some file system tasks.

On the AD Administration screens, you can choose tasks from two menus:

Maintain Database Objects Menu

You can perform the following tasks from this menu:

- Validate APPS schema(s)
- Compile APPS schema(s)
- Recreate grants and synonyms for APPS schema(s)
- Compile flexfield data in AOL tables
- Maintain multi-lingual tables
- Check DUAL table
- Maintain Multiple Reporting Currencies schema(s)
- Convert to MultiOrg
- Convert to Multiple Reporting Currencies

Maintain Applications Files Menu

You can perform the following tasks from this menu:

- Create Applications environment file
- Relink Applications programs
- Copy files to destinations
- Verify files necessary for runtime
- Generate message files
- Generate form files
- Generate report files
- Generate graphics files
- Generate product jar files

AutoUpgrade (adaimgr or adaimgr.exe)

AutoUpgrade expands the functionality of an Oracle Applications upgrade. You run this utility after you complete the basic installation tasks with Rapid Install — installing the Oracle8 technology stack and file system — to perform tasks such as updating database objects, adding a localization, or converting to Public Sector financials.

On the AutoUpgrade screens, you can:

- Select products to license
- Specify product details
- Specify tablespace names, ORACLE schema names, and sizing factors for AutoUpgrade to use when upgrading your Oracle Applications objects in the database
- Select tasks that pertain to your installation from the complete set of available AutoUpgrade tasks
- Run the selected tasks

AutoPatch (adpatch or adpatch.exe)

AutoPatch is used to apply individual patches, mini-packs, or maintenance packs. A *mini-pack* (known as a patch set in previous releases) is a collection of individual patches for a product, while a *maintenance pack* (known as a release update in previous releases) is a collection of *mini-packs* for all Applications products. For example, you install Oracle Applications Release 11*i* using Rapid Install. Any patches that accompany that release are installed using AutoPatch.

In addition to maintaining existing products, AutoPatch is used during the process of additional tasks such as adding a language or a new product that was not a part of the base release.

AD Controller (adctrl or adctrl.exe)

AD Controller is used in conjunction with AD Administration, AutoUpgrade, and AutoPatch to determine the status of AD Utilities workers and restart failed AD Utilities tasks.

AD Configuration (adutconf.sql)

This SQL script reports standard information about the installed configuration of Oracle Applications. It generates a file called adutconf.lst that contains the following:

- information about the product group
- whether MultiOrg is installed
- whether Multiple Reporting Currency (MRC) is installed

- information about all installed products, including those that are shared and dependent
- information on all registered schemas
- the base language and other installed languages

AD File Identification (adident or adident.exe)

With AD File Identification, you can identify the version of one or more Oracle Applications files. This utility is useful for collecting information about your site when contacting Oracle Support Services.

AD Splicer (adsplice or adsplice.exe)

Use AD Splicer to install products that were not included in the current Oracle Applications release. It modifies your APPL_TOP and database so AutoPatch and AD Administration can recognize the new product as being valid.

File Character Set Conversion (adncnv or adncnv.exe)

File Character Set Conversion is used to convert the character set of unloaded files. Conversion may be necessary for any text files shipped by Applications, including SQL*Plus scripts, PL/SQL scripts, loader files, driver files, ODF files, and HTML files. File character set conversion for files is normally done automatically by the Rapid Install and by AutoPatch.

ODF Comparison (adodfcmp or adodfcmp.exe)

Use ODF Comparison to compare the data model of your database with the standard data model from the current release of Oracle Applications.

AD Relink (adrelink.sh)

Use this utility to relink Oracle Applications executable programs with the Oracle8 Server product libraries.

AD Merge Patch (admrgpch or admrgpch.exe)

This utility merges multiple AutoPatch compatible patches into a single integrated patch. It will not merge patches of different releases or other incompatibilities.

DataMerge (addmimp or addmimp.exe)

Similar to the Oracle8i import utility, this utility runs during an upgrade to import seed data needed by Oracle Applications.

AD Run SQL (adurs or adurs.exe)

Use this utility to run a specified SQL file in a special SQL handling mode used by AutoUpgrade and AutoPatch.

AD Rebase (adrebase.exe)

The AD Rebase utility is an NT-only utility which optimizes memory utilization of Applications, RDBMS and Tools executable programs.

Verify Apps (advrfapp.sql)

This script validates the APPS schema(s) when that option is selected from the Database Objects menu of AD Administration (it can also be run manually). It also checks for certain conditions that are not optimal, even though they do not produce fatal problems. An output file, <APPS schema name>.lst, is produced for each APPS schema.

License Manager (LicenseMgr)

When you want to add additional products or languages to your Oracle Applications installation, you use the Oracle Applications License Manager to license these new additions.

How They Work

You can use the AD Utilities for a variety of tasks. This section provides a chart outlining when a particular utility would be used, preparation steps required for all AD Utilities, and the common steps for running the three most-often used utilities (AD Administration, AutoUpgrade, and AutoPatch), which call many of the other utilities for specific tasks.

The chart indicates when you would utilize a utility — during an Installation, Upgrade, Maintenance, or Patching of Oracle Applications.

AD Utility	Install	Upgrade	Maintain	Patch
AD Administration	✓	✓	✓	✓
AutoUpgrade	x	✓	x	x
AutoPatch	✓	✓	✓	✓
AD Controller	✓	✓	✓	✓
AD Configuration	✓	✓	✓	x
AD File Identification	x	x	✓	✓
AD Splicer	✓	x	x	x
File Character Set Conversion	✓	✓	✓	✓
ODF Comparison	x	✓	✓	✓
AD Relink	x	x	✓	✓
AD Merge Patch	x	x	x	✓
Data Merge	x	✓	x	x
AD Run SQL	x	✓	✓	✓
AD Rebase	x	✓	✓	x
License Manager	✓	✓	✓	x

Running AD Utilities

To run the AD Utilities, you perform tasks such as starting the utility and answering questions at the prompts.

Getting Ready

Before you run any of the AD Utilities, there are a few prerequisite preparation steps that are common to all the utilities.

> **Note:** There may be additional platform-specific preparation steps. Check the Oracle Applications Release11*i* Release Notes for your platform.

1. Log in as applmgr (or your equivalent default main applications login).
2. Run the environment or command file for the appropriate product group.

   For UNIX users:

The environment file is typically APPSORA.env. To run the file, from a Bourne or Korn shell, type the following:

```
$ . $APPL_TOP/APPSORA.env
```

> **Note:** Depending on your setup, you may have already run this file when you logged in.

For NT users:

Run APPSORA.cmd (in %APPL_TOP%). Verify that APPL_CONFIG is set to the name of the product group registry subkey:

```
$ echo %APPL_CONFIG%
```

3. Verify that ORACLE_HOME is set to the proper database directory, and that TWO_TASK identifies the correct database.

   For UNIX users:

   Type the following:

```
$ echo $TWO_TASK
$ echo $ORACLE_HOME
```

   For NT users:

```
C:\> echo %LOCAL%
C:\> echo %ORACLE_HOME%
```

4. Ensure that $ORACLE_HOME/bin is in your PATH. NT users will check for %ORACLE_HOME%\bin.

   For UNIX users:

   At the prompt, type:

```
$ echo $PATH
```

   If $ORACLE_HOME/bin is not in the path, add it using the following command:

```
$ PATH=$PATH:$ORACLE_HOME/bin
$ export PATH
```

For NT users:

At the prompt, type:

```
C:\> echo %PATH%
```

If it is not there, add it using the following command:

```
C:\> Set PATH=%ORACLE_HOME%\bin;%PATH%
```

Other directories, such as the location of the JRE executable (from the Java runtime environment), should also be in your path. The adovars.env file, or adovars.cmd for NT users, should be updated to include all non-database specific directories in your PATH.

5. 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 of AD Administration.

   Additional Information: Administer Concurrent Managers, *Oracle Applications System Administrator's Guide*

6. Ensure that there is sufficient temporary disk space.

   You should have at least 50 MB in the temporary directories denoted by $APPLTMP and $REPORTS60_TEMP or %APPLTMP% and %REPORTS60_TEMP% for NT. You should also have space in the operating system's default temporary directory (usually /tmp or /usr/tmp for UNIX and C:\temp for NT).

   Note: There may be utility-specific preparation steps. Check the Running <utility name> section of the specific AD Utility.

Stopping and Restarting

You can exit the utility by entering *abort* at any prompt. Restart the utility by giving the <utility name> command again, where <utility name> is the command for the AD Utility (e.g. adpatch, adadmin or adaimgr). When you restart, you can specify the log file from the previous session or enter a new file name. When you reuse a log file, the utility adds the message "Start of <utility name> session" to the end of the file and appends messages from the new session as it generates them.

You can then do one of the following:

- Continue Session (the default)

  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 continue the previous session. It then starts from the beginning.

Responding to Prompts

Prompts for information typically include a default answer in square brackets:

```
<utility name> records your <utility name> session in a text file
you specify.  Enter your <utility name> log file name or press [Return]
to accept the default file name shown in brackets.

Filename [<utility name>.log] :
```

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

> **Attention:** Read the AD Utility screen prompts carefully to be sure you supply the correct information.

Applications Directory

The utility shows the value of the environment variable APPL_TOP and asks if this is the top applications directory. Here is an example, where /d01/appl/115 is your APPL_TOP:

```
Your default directory is '/d01/appl/115'.
Is this the correct APPL_TOP [Yes]?
```

Press [Return] to accept this default value. If the default directory is not correct, answer No to exit. Resume running the utility after you set APPL_TOP to the correct directory.

Log File

The log file prompt refers to the name of a log file that records the session. The default file name is <utility name>.log (for example, for AD Administration, the default log file is adadmin.log).

```
<utility name> records your <utility name> session in a text file you specify.
Enter your <utility name> log file name or press [Return] to accept the default
name shown in brackets.

Filename [<utility name>.log] :
```

The utility places the log file in $APPL_TOP/admin/<db_name>/log, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. For NT, it puts the file in %APPL_TOP%\admin\<db_name>\log, where <db_name> is the value of your LOCAL variable.

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 steps that operate in parallel mode. Review these adwork<number>.log files (adwork01.log, adwork02.log, ...) for detailed information about the errors.

Software Compatibility

The utility (AutoUpgrade only) prompts as follows:

```
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 your technology stack product versions. If you are uncertain whether a release combination is certified, check Oracle *MetaLink* (http://metalink.us.oracle.com/) or contact Oracle Support Services.

Email Notification

You can choose to be notified by email if a <utility name> worker fails at any time during the installation process. Enter your email address (for example, appldba@mycompany.com) at the system prompt.

```
You can be notified by email if a failure occurs.
Do you wish to activate this feature [Yes] ?
```

If you want more than one person to be notified, enter each email address separated by a space. The email notification contains the last 100 lines of the log file for the worker that failed.

```
You chose to be notified by email when a failure occurs.
Please enter the email ID(s) (separated by a space) that notifications should be
sent to [applmgr] :
```

Batch Size

For upgrades, enter a batch commit size to be used.

```
Please enter the batchsize [1000] :
```

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. 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.

Applications Environment Name

The Applications environment name is a default you set when you run Rapid Install. Once the name is provided, it is stored for use by all other AD Utilities. This prompt appears when AutoUpgrade is run and, if necessary, the default can be changed. Of the AD Utilities in this manual, only AutoUpgrade prompts for this item. This information is necessary for AutoPatch, which uses it to create a directory structure to store backup files under the patch directory. In future releases, the Applications environment name will be used by the Oracle Enterprise Manager (OEM) to identify the Applications environment.

```
Please enter the name of the Oracle Applications Environment that this APPL_TOP
belongs to.

The Applications Environment name must be unique across all Oracle
Applications Environments at your site, must be from 1 to 8 characters
long, and may only contain alphanumeric characters.

Sample Applications Environment names are: "prod", "test", and "demo".

Applications Environment Name [prod] :
```

Files Installed in APPL_TOP

The server configuration information is set when you run Rapid Install. The configuration determines the types of files that were unloaded by Rapid Install (Java files, HTML files, forms files, and concurrent programs files). After these questions are answered the first time, subsequent runs of any AD Utility use the

stored information. AutoUpgrade prompts for answers pertaining to server configuration and, if necessary, the defaults can be changed.

```
NOTE: If you do not have or choose not to have certain types of files installed
in this APPL_TOP, you may not be able to perform certain tasks.

Example 1: If you don't have files used for installing or upgrading
the database installed in this area, you cannot install or upgrade
the database from this APPL_TOP.

Example 2: If you don't have forms files installed in this area, you cannot
generate them or run them from this APPL_TOP.

Example 3: If you don't have concurrent program files installed in this area,
you cannot relink concurrent programs or generate reports from this APPL_TOP.

Do you currently have or want to install files used for installing or upgrading
the database in this APPL_TOP [Yes] ? Yes *

Do you currently have or want to install Java and HTML files for HTML-based
functionality in this APPL_TOP [Yes] ? Yes *

Do you currently have or want to install Oracle Applications forms files
in this APPL_TOP [Yes] ? Yes *

Do you currently have or want to install concurrent program files
in this APPL_TOP [Yes] ? Yes *
```

APPL_TOP Name

Like the Applications environment name, the APPL_TOP name is a default you set when you run Rapid Install. Once the name is provided, it is stored for use by all other AD Utilities. This prompt appears when AutoUpgrade is run and, if necessary, the default can be changed. Of the AD Utilities in this manual, only AutoUpgrade prompts for this item. This information is used by AutoPatch in creating a directory structure to store backup files under the patch directory. In future releases, the APPL_TOP name will be used by the Oracle Enterprise Manager (OEM) to identify the APPL_TOP.

```
Please enter the name Oracle Applications will use to identify this APPL_TOP.

 The APPL_TOP name you select must be unique within an Oracle Applications
 Environment, must be from 1 to 8 characters long, and may only contain
 alphanumeric and underscore characters.
```

```
AutoPatch has computed a default APPL_TOP name for you
based on the servers you have implemented in this APPL_TOP.

APPL_TOP Name [tafnw1] :
```

The default name is derived from the configuration of the environment and consists of variations of the string tafnw1, where

t stands for *server*

a stands for the *admin* server

f stands for the *forms* server

n stands for the *node* server

w stands for the *web* server

The trailing *1* allows for multiple APPL_TOPs within an Applications environment with the same configuration (the 1 must be manually changed to a 2 for the second identical configuration).

If a given server is not implemented in this APPL_TOP, its corresponding letter is replaced by an underscore.

Examples: If all servers are implemented, the default APPL_TOP name is tafnw1. If only the forms server is implemented, the default is t_f__1. If a second forms server is implemented, the default is still t_f__1, and it must be manually changed to t_f__2.

> **Additional Information:** *Oracle Enterprise Manager: Getting Started with the Oracle Management Pack for Oracle Applications*

ORACLE Database

The ORACLE database prompt shows you the database and database directory you are set to use. The AD Utility you are running asks you to confirm that they are the correct ones. 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/8.0.6'.

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

If the database and directory are not correct, answer No to exit. Resume running the utility after you change the values of the ORACLE_SID and ORACLE_HOME to identify the correct database and directory.

SYSTEM and AOL user passwords

Enter the database SYSTEM password at the following prompt:

```
<utility name> needs the password for your 'SYSTEM' ORACLE schema
in order to determine your installation configuration.

Enter the password for your 'SYSTEM' ORACLE schema:
```

The utility connects to the database and determines the ORACLE user name for the AOL schema that uniquely identifies your existing product group. It prompts for the password of this schema, then verifies the connection:

```
The ORACLE username specified below for Application Object Library
uniquely identifies your existing product group: APPLSYS

Enter the ORACLE password of Application Object Library [APPS] :
```

> **Note:** If the utility fails to connect after being provided with the password, make sure you are using an ORACLE_HOME with an RDBMS version that is certified with the Applications release and that you have linked your AD programs with this certified version.

The AD Utilities obtain the remaining information they need directly from the database, including the installed languages, module information, and product dependencies.

> **Attention:** There may be utility-specific requirements at this point. Refer to the Running <utility name> section of the utility you are running for a list of additional information.

Configuration and Environment Files

The following configuration and environment files are pertinent to, and used by, most of the AD Utilities.

File name	Description
adconfig.txt	The main configuration file located in $APPL_TOP/admin or in %APPL_TOP%\admin on NT. This file contains environment information for all of Applications and is used by all AD Utilities. Do not update this file manually.
def.txt	The file that contains the defaults used by AutoPatch and AD Administration when operating in non-interactive mode. The actual name of the file is determined when you create it. Must be created in $APPL_TOP/admin/<db_name>. In NT, it must be in %APPL_TOP%\admin\<db_name>. Refer to AD Administration in Chapter 2 and AutoPatch in Chapter 4 for more information.
applprod.txt	The AD Utilities product description file, which is used to identify all products and product dependencies. It is located in the admin directory under APPL_TOP.
applterr.txt	The AD Utilities territory description file, located in the admin directory under APPL_TOP. It contains information on all territories and localizations that are supported.
applora.txt	Contains information about required init.ora parameters for runtime. Located in the admin directory under APPL_TOP.
applorau.txt	Contains information about required init.ora parameters for install and upgrade. Located in the admin directory under APPL_TOP.
<db_name>.env	The main environment file used to configure the environment to run Applications. It is created by Rapid Install, but can be recreated by AutoUpgrade, AD Splicer or AD Administration. Located directly under APPL_TOP. APPLSYS.env is the default name, however, other names may be substituted.
adovars.env	Called by the main environment file and is used to set environment variables for Java and HTML. It is located in the admin directory under APPL_TOP.
fndenv.env	Sets additional environment variables used by Oracle Applications Application Object Library. This file should not be modified; the default values should be applicable for all customers.

Creating an Environment File

The environment file sets environment variables necessary to use an Oracle Applications product group. It can be created by AutoUpgrade, AD Splicer or AD Administration. This section explains the questions asked when you create an environment file.

> **Additional Information:** Running AD Administration Interactively in Chapter 2; Setting Tasks in Chapter 3

Environment File Name

AD Administration or AD Splicer prompt for the file name to use when creating the environment file. The default is <db_name>.env, where <db_name> is the name of the database in which the current product group is installed. AutoUpgrade does not ask this question, as the file name is provided as a parameter on the Tasks screen.

```
Enter the name of your Oracle Applications environment file below.
File name [test115.env] :
```

Parallel Concurrent Processing

The utilities ask whether you want to enable parallel concurrent processing. The option you select determines whether the utilities set the environment variable APPLDCP (Distributed Concurrent Processing feature) to OFF, ON, or OSQ, respectively:

```
How do you wish to enable Parallel Concurrent Processing:

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

The default choice is 1 - Not enabled.

Enter your choice [1] :
```

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 in to 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 File Names

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

```
The concurrent managers can create output files which
use a name that is no longer than 8 characters and
an extension which is no longer than 3 characters.

Do you wish to use the 8.3 file name convention [No] :
```

Directories for Log and Output Files

The utilities prompt for information about log and output file directories. You 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.

```
The concurrent managers can put all the log and report files in a common area
where the client machines can view them.

Enter the name of this common area below, or press [Return]
if you want log and report files for each application to go
in that application's log and output subdirectories.
Enter the name of the common area:
```

Press [Return] at this prompt if you plan to put 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.

If you entered a common area, then you will be prompted to specify the subdirectories that hold the log and output files, respectively. These log and output directories must already exist. The utilities set the environment variables APPLLOG and APPLOUT to these directories in the group's environment file. Use the default values "log" and "out" if any *one* of these conditions applies:

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

  ```
  You have entered '$APPL_TOP/admin'. Is this correct [Y] ? Y
  Enter the log subdirectory name for this product group [log] :
  Enter the output subdirectory name for this product group [out] :
  ```

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 REPORTS60_TMP variable to this directory.

```
Enter the directory for Applications temporary files :
Enter the directory for Oracle Reports temporary files :
```

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*

Additional Directories for Temporary Files

The utilities 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.

```
Some PL/SQL programs produce temporary log/output files. The directories
used for this must be listed in the init.ora parameter "utl_file_dir".
The value of utl_file_dir for this database is:

"/sqlcom/inbound, /sqlcom/outbound, /sqlcom/log, /sqlcom/out"

Enter the directory for temporary log/output files from PL/SQL programs.
Directory:
```

> **Additional Information:** Database Initialization Parameters, Chapter 6

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

> **Note:** AutoUpgrade and AD Administration will fail at this point if the utl_file_dir parameter has no value.

Web Server

Oracle Applications forms and context-sensitive help are accessed from the desktop client through a connection to a web server. AutoUpgrade and AD Administration need to know the name of the machine where 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.

  ```
  What is the name of the machine, including domain name, hosting the
  web server that will be used for accessing Applications forms?

  Applications forms web server host machine [ ] ?
  ```

- The port that the Applications forms web server is running on. The default is 80.

  ```
  What port is the Applications forms web server running on [80] ?
  ```

This information is recorded in the FORMS60_MAPPING variable in your environment file.

> **Note:** The utilities also create a Java security file (env.txt) in OA_HTML for web use. If OA_HTML does not exist or does not have write privileges for the applmgr account, AutoUpgrade generates an error at this point. You must supply the OA_HTML information in the $APPL_TOP/admin/adovars.env file before you can continue. For NT users, the information should be in the adovars.cmd file in %APPL_TOP%\admin.

> **Additional Information:** Edit adovars.env file, *Installing Oracle Applications, Release 11i*

Verify the New Environment File

After you complete the final question, AutoUpgrade, AD Splicer, or AD Administration creates the environment file. The following message should appear:

```
Computing APPLFULL and APPLSHAR values...

Creating environment file...

Done.

Review the messages above, then press [Return] to continue.
```

The new environment file is placed directly under APPL_TOP.

AD Timing Report

The AD Timing Report (adt<session_id>.lst, where <session_id> is your AD program session ID) is produced by AutoUpgrade, AutoPatch, and AD Administration. This report provides information on long-running processes during the operation of these three utilities. The output is placed in $APPL_TOP/admin/<db_name>/out. NT users will find it in %APPL_TOP%\admin\<db_name>\out. It contains the following information:

- Time-consuming jobs
- Failed jobs
- Incomplete jobs
- Total run time
- Percent Usage by Product
- Percent Usage by Phase and Product

The report can also be generated manually during an upgrade or anytime thereafter to view timing statistics from a prior session, where <session id> is the session of the timing statistics you would like to see, and <output file> is the name of the file where the statistics will be written:

For UNIX users:

```
$ cd $AD_TOP/admin/sql
$ sqlplus <APPS username>/<APPS password> @adtimrpt.sql <session id> \
  <output file>
```

For NT users:

```
C:\> cd %AD_TOP%\admin\sql
C:\> sqlplus <APPS username>/<APPS password> @adtimrpt.sql <session id> \
    <output file>
```

2

AD Administration

This chapter contains information about using the AD Administration utility. It contains the following sections:

- What is AD Administration?
- Running AD Administration Interactively
- Running AD Administration in Non-Interactive Mode

What is AD Administration?

AD Administration (adadmin) performs maintenance tasks on an installed Oracle Applications system to ensure that it runs smoothly. These tasks fall into two categories: database and file system. Like AutoUpgrade and AutoPatch, AD Administration can run parallel workers for most database tasks and some file system tasks.

Database Tasks

Database tasks are performed from the Maintain Applications Database Objects menu. These tasks are performed with installed database objects, and include the following:

- Validate APPS schema(s)
- Compile APPS schema(s)
- Recreate grants and synonyms for APPS schema(s)
- Compile flexfield data in AOL tables
- Maintain multi-lingual tables

- Check DUAL table
- Maintain Multiple Reporting Currencies schema(s)
- Convert to Multiple Reporting Currencies
- Convert to MultiOrg

File System Tasks

File system tasks are performed for product files from the Maintain Applications Files menu. They include the following jobs, many of which require database access:

- Create Applications environment file
- Relink Applications programs
- Copy files to destinations
- Verify files necessary for runtime
- Generate message files
- Generate form files
- Generate report files
- Generate graphics files
- Generate product jar files

Running AD Administration Interactively

Before starting AD Administration, review and complete the Running AD Utilities section of Chapter 1. Once AD Administration is started, use this section and the Responding to Prompts section of Chapter 1 to guide you through the list of questions that you need to answer. Start AD Administration with its command name.

For UNIX users:

```
$ adadmin
```

For NT users:

```
C:\> adadmin
```

Log File

The default AD Administration log file name is adadmin.log. It is located in $APPL_TOP/admin/<db_name>/log, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. NT users will find the log file in %APPL_TOP%\admin\<db_name>\log.

Additional Information: Running AD Utilities in Chapter 1

Using the Main Menu

After you answer the AD Administration questions, the utility takes you to the Main Menu. Here you can choose to maintain Applications database objects, maintain Applications files, or exit.

Maintaining Applications Database Objects

You use the Maintain Applications Database Objects menu to perform tasks that involve Oracle Applications database objects.

```
        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.    Maintain Multiple Reporting Currencies schema(s)
    8.    Convert to MultiOrg
    9.    Return to Main Menu

Enter your choice :
```

Select the task number to perform a task. The tasks in this section should be run only on your administration server.

Note: The options you see on this and other AD Utilities menus depend on the configuration of your Applications database. The options and their numbers shown above may differ slightly from those in your installation.

The first time a task is selected from the Database Objects menu, AD Administration validates your init.ora parameters against applora.txt. It only checks the parameters for the first task selected. It does not recheck if other tasks from the Database Objects menu were selected in the same AD Administration session.

1. Validate APPS schema(s)

   Runs a SQL script (advrfapp.sql) against the APPS schemas, verifying the integrity of each schema. It also checks for certain conditions that are undesirable, but will not produce fatal problems. You can run this task at any time, but it is most effective if run:

   - immediately after an upgrade
   - after a patch is applied (for multiple patches, run it once after you apply all the patches)
   - before converting to MultiOrg or Multiple Reporting Currencies (MRC)
   - after performing an export/import (migration)
   - when doing custom development in APPS schema(s)

   You may also want to run this task whenever you receive a runtime error that suggests a problem coming from the AD_DDL package. For example, when flexfield view generator or document sequences fail.

   An output file, <APPS schema name>.lst, is produced for each APPS schema. The file is located at $APPL_TOP/admin/<db_name>/out, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. For NT users, the file is located in %APPL_TOP%\admin\<db_name>\ out, where <db_name> is the value of your LOCAL variable. Review this file and fix all problems. The output file for each APPS schema contains information about fixing problems. Rerun this report as you make changes until there are no more problems listed.

   The <APPS schema name>.lst file is divided into three sections:

   - Problems you MUST fix - not specific to this APPS schema
   - Problems you MUST fix - specific to the APPS schema named <APPS schema name>
   - Issues you may want to address - specific to the APPS schema named <APPs schema name>

   You must fix issues that are included in the first two sections before you can run MRC maintenance. Issues in the third section can be ignored, however, it may be a good idea to fix them at some point.

Validate APPS can also be run manually through SQL*Plus by using the following command:

For UNIX users:

```
$ cd $AD_TOP/admin/sql
$ sqlplus <system username>/<system password> @advrfapp.sql \
  <APPS schema name> <AOL schema name>
```

For NT users:

```
C:\> cd %AD_TOP%\admin\sql
C:\> sqlplus <system username>/<system password> @advrfapp.sql \
     <APPS schema name> <AOL schema name>
```

> **Note:** If Validate APPS is run manually, it must be run separately for each APPS schema.

2. Compile APPS schema(s)

   Spawns parallel workers to compile invalid database objects in the APPS schemas. It uses the same parallel compile phases as AutoUpgrade. You can run this task at any time, but it is most effective if run:

   - after custom packages are moved to the APPS schema and need to be compiled
   - after applying patches that alter packages in the APPS schema

3. Recreate grants and synonyms for APPS schema(s)

   Identifies missing grants from a base product schema to the corresponding APPS schema or missing synonyms in the APPS schema for the object and recreates them.

   > **Note:** This task does not set up grants and synonyms for the multicurrency schemas. You must choose those tasks specifically from the Database Objects menu. This task also ignores overflow tables created for Index-Organized Tables (IOT).

   This option 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.

Run this task when grants and synonyms are missing from an installation. This may occur as a result of custom development, incomplete database migrations (exports/imports), or AD Utility sessions that failed to run successfully to completion.

4. Compile flexfield data in AOL tables

   Compiles flexfield data structures in Oracle Application Object Library (AOL) tables. If you choose not to compile the structures, each one is compiled the first time any user accesses the flexfield.

   Run this task after you apply a patch that changes the setup of flexfields. The patch usually indicates when this step should be performed. Running this task is never required, as flexfields automatically compile the data when they are first used. But it can alleviate a one-time runtime performance issue when the data is compiled automatically.

5. Maintain multi-lingual tables

   Calls PL/SQL routines to maintain multi-lingual tables for some Applications (including AOL, AK, and AZ) by adding any missing, untranslated rows. You generally need to run this task only when requested by Oracle Support Services, or when instructed to do so by Oracle Applications documentation or patch readme files.

6. Check DUAL table

   Verifies that the DUAL table exists in the SYS schema, is accessible by Applications, and contains only one row.

7. Maintain Multiple Reporting Currencies schema(s)

   Appears as a menu choice on the Database Objects menu *only* if Multiple Reporting Currencies (MRC) functionality is installed in your database. If you do not already have MRC installed, you will see a Convert to Multiple Reporting Currencies option, which you can use to install MRC.

   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.

   The Maintain Multiple Reporting Currencies 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 the existing MRC schema(s). The default is No.

  ```
  Update MRC schema itself(MRC schema objects are always updated)[No]?
  ```

  Note: If you answer "Yes" to Update MRC schema, the Concurrent Managers must be shut down.

- Calls a PL/SQL procedure which maintains database objects in the MRC schema(s).
- Compiles all invalid objects in the MRC schema(s) in parallel. The default is No.

  ```
  Compile invalid objects after updating MRC schema objects [No]?
  ```

  Invoker's Rights conversion takes place with the compilation.
- Recreate MRC triggers. The default is No.

  ```
  Recreate MRC triggers in the APPS schema(s) [No]?
  ```

The three optional questions above are set up for rapid MRC maintenance after patches. Usually the user will accept the default answers of "No". You should type "Yes" to these questions:

- after an upgrade
- when instructed to do so in a patch readme file

Once AD Administration completes this process, check the log file (adadmin.log) in the $APPL_TOP/admin/<db_name>/out directory and correct all problems. NT users will find this file in %APPL_TOP%\admin\<db_name>\out. Rerun this report as you make changes until there are no more problems listed.

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.

Additional Information: Multiple Reporting Currencies, *Oracle Applications Concepts; Multiple Reporting Currencies in Oracle Applications*

8. Convert to Multiple Reporting Currencies

Appears on the Database Objects menu only if Multiple Reporting Currencies functionality is *not* 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 the Multiple Reporting Currencies (MRC) schema(s)
- Creates the correct system privileges and grants in the MRC schema(s)
- Registers the MRC schema(s) with Oracle Applications
- Sets up one or more data groups for the 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 that creates database objects in the MRC schema(s)
- Compiles all invalid objects in the MRC schema(s) in parallel
- Creates MRC triggers in the APPS schema(s)

  Additional Information: *Multiple Reporting Currencies in Oracle Applications*

9. Convert to MultiOrg

Displays on the Database Objects menu only if MultiOrg and/or Multiple Sets of Books Architecture are *not* installed in your database.

This task converts a standard product group (not Multiple Sets of Books Architecture and not MultiOrg) into a MultiOrg 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. This site-level profile option must remain set at all times.

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 MultiOrg 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 MultiOrg 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 the APPS schema
- Disables all triggers in the APPS schema
- Converts seed data and transaction data to MultiOrg in parallel
- Re-enables all previously enabled triggers in the APPS schema

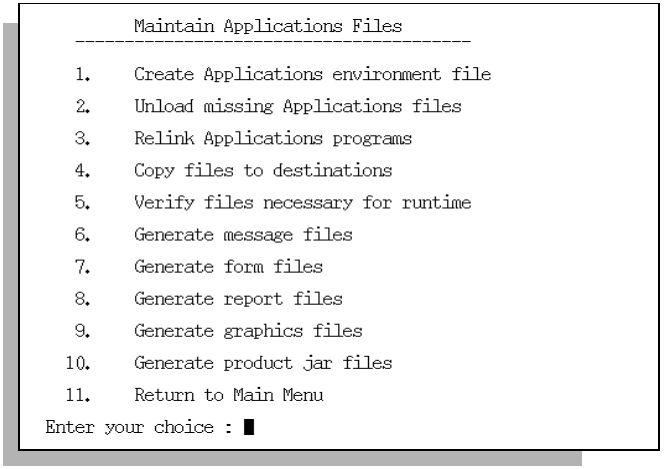
Additional Information: *Multiple Organizations in Oracle Applications*

10. Return to main menu

Select this task to return to the AD Administration main menu.

Maintaining Applications Files

Use the Maintain Applications Files menu to perform tasks related to the Oracle Applications product files.



Select the task number to perform a task.

Unlike the database tasks that need to be performed on the admin server, the file system tasks need to be performed on different servers. The following table indicates which tasks are performed on which servers:

Task	Admin Server	Forms Server	Web Server	Concurrent Processor Server
Create Applications environment file	✓	✓	✓	✓
Relink Applications programs	✓	✓	✓	✓
Copy files to destinations	✓	✓	✓	✓
Verify files necessary for runtime	✓	✓	✓	✓
Generate message files	✓	✓	✓	✓
Generate form files	x	✓	x	✓

Task	Admin Server	Forms Server	Web Server	Concurrent Processor Server
Generate report files	x	x	x	✓
Generate graphics files	✓	✓	x	✓
Generate product jar files	✓	x	✓	x

1. Create Applications environment file

Creates an environment file that defines Oracle Applications environment variables. Enter the file name as a parameter. The environment file can have any name, but it should use the .env extension in UNIX and a .cmd extension in NT.

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. Making a backup of the existing environment file is recommended.

After you supply the file name, the utility asks for information necessary for the Applications environment.

> **Additional Information:** Creating an Environment File in Chapter 1

After the utility generates the environment file, you can make customizations in adovars.env and run the generated environment file as necessary.

2. Unload missing Applications files

This feature is no longer supported in Release 11i. Rapid Install unloads files for you. The menu item will be removed in a subsequent release.

3. Relink Applications programs

Relinks Oracle Applications executable programs with the Oracle8 Server libraries so that they function with the Oracle8 Server database. For each product, you 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.sh utility manually.

Additional Information: AD Relink (adrelink.sh) in Chapter 5

4. Copy files to destinations

   Copies files from each product area to 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 copy files to the other destinations only 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 in UNIX...	are copied to in NT...
Java files	$JAVA_TOP	%JAVA_TOP%
HTML files	$OAH_TOP	%OAH_TOP%
Media files	$OAM_TOP	%OAM_TOP%

   The directories for the variables are specified in the adovars.env file for UNIX and the adovars.cmd file for NT.

   Note: When this option is used to copy reports or graphics files, the default destination is under AU_TOP.

   Additional Information: Web Server in Chapter 1

5. Verify files necessary for runtime.

   Verifies that all files needed to run Oracle Applications for the current configuration are in the current APPL_TOP. Select this task if you encounter a problem with a missing file at run time.

6. 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. Generally, you will only regenerate an Applications message when directed to do so in the readme.txt instructions for a patch.

7. Generate form files

   Generates binary Oracle forms files (extension **.**fmx) 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. This task performs the following actions:

   - 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, menu files, and executable files
   - Asks for which products you want to generate Oracle Forms objects
   - Asks if you want to generate specific forms objects for each selected product
   - Displays the current set of installed languages and asks if you want to generate Oracle Forms files in these languages
   - 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 (Perform this step only 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

   Note: If any Oracle Forms, Reports, or Graphics objects or Oracle product JAR files did not generate successfully, AD Administration 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 AD Administration log file to determine whether the problems require additional attention. If you choose not to continue and later restart your session, AD Administration attempts to generate only the files that did not generate successfully.

Form Generation Warnings

When generating form files, you may encounter messages like this:

```
WARNING generating form forms/US/POXPOVCT.fmx from input file
      /d01/apps/115/au/11.5.0/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 is a warning you can ignore. Certain warnings are acceptable and occur while you are 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. You can safely ignore the warnings in the following list when you generate Applications forms:

```
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 usually be ignored. They may demand attention when applying patches.

8. Generate report files

   - Generates the binary Oracle Reports report files (extension .rdf) for all installed languages. It performs the following actions:
   - 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 or executable files
   - Asks for which products you want to generate Oracle Reports objects
   - Asks if you want to generate specific Oracle Reports objects for each selected product
   - Displays the current set of installed languages and asks if you want to generate Oracle Reports files in these languages
   - 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 (Perform this step only if you chose to generate specific Oracle Reports objects for each selected product.)
- Generates all selected Oracle Reports objects for all selected products

9. Generate graphics files

This task generates the Oracle Graphics files (extension .ogd) from the graphics definition files (extension .ogx) for all installed languages. It performs the following actions:

- Displays the current character set (from NLS_LANG) and asks if you want to generate Oracle Graphics objects in this character set
- Asks if you want to regenerate Oracle Graphics PL/SQL library files or executable files
- Asks for which products you want to generate Oracle Graphics objects
- Asks if you want to generate specific Oracle Graphics objects for each selected product
- Displays the current set of installed languages and asks if you want to generate Oracle Graphics files in these languages
- Creates a list of all Oracle Graphics objects to generate
- Displays the list of Oracle Graphics objects to be generated and allows you to select whether to regenerate specific objects or all objects (Perform this step only if you chose to generate specific Oracle Graphics objects for each selected product.)
- Generates all selected Oracle Graphics objects for all selected products

10. Generate product jar files

This option is recommended anytime you apply an Oracle Forms patch. This task:

- Generates all JAR files that are out of date
- With the force option, generates all JAR files for all products
- Generates product JAR files in APPL_TOP and JAVA_TOP
- Copies Oracle Forms registry file (Registry.dat) from ORACLE_HOME/forms60/java/ to JAVA_TOP/oracle/forms/registry

- Signs JAR files, if on web server

This option will fail if any Oracle product JAR files do not generate successfully. Review the AD Administration log file to determine whether or not the problems require additional attention. When you restart your failed AD Administration session, adadmin will only attempt to generate the Oracle product JAR files that did not generate successfully.

11. Return to Main Menu

    Select this task to return to the AD Administration main menu.

Running AD Administration in Non-Interactive Mode

AD Administration has the capability to run some file system and database tasks without user intervention. This non-interactive feature allows you to schedule it to run routine tasks.

Creating a Defaults File

Before you can run AD Administration non-interactively, you must first create an AD Administration defaults file for your current environment (APPL_TOP and ORACLE_SID/TWO_TASK for UNIX or LOCAL for NT). Therefore, the first time you run your routine task, you need to run it interactively. To create the defaults file:

1. Specify defaultsfile=<Defaults File Name> on the AD Administration command line. The defaults file must be located under $APPL_TOP/admin/<db_name> for UNIX systems, where testdb1 is the database name (ORACLE_SID/TWO_TASK). For example:

   ```
   $ adadmin defaultsfile=$APPL_TOP/admin/testdb1/adadmindef.txt
   ```

   For NT systems, the file must be located under %APPL_TOP%\admin\ <db_name>, where testdb1 is the database (LOCAL). For example:

   ```
   C:\> adadmin defaultsfile=%APPL_TOP%\admin\testdb1\adadmindef.txt
   ```

2. Run AD Administration through the maintenance task that you would like to run non-interactively in the future.

   Note: All questions during the session must be answered. If they are not, AD Administration may abort when running in non-interactive mode.

3. Verify that your defaults file exists.

Once you have an AD Administration defaults file for your current environment, you can run AD Administration non-interactively. The command parameter is *interactive=no*. Here is an example for UNIX:

```
$ adadmin defaultsfile=$APPL_TOP/admin/testdb1/adadmindef.txt \
  logfile=adadmin_noninteractive.log workers=5 interactive=no
```

For NT, the command looks like this:

```
C:\> adadmin defaultsfile=%APPL_TOP%\admin\testdb1\adadmindef.txt \
  logfile=adadmin_noninteractive.log workers=5 interactive=no
```

This command does the following:

- Reads the defaults file in the patch shown in the example
- Records the non-interactive session in the log file called adadmin_ noninteractive.log
- Uses five parallel workers
- Runs AD Administration in non-interactive mode

Restart a Failed Non-Interactive Session

To restart a failed non-interactive session, use the `restart=yes` command line parameter to prevent AD Administration from reasking restart questions that the defaults file cannot answer.

```
$ adadmin defaultsfile=$APPL_TOP/admin/testdb1/adadmindef.txt \
  logfile=adadmin_noninteractive.log orkers=5 nteractive=no restart=yes
```

3

AutoUpgrade

This chapter describes how to run AutoUpgrade. It explains the questions that AutoUpgrade asks and illustrates how to use the AutoUpgrade screens to configure your upgrade. This chapter contains these sections:

- What is AutoUpgrade?
- Running AutoUpgrade
- Using AutoUpgrade Menus and Screens
- Running the Upgrade
- Monitoring AutoUpgrade
- After Running AutoUpgrade

What is AutoUpgrade?

AutoUpgrade is an AD Utility that you use to upgrade to the latest version of Oracle Applications. You run AutoUpgrade after you complete the pre-upgrade tasks as outlined in the latest version of *Upgrading Oracle Applications*, including the basic installation of the R11*i* components with Rapid Install—installing the Oracle8 technology stack and file system —to perform tasks such as updating the Oracle Applications database, licensing a localization or translation, converting to government financials, or licensing products.

Using AutoUpgrade

AutoUpgrade performs the tasks that are necessary to upgrade your Oracle Applications products to a new release level.

> **Attention:** If the database you are upgrading uses Replication, this feature must be disabled before running AutoUpgrade. For additional information on how to disable Replication, see Disabling the Replication Facility in the *Oracle8i Replication Manual*.

You can use AutoUpgrade to complete various tasks to enhance the basic upgrade. Some of the tasks that you may want to perform are:

- License additional products

  License any additional products not already licensed. Rapid Install copies all required files to your APPL_TOP.

- License localization (country specific functionalities) products

  The APPL_TOP created by Rapid Install includes all files belonging to localizations at the base release level. You can use AutoUpgrade to license any localization modules.

> **Note:** You cannot have more than one product group per database instance. If AutoUpgrade finds more than one product group, it will stop with a fatal error. See the *Applications Concepts Manual* for additional information on product groups.

> **Note:** To fully license a new product or localization (country specific functionalities) after an upgrade, see License Manager in Chapter 5.

How AutoUpgrade Works

AutoUpgrade operation is guided by the answers you give to a series of AutoUpgrade questions. To run AutoUpgrade:

- Answer the initial questions
- Specify the tasks AutoUpgrade will perform

- Choose the number of workers
- Monitor AutoUpgrade as it runs
- Fix AutoUpgrade errors and restart as necessary

On the AutoUpgrade screens, you can:

- Select products to license
- Specify tablespace names, ORACLE schema names and passwords, and sizing factors for AutoUpgrade to use
- Select from the complete set of available AutoUpgrade tasks the ones that pertain to your upgrade

Note: You *cannot* de-install a product you have already licensed.

Running AutoUpgrade in a Multi-server Environment

Rapid Install delivers the Oracle Applications file system components on every server. AutoUpgrade upgrades the database objects shared by all servers, therefore, you only need to run AutoUpgrade on the administration server. There is no need to run it on the other servers.

Attention: You cannot run multiple sessions of AutoUpgrade simultaneously against a product group.

Additional Information: Tasks Screen in this chapter

Running AutoUpgrade

Before starting AutoUpgrade, review and complete the Running AD Utilities section of Chapter 1.

Starting and Stopping AutoUpgrade

After setting up your environment, you can start AutoUpgrade (adaimgr) from any directory by typing *adaimgr*. AutoUpgrade displays an introduction screen and asks questions about your upgrade.

> **Note:** Although it can be used with non-graphic terminal devices, we recommend that when you use AutoUpgrade you use a window-based terminal so you can monitor and control upgrade processes in other windows while AutoUpgrade is running.

You can exit from the AutoUpgrade menus and screens by entering *abort* at any AutoUpgrade prompt. After you exit, you can restart AutoUpgrade from where your last session ended or from the beginning.

> **Warning: Once AutoUpgrade begins to upgrade products in the database, do not attempt to stop it without first contacting Oracle Support Services.**

> **Attention:** If an error occurs while AutoUpgrade is upgrading the database, you can correct the error without stopping AutoUpgrade.

Additional Information: Correcting AutoUpgrade Errors in this chapter

Responding to Prompts

AutoUpgrade asks the same questions as AD Administration. In addition, it asks some additional questions. Use this section and the Responding to Prompts section of Chapter 1 to guide you through the questions AutoUpgrade will ask.

AutoUpgrade log file

As stated in chapter 1, the default log file name is <utility name>.log. For AutoUpgrade the default log file name is adaimgr.log. It resides in $APPL_TOP/admin/<db_name>/log, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. NT users will find this file in %APPL_TOP%\admin\<db_name>\log, where <db_name> is the value of your ORACLE_SID or LOCAL variable.

Identify your Organization Type

AutoUpgrade prompts you to indicate whether your organization would like to continue using the industry that is currently installed.

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

If you are upgrading an existing product group, AutoUpgrade automatically determines whether your products are installed for commercial use. If you installed the commercial versions of products, AutoUpgrade prompts you to select one of these options. Choose Option 2 to convert commercial Oracle Financials products to Oracle Public Sector Financials products during the upgrade.

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

Language

AutoUpgrade automatically displays the language or languages that are currently installed and the base language.

```
Reading FND_LANGUAGES to see what is currently installed.
Currently, the following language is installed:

Code   Language                              Status
----   ------------------------------------ ---------
US     American English                     Base

Your base language will be AMERICAN.
```

> **Additional Information:** Language Codes in the *National Language Support Guide*

Restarting AutoUpgrade

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

> **Additional Information:** Running AD Utilities in Chapter 1

Restart AutoUpgrade by re-issuing the *adaimgr* command. When AutoUpgrade prompts for the name of the log file, specify the file from the previous session or supply a new file name. When you reuse a log file, AutoUpgrade adds the message Start of AutoUpgrade 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

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

```
Your previous AutoUpgrade session did not run to completion.
Do you wish to continue with your previous AutoUpgrade session [Yes] ?
```

Continue Session

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

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

Start New Session

AutoUpgrade asks you to confirm your choice if you choose not to continue the previous session. It then restarts from the beginning. If you restart from the beginning, you must re-enter any previous configuration information that you want to retain.

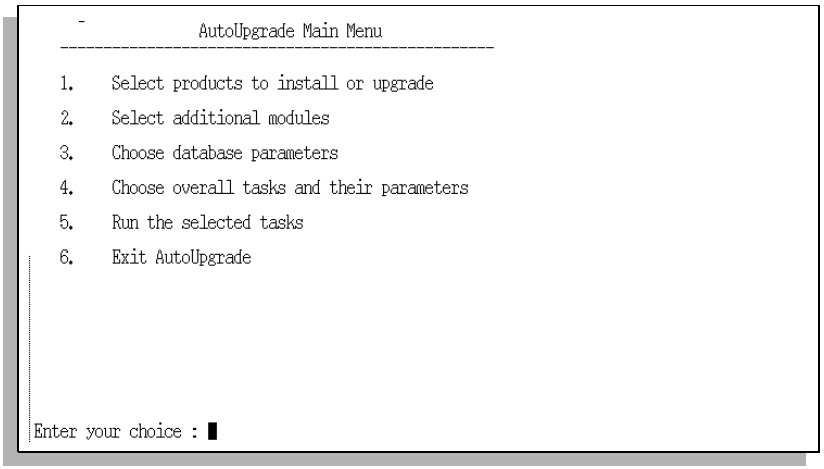
Warning: Do not restart AutoUpgrade from the beginning if it began to upgrade products in the database and then stopped. If this happens, determine why AutoUpgrade stopped, correct the problem, and restart your previous session. Alternatively, you can restore the last saved database and filesystem, then start AutoUpgrade again from the beginning.

Using AutoUpgrade Menus and Screens

The user interface for AutoUpgrade consists of a main menu and several screens that present upgrade information. After you answer the initial AutoUpgrade questions, you will see the AutoUpgrade Main Menu. Depending on the options you select from the Main Menu, you will see the associated screen. Review this section to see how these screens work.

Main Menu

The Main Menu presents the options you use to configure and run your upgrade.



Type the option number to select an option. You should generally choose the options in numeric order.

1. Select products to upgrade or install (license)

   Choose this option to license Oracle Applications products not already licensed. You can also review actions for currently licensed products using this screen.

2. Select additional modules

   Choose this option to license localization modules for Oracle Applications products.

3. Choose database parameters

   Choose this option to set the ORACLE user ID (schema name) and password, sizing factor, and tablespaces for each Oracle Applications product.

4. Choose overall tasks and their parameters

   With this option, you specify the tasks that AutoUpgrade performs. These tasks include actions such as installing database objects and generating flexfields.

5. Run selected tasks

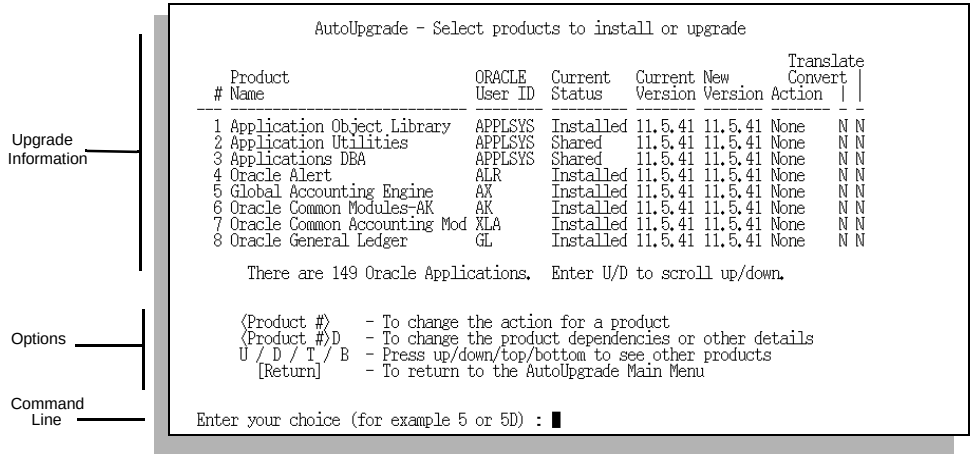
   Choose this option to start the upgrade. Do this only after you have configured the upgrade with the preceding options.

6. Exit AutoUpgrade

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

About the AutoUpgrade Screens

AutoUpgrade screens have a common user interface—they present upgrade information at the top and list your options at the bottom. The cursor appears at the end of the command line. You press [Return] to accept a default answer or return to the previous screen.



Selecting Options

You select an option by entering the letter, number, or combination of number and letter that designates the option. AutoUpgrade displays an error message if your

choice is invalid. When you select a valid option, AutoUpgrade carries out an action, prompts for more information, or displays another screen.

In the screen example, Oracle General Ledger is listed as product number 8. To change the dependencies and details for Oracle General Ledger, you press 8, the letter D, and then [Return]. AutoUpgrade then displays the product's Product Detail screen.

In most screens you can enter multiple values separated by spaces. For example:

```
7 9 21
```

toggles the license action for these three products.

Viewing Additional Screen information

When an AutoUpgrade screen contains more than one page of information, select the U option to scroll up or the D option to scroll down through the list of information. Use the T option to scroll to the top of the list or the B option to scroll to the bottom.

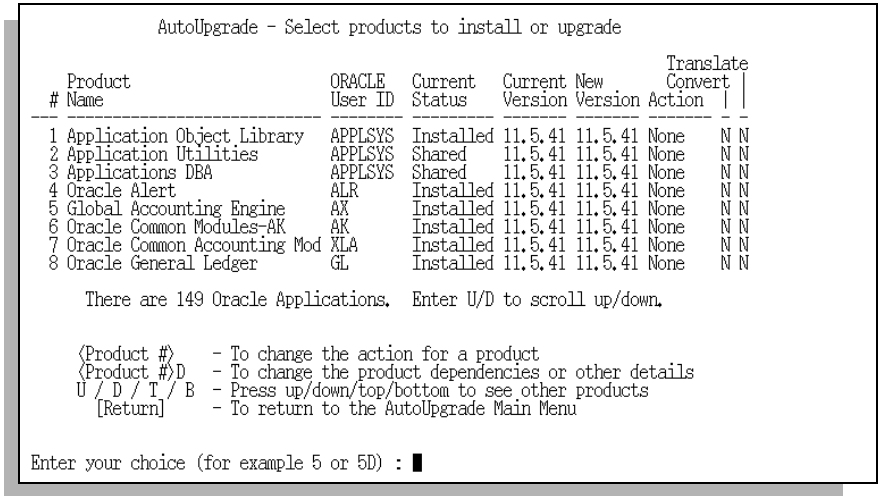
In the example, the screen lists eight Oracle Applications products. To display information about products 9 through 16, press D and [Return] to scroll down the list. Press B and [Return] to scroll to the bottom of the list and display the previous eight products.

Returning to the Previous Screen

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

Select Products Screen

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



```
            AutoUpgrade - Select products to install or upgrade

                                                                    Translate
  Product                          ORACLE   Current   Current New      Convert |
# Name                             User ID  Status    Version Version Action  | |
--- ------------------------------ -------- --------- ------- ------- ------- - -
  1 Application Object Library     APPLSYS  Installed 11.5.41 11.5.41 None    N N
  2 Application Utilities          APPLSYS  Shared    11.5.41 11.5.41 None    N N
  3 Applications DBA               APPLSYS  Shared    11.5.41 11.5.41 None    N N
  4 Oracle Alert                   ALR      Installed 11.5.41 11.5.41 None    N N
  5 Global Accounting Engine       AX       Installed 11.5.41 11.5.41 None    N N
  6 Oracle Common Modules-AK       AK       Installed 11.5.41 11.5.41 None    N N
  7 Oracle Common Accounting Mod   XLA      Installed 11.5.41 11.5.41 None    N N
  8 Oracle General Ledger          GL       Installed 11.5.41 11.5.41 None    N N

     There are 149 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 AutoUpgrade 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 license.

Lists the product numbers. You use these numbers to change the license actions or product details. When licensing or upgrading multiple copies of a product, check the ORACLE User ID column to ensure that you are selecting the appropriate copy.

Product Name Lists the names of 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 licenses 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 installed in the database.

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

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

Convert Automatically set to Y if you will upgrade products for public sector, education, or not-for-profit use, and cannot be changed on this screen.

Translate Indicates if AutoUpgrade is licensing a language other than American English. This column's value is determined by your existing installed languages and cannot be changed on this screen.

Setting Upgrade Actions

Enter the product number to change a product's licensed action. To license a product, select the product you would like to license. AutoUpgrade will change its action to Install and automatically set the actions of its dependent products to Shared. AutoUpgrade upgrades existing licensed products automatically. You cannot change the action for an existing fully-installed product.

The following entries summarize the upgrade actions:

- None

  AutoUpgrade does not license the product if the action is None. When you select a product currently set to None, AutoUpgrade changes the product's 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 that other products be fully-installed, these other products' actions will be changed to Install, if not already installed.

- Install

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

- Shared

  AutoUpgrade 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 license a product set to Shared by selecting the product. You cannot directly change the action from Shared to None or from None to Shared. Some products are "shared-only products" and cannot be set to Install.

- Upgrade

  When you are upgrading an existing product group, AutoUpgrade automatically sets this action for fully-installed products not yet upgraded to the current release. You cannot change the action for an existing fully-installed product.

Changing Select Products Screen Information

When you are upgrading, be sure to:

- Verify that AutoUpgrade has set the Action, Convert, or Translate columns correctly for each product.
- Change the action to Install for any additional products you would like to license in this release.
- If your database uses Multiple Sets of Books Architecture, 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: AutoUpgrade will fail during the upgrade if the dependencies have not been set correctly for every product in each product installation group.

Upgrading Multiple Sets of Books Architecture Databases

If you are installing Oracle Applications for the first time in Release 11i and you need multiple sets of books functionality, you must use the Multiple Organization architecture.

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

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 MultiOrg 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, AutoUpgrade 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 151 products listed in the Select Products screen before you select the A option, the first copied product displays as product number 152.

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

AutoUpgrade 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 schema names and passwords using the Database Parameters screen.

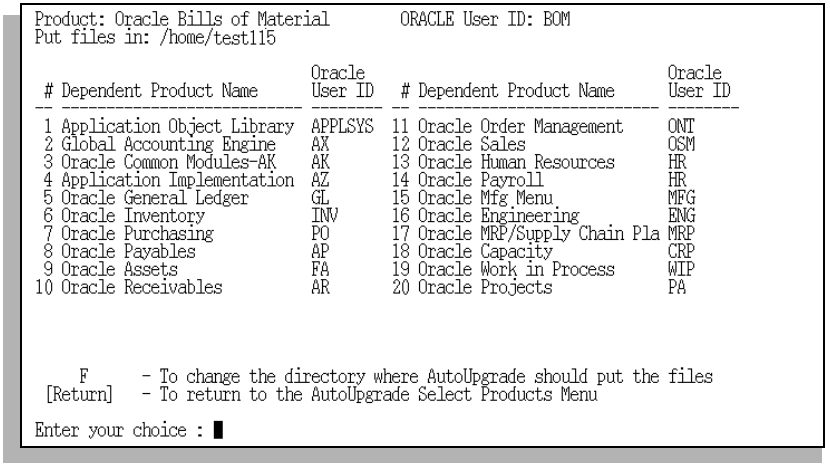
Additional Information: Database Parameters Screen in this chapter

AutoUpgrade initially sets the upgrade action for each copy of a product to None. Change the upgrade action to Install to license a product in a set of books. Leave the upgrade action set to None on products for which you do not want to license additional copies.

Note: You may see messages that read "Warning: duplicate job afdrpsyn.sql found in driver files!". You can safely ignore these messages.

Product Detail Screens

Each copy of an Oracle Applications product has a Product Detail screen where you can set specific upgrade 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: /home/test115

                              Oracle                                  Oracle
 # Dependent Product Name      User ID  # Dependent Product Name       User ID
-- --------------------------- -------- -- ------------------------- --------
 1 Application Object Library  APPLSYS  11 Oracle Order Management    ONT
 2 Global Accounting Engine    AX       12 Oracle Sales               OSM
 3 Oracle Common Modules-AK    AK       13 Oracle Human Resources     HR
 4 Application Implementation  AZ       14 Oracle Payroll             HR
 5 Oracle General Ledger       GL       15 Oracle Mfg Menu            MFG
 6 Oracle Inventory            INV      16 Oracle Engineering         ENG
 7 Oracle Purchasing           PO       17 Oracle MRP/Supply Chain Pla MRP
 8 Oracle Payables             AP       18 Oracle Capacity            CRP
 9 Oracle Assets               FA       19 Oracle Work in Process     WIP
10 Oracle Receivables          AR       20 Oracle Projects            PA




    F      - To change the directory where AutoUpgrade should put the files
 [Return]  - To return to the AutoUpgrade Select Products Menu

Enter your choice :
```

Use this screen to verify product dependencies for product installation groups.

Changing Product Detail

These options are applicable only if you are upgrading a multiple set of books architecture product group.

Option F: Change Directory for Product Files This option is no longer supported in Release 11i and will be removed in a future release.

Product #: Change Dependent Product IDs To change which copy of a dependent product a given 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 the same set of books.

- You cannot change the dependent product user ID for SOA products.

  Additional Information: Product Tablespace Requirements in Chapter 6

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 Architecture.

> **Additional Information:** Upgrading Multiple Sets of Books Architecture Databases, and Verifying Dependencies in Multiple Sets of Books Architecture, 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.

AutoUpgrade prompts for an ORACLE username and password. The username must be unique. The new entry for this product is added to the bottom of the list of products displayed on the Select Products screen. The upgrade 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 under these conditions:

- You have not installed the product previously
- The upgrade action is None
- The Select Products screen lists another entry for this product

Verifying Dependencies in Multiple Sets of Books Architecture

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 one set of books, its dependent product Oracle Payables must be installed in the same set of books.

A product installed in your sets of books may acquire a new dependent product in a new release. If the dependent product is required in a set of books being upgraded, and it was not copied there automatically by AutoUpgrade, you must copy the product manually and associate the copy with the fully installed products in that set of books.

You do not have to copy Oracle Applications SOA products for use as dependent products. A product that shares one of these products can use the main installation of that product.

> **Additional Information:** Product Tablespace Requirements in Chapter 6

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

```
AutoUpgrade 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.
```

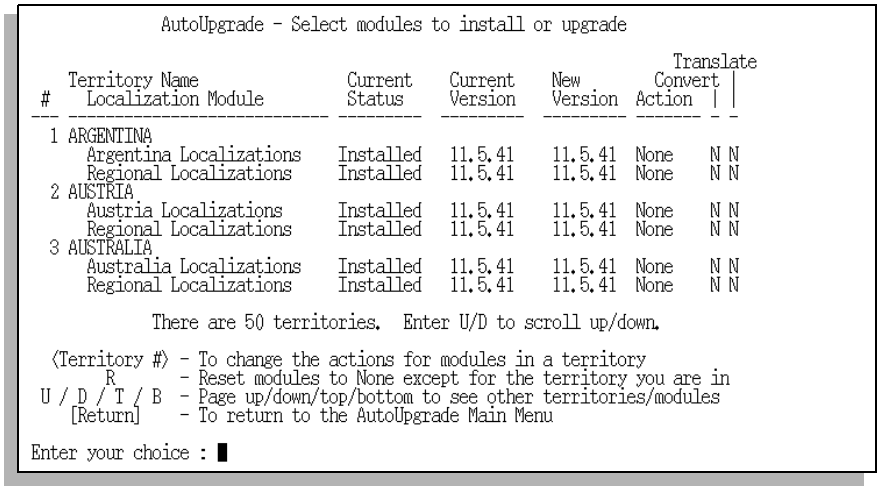
In this case, you need to create a new entry for Oracle Payables and make it a dependent product of PO2. If AP2 already exists, you can go on to Step 4.

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.

> **Note:** The Localization products cannot be altered on the "Select Products" screen. You can use the "Select Modules" screen to modify the actions for these products.

Select Modules Screen

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



You select the localization modules you want to license on this screen, not on the Select Products screen. You can use the Product Detail screen to determine where Rapid Install unloaded the localization files.

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

> **Additional Information:** Select Products Screen, and Product Detail Screens in this chapter

Use this screen to verify that AutoUpgrade has set the upgrade action correctly for each module. Whether licensing or upgrading, you can use this screen to enter the territory number to license 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.

Lists the module numbers. Select the territory # to license the localization modules required.

Territory Name - Localization Module Lists the names of the Oracle Applications modules by territory.

Current Status and Current Version 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. Localization modules must be either fully installed or not installed. Localization modules are not installed as shared.

New Version Lists the version numbers for the localization modules in the new software release.

Action Tells what AutoUpgrade will do with the module.

Convert Always set to N in Release 11.5 and cannot be reset.

Translate Set to Y if you currently have non-US languages installed in your database.

Database Parameters Screen

Use the Database Parameters screen to control the placement and sizing factor of products in the Oracle8i Server database.

```
                AutoUpgrade - Choose database parameters

                              - O -    - S -   --- M ---  --- I ---  --- D ---
  Product            Action  ORACLE   Sizing  Main       Index      Default
 # Name                   |  User ID  Factor  Tablespace Tablespace Tablespace
-- ---------------------  -  -------  ------  ---------- ---------- ----------
 1 Application Object Lib     APPLSYS     100  USER_DATA  USER_IDX   USER_DATA
 2 Application Utilities      APPLSYS     100  USER_DATA  USER_IDX   USER_DATA
 3 Applications DBA           APPLSYS     100  USER_DATA  USER_IDX   USER_DATA
 4 Oracle Alert               ALR         100  USER_DATA  USER_IDX   USER_DATA
 5 Global Accounting Engi     AX          100  USER_DATA  USER_IDX   USER_DATA
 6 Oracle Common Modules-     AK          100  USER_DATA  USER_IDX   USER_DATA
 7 Oracle Common Accounti     XLA         100  USER_DATA  USER_IDX   USER_DATA
 8 Oracle General Ledger      GL          100  USER_DATA  USER_IDX   USER_DATA

      There are 149 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 AutoUpgrade 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.

Use this screen to:

1. Change the default ORACLE user ID and password for each product with the O option. We do not recommend that you change the default Oracle schema, but you can change the default password if you wish.
2. Verify that tablespaces for existing products are set correctly.
3. Set the sizing factor for new objects for a product (or for new products).
4. Specify the tablespaces for each new product you are licensing with the M, I, and D options.

   Additional Information: Product Tablespace Requirements in Chapter 6

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 upgrade action for the product:

C	convert
I	install (license)
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 may be able to change upgrade 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 Upgrade Actions in this chapter

Option O: ORACLE User ID This column lists the ORACLE username that owns each product's database objects. You cannot use the O option to change a product's user ID (username and password) when you are upgrading 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 (up to 30 characters). Then enter the new password. To change only the password, press [Return] when AutoUpgrade 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 Application Object Library (AOL)
  - Oracle Applications DBA (AD)
  - Oracle Applications Utilities (AU)
  - Oracle Applications Shared Technology (SHT)

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

  > **Additional Information:** Changing Oracle Applications Passwords in Chapter 6

- 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, AutoUpgrade automatically changes the schema for the remaining ones.

  > **Note:** Except for these Oracle Human Resources products, each product should be installed under its own separate schema. In prior releases, some products that were new to that release, were installed along with a base product. When upgrading Oracle Applications, these products are forced to be installed in the same schema as that of the base product.

- 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 AutoUpgrade 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.

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*; Product Tablespace Requirements in Chapter 6

> **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 Oracle8i tablespaces in which AutoUpgrade 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 upgrade. AutoUpgrade verifies tablespace names when you enter them, and when returning to the Main Menu.

If you specify new main or index tablespaces when upgrading, AutoUpgrade 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 AutoUpgrade Tasks screen to choose the tasks that AutoUpgrade will perform during an upgrade.

```
          AutoUpgrade - Choose overall tasks and their parameters

 # Task                                          Do it?   Parameters
-- ---------------------------------------------  ------  ----------
 1 Check init.ora parameters                      YES
 2 Create Applications environment file           YES        apdb1.env
 3 Verify files necessary for install/upgrade     YES
 4 Check SYS.DUAL table                           YES
 5 Install or upgrade database objects            YES
 6 Load US data or other language data            YES
 7 Compile flexfield data in AOL tables           YES
 8 Generate message files                         YES
 9 Compile APPS schema(s)                         YES


      There are 9 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 AutoUpgrade Main Menu

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

The task default values are 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 may be multiple pages to this screen. Be sure to review the setting of every task. You can use AD Administration to run many of the same tasks to maintain products after you have finished upgrading the database using AutoUpgrade.

  Additional Information: AD Administration in Chapter 2

Setting Tasks

This section lists AutoUpgrade 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 parameter for creating an environment file.

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

1. Check init.ora parameters

   When set to YES, AutoUpgrade checks the database initialization (init.ora) parameters when you start your 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 the Initialization Parameters section of Chapter 6.

   Additional Information: Running the Upgrade in this chapter

2. Create Applications environment file

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

   The default file name is <db_name>.env, where <db_name> is the name of the database in which the current product group is installed. The default is <db_name>.cmd for NT users.

3. Verify files necessary for upgrade

Verifies that all files necessary to upgrade Oracle Applications are present. Once verified, AutoUpgrade prompts you to continue.

> **Attention:** If there are any missing files, when verifying files required for upgrade, AutoUpgrade will stop.

4. Check SYS.DUAL table

   The SYS.DUAL table is necessary to run Oracle Applications. This task checks for the existence of SYS.DUAL, verifies that only one row exists, and confirms proper permissions to access the table.

5. Upgrade database objects

   Upgrades the product group's database objects.

6. 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 licensing or upgrading database objects. Otherwise you will not be able to use Oracle Applications in any language.**

7. 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 AutoUpgrade, each is compiled the first time a user accesses the flexfield.

8. 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 users and for some concurrent requests.

9. Compile APPS schema(s)

Attempts to compile all invalid objects in the APPS schema(s). This task runs in parallel mode. AutoUpgrade lists any invalid objects in the AutoUpgrade worker log file (adwork*xx*.log). It will not stop if there are invalid objects.

> **Note:** You may see some invalid packages at this point in the upgrade. This is normal, and will usually be resolved after performing the post-upgrade steps.

The defaults for tasks vary depending on the types of files installed in the current APPL_TOP. The following table lists default values for these 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.

Task		A	W	F	C
1.	Check init.ora parameters	✓	x	x	x
2.	Create Applications environment file	✓	✓	✓	✓
3.	Verify files necessary for upgrade	✓	x	x	x
4.	Check SYS.DUAL table	✓	x	x	x
5.	Upgrade database objects	✓	x	x	x
6.	Load US data or other language data	✓	x	x	x
7.	Compile flexfield data in AOL tables	✓	x	x	x
8.	Generate message files	✓	✓	✓	✓
9.	Compile APPS schema(s)	✓	x	x	x

Upgrading Database Objects (Database Tasks)

The tasks you can perform in AutoUpgrade are divided into two categories: database tasks and generation tasks. You can perform both tasks at once, or you can separate them into multiple AutoUpgrade sessions, which you will need to do if you are running in a multi-server environment.

To perform both 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 upgrading or licensing in your current APPL_TOP. If you choose to run separate AutoUpgrade sessions for the two categories of tasks, you *must* perform the database-related tasks first, then the generation tasks.

If you previously stopped an AutoUpgrade session before starting the database-related tasks, restart AutoUpgrade, 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)

Once you have upgraded or installed database objects, you can perform the generation tasks.

Upgrading Files Needed for Runtime (Generation Tasks)

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

Restart AutoUpgrade, 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 on 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:

- ❑ Create environment file
- ❑ Verify files necessary for upgrade
- ❑ Generate message files

Once you have completed these tasks, you may exit AutoUpgrade without saving your restart files.

Attention: Once you have completed your upgrade, you must use AD Administration to perform any maintenance tasks.

Additional Information: Maintaining Applications Files in Chapter 2

Running the Upgrade

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

The tasks you set on the Tasks screen determine which of the following questions AutoUpgrade asks. After you answer all the questions (that your tasks require), AutoUpgrade automatically starts the upgrade.

Environment File

If you chose to create an environment file, AutoUpgrade asks a series of questions about parallel concurrent processing, file-naming conventions, directory paths for log, output, and temporary files, and so on.

Additional Information: Creating an Environment File in Chapter 1

Number of Workers

If you are upgrading, translating, converting, or compiling database objects, AutoUpgrade launches multiple worker processes to perform these steps in parallel. It automatically determines the default value for the number of workers as being two plus the number of CPUs on the machine where your database server is running. For example, on single-processor machines, the default is 3.

AutoUpgrade prompts for the number of workers that you want to use. Choose one or two more than the number of CPUs on your database server machine — up to 99

workers. It is inefficient to have more than two workers over the number of CPUs on your database server because the CPU may waste time switching between the processes.

Initialization Parameters

If you chose to verify initialization (init.ora) parameters, AutoUpgrade verifies the parameters and displays a warning message if one or more parameters have incorrect values:

```
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 at lease one parameter value is incorrect, press [Return] to quit. Change the initialization parameter(s) to the recommended value(s), then shut down and restart the database. You can then restart AutoUpgrade and continue your session.

Warning: AutoUpgrade may fail during the upgrade if the initialization parameters are not set correctly.

Additional Information: Initialization Parameters in Chapter 6

Verifying Files

If you selected the task to Verify Files Needed For Install/Upgrade, AutoUpgrade 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] :
```

This log file is stored in the APPL_TOP/admin/<db_name>/out directory. The default file name is adiuvf.lst (files needed for upgrade). The list of missing files can also be found in the AutoUpgrade log file (adaimgr.log).

If any files needed for the upgrade are missing, AutoUpgrade displays a failure message and exits. You should correct the problem and then restart AutoUpgrade.

Monitoring AutoUpgrade

Monitor AutoUpgrade as it runs to check for problems with the upgrade. See the Monitoring AutoUpgrade section later in the chapter for details.

Exiting AutoUpgrade

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

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

Monitoring AutoUpgrade

You need to monitor the AutoUpgrade process for manager and worker messages and log and restart files. This section describes the AutoUpgrade status messages, tells you where to find AutoUpgrade log and restart files, and explains how to correct AutoUpgrade errors.

Manager and Worker Messages

AutoUpgrade acts as a *manager* that coordinates a number of *workers*, assigning them jobs that they run to 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 if a worker cannot complete its job. For example:

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

In this example, worker 1 failed on the file afform.drv 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.

Your AutoUpgrade 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.

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 AutoUpgrade actions far more extensively than the messages you see on the screen. You will need to review the log files if AutoUpgrade encounters errors.

> **Additional Information:** Correcting AutoUpgrade Errors in this chapter

Manager Log and Restart Files

The manager has a log file that you name at the beginning of the AutoUpgrade 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. NT users will find this file in $APPL_TOP$\admin\<db_name>\log, where <db_name> is the value of ORACLE_SID or LOCAL.

The manager uses the following restart files, which reside in $APPL_TOP/admin/<db_name>/restart or in %APPL_TOP%\admin\<db_name>\restart on NT:

airfinit.rf9	the restart file that holds your responses to the initial AutoUpgrade questions
airfprod.rf9	the product restart file that records each product's upgrade action, user ID, tablespaces, and shared dependencies
airfmod.rf9	the module restart file that records localization modules to be upgraded or licensed
airftask.rf9	the task restart file that records the upgrade tasks you chose to perform
airfmain.rf9	the main restart file that records which upgrade steps have been completed

airftodo.rf9 the "to do" file that records which upgrade steps are left

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

Warning: Do not modify or delete any manager restart files unless specifically told to do so by Oracle Support Services.

Attention: If AutoUpgrade fails and you choose to continue past the error, you must correct the problem manually. If you do not correct the problem, AutoUpgrade may not be able to continue. If you encounter difficulties in resolving the problem, contact Oracle Support Services.

By default, AutoUpgrade deletes these files (but leaves backup versions with the extensions .bak, .bk2, or .bk3) when the upgrade finishes. AutoUpgrade does not delete these files if you answer No to the prompt:

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

Worker Log and Restart Files

Worker log files reside in the directory $APPL_TOP/admin/<db_name>/log, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. NT users will find it in %APPL_TOP%\admin\<db_name>\log, where <db_name> is the value of ORACLE_SID or LOCAL. The files are named adwork*xx*.log, where *xx* indicates the worker number. For example, worker 1 uses the log file adwork01.log.

Each worker may also have a restart file called adwork*xx*.rf9. These files are stored in $APPL_TOP/admin/<db_name>/restart or in %APPL_TOP%\admin\<db_name> \restart on NT. 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 AutoUpgrade 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 ORACLE_HOME directory. Also ensure that APPL_TOP is set to the top Applications directory.

2. Run the environment file. (This step does not apply to NT users.)

   Rapid Install creates the product group's environment file when laying down the new APPL_TOP. Load this file into your current environment with the following command:

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

   Here is an example:

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

3. Split or copy the worker log file.

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

   For UNIX users:

   ```
   $ 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
   ```

   For NT users:

   ```
   C:\> cd %APPL_TOP%\admin\apptest\log
   C:\> dir adwork*.log

   Jan  6 17:11     9151702       adwork01.log
   Jan  6 17:07     4387663       adwork02.log
   Jan  6 17:11       3017393         adwork03.log
   ```

   The sample response 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:

For UNIX users:

```
$ tail -100 adwork<xx>.log > <filename>.log
```

Here is an example:

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

The tail command in the example writes the last 100 lines of the worker log file to the specified file name.

> **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 file name. Here is an example:

For UNIX users:

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

For NT users:

```
C:\> copy adwork02.log edit02.log
```

Review the copied log file (for example, edit02.log). 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 understand what to do.

6. Run AD Controller (adctrl) to restart the failed job.

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

   Additional Information: AD Controller (adctrl) in Chapter 5

Note: If you are upgrading a database that contains more than one language, you may run out of space in some tables during the upgrade. Contact Oracle Support Services if this happens.

Acceptable Errors

If a failure occurs while AutoUpgrade is running in parallel mode, the AutoUpgrade worker will fail and record an error message in the worker log file. AutoUpgrade 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 AD Controller (adctrl). Contact Oracle Support Services if the worker encounters the same error while running the job again.

Additional Information: AD Controller (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 easier to rerun:

```
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*i* Error Messages guide

If you consistently receive ORA-1555 errors, the problem may be due to insufficient rollback space. Try increasing the size of your rollback segments or adding additional rollback segments before restarting the workers.

Additional Information: Rollback Segments in *Oracle8i Server Organization and Requirements*; *Oracle Applications Installation Update*

Database Object Differences

Several different types of warnings may be reported when AutoUpgrade 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.
```

AutoUpgrade 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. 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 upgrade to determine whether you need to modify your customizations.

Import Error Messages

AutoUpgrade may run the Oracle Import utility (IMP) when 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 AutoUpgrade or the AutoUpgrade 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 Compilation Error Message

Toward the end of your AutoUpgrade session, you may receive a flexfield compiler error like this:

For UNIX users:

```
Compiling all application flexfields.

/d01/appl/115/fnd/11.5.0/bin/fdfcmp APPLSYS/FND 0 Y
Log filename    : /d01/appl/115/admin/apptest/log/l460306.req
Report filename : /d01/appl/115/admin/apptest/out/SYSADMIN.460306

An error occurred while compiling application flexfields.
Continue as if it were successful [No] :
```

For NT users:

```
Compiling all application flexfields.

C:\>\appl\115\fnd\11.5.0\bin\fdfcmp APPLSYS/FND 0 Y
Log filename    : C:\>\appl\115\admin\apptest\log\l460306.req
Report filename : C:\>\appl\115\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 AutoUpgrade tasks while you determine the cause of the problem. To investigate the problem, inspect the flexfield 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-upgrade steps are complete and examine the flexfields for setup errors. You can recompile flexfields using AD Administration after you have finished upgrading the database. Flexfields also compile automatically when they are accessed for the first time.

> **Additional Information:** Define Descriptive Flexfield Segments, and Define Key Flexfield Segments, *Oracle Applications Flexfields Guide*

After Running AutoUpgrade

After you run AutoUpgrade, perform the post-upgrade steps immediately.

Finishing Your Upgrade

You need to perform manual post-upgrade steps that complete the upgrade. This brings your database back into service as soon as possible.

> **Additional Information:** Chapters 5 - 8, *Upgrading Oracle Applications*

Upgrading Another Product Group

If you plan to upgrade another product group, make sure to follow the steps in Chapters 1 - 4 of *Upgrading Oracle Applications,* to prepare your system for that upgrade before you run AutoUpgrade again.

4

AutoPatch

This chapter contains information about using the AutoPatch utility (adpatch). It contains the following sections:

- What is AutoPatch?
- Running AutoPatch
- Starting AutoPatch
- After Running AutoPatch
- Running AutoPatch Again
- Controlling AutoPatch Behavior
- Adding Translations
- Applying Java Patches

What is AutoPatch?

AutoPatch is a utility used to apply individual patches, mini-packs, or maintenance packs. A *mini-pack* (known as a patchset in previous releases) is a collection of individual patches for a product, while a *maintenance pack* (also known as a release update) is a collection of mini-packs for all Applications products. In regard to Oracle Applications release numbering convention, Release 11.5 is a minor release and is installed using Rapid Install. Release 11.5.2 would be a maintenance pack and be installed with AutoPatch.

AutoPatch may ask extra questions and do extra checking when you apply a maintenance pack. After you apply the maintenance pack, it updates the information in your database to indicate the current release update level.

> **Attention:** In an MSOBA database, if a new set of books is added, the database portion of all patches must be re-applied to the new set of books. You may not use AutoPatch for custom development.

In addition to maintaining existing products, AutoPatch is used for additional tasks such as adding a language or a new product that was not a part of the base release. AutoPatch resides in the AD_TOP/bin directory.

How It Works

AutoPatch replaces some of the existing product files with new versions of those files that are included with the patch. It may also make changes to your Oracle Applications database objects.

If you are installing Oracle Applications in a multi-server environment, you must run AutoPatch on all relevant machines to install the necessary files.

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].

In preparation for running AutoPatch, you perform these steps:

1. Log in as applmgr and set up your environment.
2. Copy the patch files to a directory on your file system by unzipping the archived patch file. The directory created by unzipping the archived patch file is called the PATCH_TOP directory.
3. Start AutoPatch from a Bourne or Korn shell.

> **Note:** See the *Installation Update* for your platform for recommended shells.

4. Answer the AutoPatch questions. AutoPatch identifies the Oracle Applications products that need to be updated.

   Additional Information: Running AutoPatch in this chapter

Once you have answered the AutoPatch questions, the utility performs these steps:

1. Extracts the appropriate files from each product's C library.
2. 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.
3. 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 into a subdirectory of your patch directory. Specifically, it backs up

   <PROD>_TOP/<subdir(s)>/<old_file_name> to
   <patch_dir>/backup/<env_name>/<appl_top_name>/<prod>/<subdir(s)>/<old_file_name>.

   Where <patch_dir> is the patch directory, <env_name> is the Applications Environment name, <appl_top_name> is the APPL_TOP name, and <prod> is the name of the product being patched.

   Additional Information: Before Running AutoPatch in this chapter

4. Replaces each product's outdated files with newer files from the patch directory.
5. Applies changed Java class files and regenerates JAR files as needed.
6. Loads the new object modules into the C libraries.
7. Backs up any files you listed in adlinkbk.txt and is relinked.
8. Relinks the Oracle Applications products with the Oracle8 Server.
9. Runs SQL scripts and exec commands, which change Oracle Applications database objects. By default, AutoPatch does this in parallel.
10. Copies any specified HTML or media files to their respective destinations.
11. Generates Oracle Forms files.
12. Generates Oracle Reports files.
13. Generates Oracle Graphics files.

14. Appends a record of how it changed your system to applptch.txt in the $APPL_TOP/admin/<db_name> directory, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. For NT, the file is located in %APPL_TOP%\admin\<db_name>, where <db_name> is the value of your ORACLE_SID or LOCAL variable.

15. Records summary information of actions actually performed to applpsum.txt located under APPL_TOP/admin.

Running AutoPatch in a Multi-server Environment

If you are installing Oracle Applications in a multi-server environment, you must run AutoPatch on every node to install the necessary files. You need to run AutoPatch only once, on the administration server, to update database objects, but you must run the file and generation portions on all nodes that require changed files. Specifically:

- Run the file portion of the patch (c<patchnum>.drv) on every node containing one or more of the files being replaced by the patch or, if in doubt, apply the patch on all servers
- Run the database portion of the patch (d<patchnum>.drv) once, from the administration server only
- Run the generation portion of the patch (g<patchnum>.drv) on every node containing one or more of the files to be generated, or, if in doubt, apply the patch on all servers

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 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 cannot run multiple sessions of AutoPatch simultaneously against a product group. You may run AutoPatch concurrently on separate nodes, but only one session may be performing database or generation tasks.

Patch Format

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 <patchnum>, where <patchnum> corresponds to the Oracle Bug Database number for the patch.

The important files in the top level directory are: readme.txt, c<patchnum>.drv, d<patchnum>.drv, g<patchnum>.drv, and in the case of Java patches, j<patchnum>.zip.

The readme.txt file contains general information about the patch. It may describe manual steps that you must perform as part of fixing the problem, 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<patchnum>.drv file contains commands for copying files and linking executables. You use AutoPatch to apply the file portion of the patch (unless the readme.txt says otherwise). The file portion of the patch changes your Oracle Applications files. In a multi-server environment, you should run this patch driver on all servers containing one or more of the files being replaced by the patch or, if in doubt, apply the patch on all servers.

You run the file d<patchnum>.drv using AutoPatch to apply the database portion of the bug patch. The database portion of the patch changes your Oracle Applications database objects. A d<patchnum>.drv file is only included if the patch requires changes to your Oracle Applications database objects, and if these changes can be easily automated. You must successfully run c<patchnum>.drv using AutoPatch before running d<patchnum>.drv. This driver file should only be run on your administration server.

> **Note:** After you run the d<patchnum>.drv, check for invalid objects in the database. If there are invalid objects, run the Compile Apps Schema option from AD Administration before you run the g<patchnum>.drv. Failure to do so may cause the g<patchnum>.drv to fail.

The g<patchnum>.drv file contains generation steps, and must be run after the file and database portions of the patch have been run. A g<patchnum>.drv file is only included if the patch requires new forms, reports, graphics, 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 are copied to 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-server environment, you should apply the patch on all servers.**

readme.txt Details

The readme.txt entry for a single patch lists:

- Bug number

  Each patch entry starts with a line giving the bug number and product that the patch fixes. For example:

```
oooooooooooooooooooooooo
o                      o
o  BUG  fnd  691976    o
o                      o
oooooooooooooooooooooooo
```

- Patch description

  A description of what the patch does. It may list the files included in the 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.

  The patch drivers should be run with AutoPatch in the following order:

  1. c<patchnum>.drv
  2. d<patchnum>.drv

3. g<patchnum>.drv

   Always check the readme.txt file for the correct order in running patch drivers. Many patches do not have database (d<patchnum>.drv) or generation (g<patchnum>.drv) driver files. You should only try to run the database and / or generation driver files if they exist.

Integrated Patches

If the patch you are applying has prerequisite fixes, Oracle ships you an integrated patch containing both the specific patch you requested and all prerequisite patches. In this case, you may find multiple entries in your readme.txt—one entry for each bug included in the patch.

Running AutoPatch

Before starting AutoPatch, review and complete the Running AD Utilities section in Chapter 1.

Perform the following steps after completing the preparation steps.

1. Back up any previously patched or customized files that you want to save. In prior releases, the file names of previously patched files ended with *O*, such as *xxxO*. With Release 11*i* all files being patched are backed up by AutoPatch into a subdirectory of the patch directory. The subdirectory is determined by the Application environment name and the APPL_TOP name entered during the initial AutoUpgrade startup questions.

   Attention: Be sure there is enough disk space for the backups AutoPatch will make. AutoPatch generally requires two times the size of the patch, mini-pack, or maintenance pack being applied (e.g. if the size of the patch is 10MB, then allocate 20MB of free disk space, at a minimum.

   Additional Information: Running AD Utilities in Chapter 1

2. 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.

For NT users:

3. Ensure that %JAVA_TOP%\apps.zip and %JAVA_TOP%\loadjava.zip are included in the set classpath statement of %APPL_TOP%\admin\adovars.cmd.
4. Verify that the gnu make home is in the path and is called gnumake.exe not make.exe.
5. Verify that the MKS toolkit and gnumake appear before the WINNT and WINNT/System32 directories in the system path.

Starting AutoPatch

Start the utility from the patch directory, which is the directory where you unloaded the AutoPatch files. Use the *adpatch* command.

For UNIX users:

```
$ adpatch
```

For NT users:

```
C:\> 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

Responding to Prompts

AutoPatch asks the same questions as AD Administration. Use this section and the Responding to Prompts section in Chapter 1 to guide you through the additional information that AutoPatch requires.

AutoPatch log file

The default log file name is adpatch.log, which resides in $APPL_TOP/admin/<db_name>/log, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. NT users will find this file in %APPL_TOP%\admin\<db_name>\log, where <db_name> is the value of ORACLE_SID or LOCAL. We recommend renaming the log file name to reflect the patch being applied.

> **Note:** When applying a patch, we recommend you use the associated driver file name as the basis for your log file. For example: d123456.log.

SYSTEM and AOL user passwords

In addition to verifying the connection and obtaining prerequisite information from the database, AutoPatch validates the password information for all Oracle Applications schemas.

Patch directory

AutoPatch asks you to specify the directory that holds the patch driver file. The default is the directory from which you started AutoPatch. If necessary, change the full path name to the directory where you unloaded the patch files.

Patch driver file

AutoPatch prompts for the patch driver file, which is typically c<patchnum>.drv in the patch directory. If you are using a different driver file, enter the file name at the prompt. Enter the full path name and file name if the file is not in the patch directory. AutoPatch checks the integrity of the patch driver to ensure it is valid, and determines what patches in the file need to be applied on your site.

If you are applying a maintenance pack (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 to help 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

By default, AutoPatch runs exec and sql commands in parallel mode and prompts you for the number of parallel workers:

```
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 AutoUpgrade parallel mode — AutoPatch starts workers, the workers run tasks they are assigned, and the workers eventually finish. Then, AutoPatch starts running again, does a few more

tasks, and exits. You can use AD Controller (adctrl) to monitor and alter AutoPatch parallel worker status just as you would with AutoUpgrade.

Note: AutoPatch runs all database actions based on phase order, which is not necessarily the order in which the commands are listed in the database patch driver.

Additional Information: AD Controller (adctrl) in Chapter 5

AutoPatch Messages

While it is running, you need to monitor AutoPatch to check for error messages. The following sections explain the types of messages you may see.

Informational Messages

Informational messages are written to an informational log file. This file resides in the same location as your AutoPatch log file. It has the same base file name, but with an .lgi extension instead of a .log extension. For example, if your AutoPatch log file is named d123456.log, your AutoPatch informational log file is named d123456.lgi.

The informational log file contains output such as the following:

For UNIX users:

```
Will not apply POXPOPAA.rdf: Files are identical.
  Patch  : /d01/appl/patch/po/reports/POXPOPAA.rdf, v115.3
  On-Site: /d01/appl/115/po/11.5.0/reports/POXPOPAA.rdf, v115.3

Not running file 'wip patch/115/sql wipmlprb.pls' against schema 'apptest'
because the corresponding PL/SQL object in the database is up to date.
Revision in File     = 115.8
Revision in Database = 115.8
```

For NT users:

```
Will not apply POXPOPAA.rdf: Files are identical.
  Patch  : C:\appl\patch\po\reports\POXPOPAA.rdf, v115.3
  On-Site: C:\appl\115\po\11.5.0\reports\POXPOPAA.rdf, v115.3

Not running file 'wip patch\115\sql wipmlprb.pls' against schema 'apptest'
because the corresponding PL/SQL object in the database is up to date.
Revision in File     = 115.8
Revision in Database = 115.8
```

These messages are normal. Typically, they are produced if you run AutoPatch more than once because AutoPatch does not update files that it has previously updated. 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 applcust.txt in the admin directory under your APPL_TOP, AutoPatch reviews 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 the patch is being applied in parallel mode, AutoPatch operates with the number of parallel worker processes that you specified. When AutoPatch fails before or after the worker processes, a message appears asking whether to continue. At this point it is best to review the log files to determine the source of the error. Once the error is

resolved, you can restart AutoPatch. If an error or a problem cannot be resolved, you should:

- Verify that all steps in the readme file were completed
- Check the Metalink site for additional information regarding the patch being applied
- Call Oracle Support

Failure During Worker Processes

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.

- Log in as applmgr from another terminal or terminal window and verify the environment
- Run the environment file
- Split or copy the worker log file. This prevents errors if the worker tries to write to its log file while you are reviewing the file
- Review the end of the log file to find the problem
- Fix the problem
- Restart the failed job using the AD Controller utility

  Additional Information: AD Controller (adctrl) in Chapter 5

Contact Oracle Support Services if you cannot determine or resolve the problem.

Additional Information: After Running AutoPatch in this chapter

Successful Completion Message

AutoPatch displays a message like the following once it runs to completion:

For UNIX users:

```
A job timing report has been generated for the current session. You
should check the file
/d01/appl/115/admin/apptest/out/adt01302.lst
for details.

Purging timing information for prior sessions.
```

```
sqlplus -s APPS/APPS
@/d01/appl/115/admin/apptest/ad/11.5.0/admin/sql/adtpurge.sql 10 1000

Done purging timing information for prior sessions.

AutoPatch is complete.

AutoPatch may have written informational messages to the file
/d01/appl/115/admin/apptest/log/adpatch.lgi

You should check the file
/d01/appl/115/admin/apptest/log/adpatch.log

for errors.
```

For NT users:

```
A job timing report has been generated for the current session. You
should check the file
C:\appl\115\admin\apptest\out\adt01302.lst
for details.

Purging timing information for prior sessions.

sqlplus -s APPS/APPS
@C:\appl\115\admin\apptest\ad\11.5.0\admin\sql\adtpurge.sql 10 1000

Done purging timing information for prior sessions.

AutoPatch is complete.

AutoPatch may have written informational messages to the file
C:\appl\115\admin\apptest\log\adpatch.lgi

You should check the file
C:\appl\115\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

Search for the words *Error*, *error*, *ERROR*, *Warning*, *warning*, and *WARNING* in the AutoPatch log files even if it appears that AutoPatch has not encountered any errors. Some warnings are diagnostic — they do not indicate errors and can be safely 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 actions. If AutoPatch does not perform a certain action, it does not generate the log file that records that type of action. The log directory, may, therefore, contain one or more of the following AutoPatch log files.

Log File	Description
adpatch.log	main AutoPatch log file (default name)
adrelink.log	for relinking
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
adwork*xx*.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 admin directory is not already restricted.

Review Customizations

If AutoPatch displayed a message indicating that you have previously registered customized files that will be replaced by the patch, you should review those files now to determine if your customizations need to be re-applied or merged.

AutoPatch Backup Files

After you update all product groups successfully and properly test the system, you can delete the backup copies of files from the patch subdirectory trees to recover disk space. However, we recommend retaining these files if space is available. You

should check for customized files before deleting these backups. You register customized files in the applcust.txt file in the admin directory under your APPL_TOP. If these backups contain customizations, you may want to reapply them before you delete the backup copies.

Additional Information: Recovering Disk Space, in Chapter 7

The file applptch.txt in the $APPL_TOP/admin/<db_name> directory contains information on which files have changed. NT users will find it in %APPL_TOP%\admin\<db_name>.

Additional Information: Patch History File in this chapter

Maintain Multiple Reporting Currencies Schema(s)

If you are using Multiple Reporting Currencies functionality, you must re-run the option to maintain your MRC schema(s) from AD Administration after you run any patch driver file that updates database objects. AutoPatch displays a reminder message when the patch is completed if it detects this 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: Database Tasks in Chapter 2

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. These scripts are located in AD_TOP/sql.

Additional Information: Pinning Packages and Sequences in Chapter 6

Attention: After running AutoPatch on the web server, you must shut down and restart the web server. It does not need to be shut down before applying the patch; it only needs to be restarted when the patch is finished.

Running AutoPatch Again

Make sure that you run AutoPatch as many times as necessary until it runs to completion. AutoPatch can be run as many times as necessary until the patch is successfully applied. You can then run AutoPatch 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:

For UNIX users:

```
$ adpatch
```

For NT users:

```
C:\> 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 file name, 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 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 session is complete before you resume in the current environment. If a previous patch session did not complete, you should resume applying that patch before you continue with a new one.

Attention: Applications patches must *always* be applied in their entirety.

Note: AutoPatch may not work properly if you partially applied a copy driver and then try to reapply the same copy driver from the beginning.

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 Architecture database.

However, you do not need to run AutoPatch for each one of your product groups if *all* the following criteria apply:

- 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 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 in to 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

There are several ways of controlling the way AutoPatch applies patches. You can run AutoPatch in test mode, pre-install mode, or non-interactive mode.

> **Attention:** Before applying any Oracle Applications patch, regardless of the mode utilized, you should read the README file (usually called readme.txt) supplied with the patch. You should also read the documentation supplied with the patch (if any).

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.
- archive any object modules into your product libraries.
- relink any executables.
- generate any forms, reports, PL/SQL libraries, or menu files.
- run any sql or exec commands (commands that change the database).
- update the Patch History File (applptch.txt).

- 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-install mode.

Pre-install Mode

You can use AutoPatch to apply patches after running Rapid Install and before running AutoUpgrade to upgrade Oracle Applications. This is useful for applying patches to AutoUpgrade itself, to AutoPatch, or to other AD Utilities before running AutoUpgrade. Be sure to read the Applications environment file to set up the environment before you run AutoPatch.

To run AutoPatch before you run AutoUpgrade, you need to run AutoPatch in pre-install mode. To do this, you must include a special command line argument *preinstall=y*. For example:

```
$ adpatch preinstall=y
```

When you run in pre-install mode, AutoPatch asks all of the normal startup questions, except those relating to the database. Use pre-install mode only when running AutoPatch on your administration server.

Take note of the following restrictions for using pre-install mode. AutoPatch *does not*:

- run SQL scripts or exec commands, or generate any files, such as forms or reports, since these actions require AutoPatch to connect to a database.
- read the Oracle Applications product driver files to determine valid on-site files.
- relink any executables, except for AOL and AD.

- apply Release Updates. It only applies individual bug patches.

Other than these restrictions, AutoPatch operation is the same in pre-install as it is in normal operation.

> **Note:** Because AutoPatch does not read driver files in pre-install 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.

Non-interactive Mode

AutoPatch can run all the patch drivers in a patch without user intervention. When you specify a patchtop location, AutoPatch locates and runs all patch drivers that are specific to the patch being applied.

Creating a Defaults File

Before you can run AutoPatch non-interactively, you must first create an AutoPatch defaults file for your current environment (APPL_TOP and ORACLE_SID/TWO_TASK, or ORACLE_SID/LOCAL for NT). To create an AutoPatch defaults file:

1. Specify defaultsfile=<Defaults File Name> on the AutoPatch command line. The defaults file must be located under your APPL_TOP in the <db_name> subdirectory under admin.

   For example:

   For UNIX users:

   ```
   adpatch defaultsfile=$APPL_TOP/admin/testdb1/def.txt
   ```

   For NT users:

   ```
   adpatch defaultsfile=%APPL_TOP%\admin\testdb1\def.txt
   ```

2. Run AutoPatch up to the point where it asks you for the directory where your Oracle Applications patch has been unloaded. Then type 'abort' at this prompt.
3. Verify that your defaults file exists.

Once you have an AutoPatch defaults file for your current environment, you can run AutoPatch non-interactively. Here are a few examples:

Applying a Single Patch Driver File

It is possible to apply just a single patch driver file non-interactively using AutoPatch. Here is an example:

Assume the following:

- Defaults file is $APPL_TOP/admin/testdb1/def.txt ($APPL_TOP%\admin\testdb1\def.txt for NT)
- Applying copy driver for patch 123456, which is located in the patch directory under your APPL_TOP.
- Using three parallel workers
- AutoPatch log file name is 123456.log

The AutoPatch command line would be:

For UNIX users:

```
adpatch defaultsfile=$APPL_TOP/admin/testdb1/def.txt logfile=123456.log \
patchtop=$APPL_TOP/patch/123456 driver=c123456.drv workers=3 interactive=no
```

For NT users:

```
adpatch defaultsfile=%APPL_TOP%\admin\testdb1\def.txt logfile=123456.log \
patchtop=%APPL_TOP%\patch\123456 driver=c123456.drv workers=3 interactive=no
```

You could also turn off the "validate schemas" feature at the same time by using the command line.

For UNIX users:

```
adpatch defaultsfile=$APPL_TOP/admin/testdb1/def.txt logfile=123456.log \
patchtop=$APPL_TOP/patch/123456 driver=c123456.drv workers=3 interactive=no \
options=novalidate
```

For NT users:

```
adpatch defaultsfile=%APPL_TOP%\admin\testdb1\def.txt logfile=123456.log \
patchtop=%APPL_TOP%\patch\123456 driver=c123456.drv workers=3 interactive=no \
options=novalidate
```

This is a good idea if you have run AutoPatch a few times in the current environment and do not expect your current set of Oracle Applications schemas or

their passwords to change. As it takes time to connect to each schema, omitting this step makes AutoPatch run faster.

Applying a Standard Patch

To apply a standard patch to your APPL_TOP and database non-interactively:

- Specify patchtop=<value> on the AutoPatch command line. The patch top value must end in a 6-to-8-digit number. For example, patchtop=$APPL_TOP/patch/123456. Or, for NT, patchtop=%APPL_TOP%\patch\123456.
- Do not specify driver=<value> on the AutoPatch command line

AutoPatch assumes the patch being applied is a standard patch, and it attempts to run the standard patch driver files in the standard order without prompting you for the patch driver file names.

AutoPatch looks for c<patchnum>.drv, d<patchnum>.drv, and g<patchnum>.drv. All other drivers must be run using the non-standard method described in the next section: Applying a Non-standard Patch.

In the example, the AutoPatch command would be:

For UNIX users:

```
adpatch defaultsfile=$APPL_TOP/admin/testdb1/def.txt logfile=patch123456.log \
patchtop=$APPL_TOP/patch/123456 workers=3 interactive=no
```

For NT users:

```
adpatch defaultsfile=%APPL_TOP%\admin\testdb1\def.txt logfile=patch123456.log \
patchtop=%APPL_TOP%\patch\123456 workers=3 interactive=no
```

Applying a Non-standard Patch

Oracle Applications ships some patches that follow the standard patch structure, but do not follow the standard patch naming conventions. Using the AD Merge Patch utility, you can also create patches with a standard structure but non-standard names. Also, Oracle Applications maintenance packs (release updates) normally do not follow standard patch naming conventions. Items that constitute non-standard patch naming conventions are:

- The last component of the patch directory is not a 6-to-8-digit number
- The patch driver files are not named *<Patch Number>.drv

In order to apply such a patch non-interactively, you must supply the following information to AutoPatch:

- The names of the patch driver files
- The order in which the patch driver files should be run (Read the readme.txt to find out the proper order for running the patch driver files).
- Optionally, the types of the patch driver files (copy, database, or generate). This is useful because AutoPatch knows not to run a database driver file unless the current APPL_TOP implements the admin server.

You provide this information to AutoPatch using the AutoPatch command-line keyword `driver=<values>`, where <values> is a comma-separated list of the patch driver files you want AutoPatch to run.

For example, suppose you want AutoPatch to apply the following patch driver files in the order specified:

- my_drv1.drv (copy driver)
- my_drv3.drv (database driver)
- my_drv2.drv (generate driver)

The naming convention used here is to add a c, d, or g at the end of the .drv extension: c is for copy driver, d is for database driver, and g is for generate driver. Specify the `driver=` command-line argument as follows:

```
driver=my_drv1.drvc,my_drv3.drvd,my_drv2.drvg
```

AutoPatch runs my_drv1.drv first, my_drv3.drv second, and my_drv2.drv last (because that is the order you specified).

> **Note:** In this example, AutoPatch understands that you really meant my_drv1.drv, not my_drv1.drvc. However, if you specify a patch driver file with some extension other than drvc, drvd, and drvg, it assumes that you intend it to run the file with exactly that extension — it does not try to re-map the extension back to drv.

Restarting a Non-interactive AutoPatch Session

If you run AutoPatch in a non-interactive session and it encounters an error, it exits to the operating system and returns failure to the operating system. When this occurs, look through the log files, diagnose the error, and fix it.

Once the error is fixed, you cannot restart AutoPatch by executing the original AutoPatch command. To restart a non-interactive AutoPatch session, use the same command-line options you first used, but add *restart=yes*.

> **Attention:** It is critical that you do not omit any of the original command-line arguments, as this would change AutoPatch's behavior and cause unpredictable results on restart.

Here is the command line you would use to restart in our example:

For UNIX users:

```
adpatch defaultsfile=$APPL_TOP/admin/testdb1/def.txt logfile=123456.log \
patchtop=$APPL_TOP/patch/123456 driver=c123456.drv workers=3 \
interactive=no restart=yes
```

For NT users:

```
adpatch defaultsfile=%APPL_TOP%\admin\testdb1\def.txt logfile=123456.log \
patchtop=%APPL_TOP%\patch\123456 driver=c123456.drv workers=3 \
interactive=no restart=yes
```

Do not specify `restart=yes` when starting a new non-interactive AutoPatch session. If there is an old AutoPatch session that did not complete successfully, AutoPatch will run the old AutoPatch session without telling you. It will ignore the PATCH_TOP and patch driver files you specify. Always omit `restart=yes` on the first attempt to run AutoPatch non-interactively. If there is an old AutoPatch session, AutoPatch will fail, and you can decide what to do about the old session before proceeding with the current one.

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 the word *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:

For UNIX users:

```
$ adpatch options=nogenrep,nogenform
```

For NT users:

```
C:\> adpatch options=nogenrep,nogenform
```

Here are the valid keywords and what they do:

This option name...	toggles these actions...
copy	copy commands in the driver file
exec	exec commands in the driver file
exectier	exectier commands in the patch driver file
forcecopy	forcecopy commands in the driver file
genform	genform commands in the driver file
genfpll	genfpll commands in the driver file
genmenu	genmenu commands in the driver file
genmesg	genmesg commands in the patch driver file
genrep	genrep commands in the driver file
genrpll	genrpll commands in the driver file
genwfmsg	genwfmsg commands in the patch driver file
jcopy	jcopy commands in the patch driver file
libin	libin commands in the driver file
libout	libout commands in the driver file
link	link commands in the driver file
makedir	makedir commands in the patch driver file
parallel*	Running actions using parallel workers
sql	sql commands in the driver file
validate	validating the password for each ORACLE schema

* The options=parallel keyword does not affect AutoPatch behavior at all — EXEC and SQL actions either run in parallel or serial mode, and Oracle Forms objects generate in parallel. The options=noparallel keyword causes both EXEC and SQL actions to run in serial (regardless of what the patch driver file says), and also causes Oracle Forms objects to generate in serial.

Using the AD Merge Patch Utility

AD Merge Patch (admrgpch) merges multiple AutoPatch-compatible patches into a single integrated patch. It is an executable located in the bin directory under AD_TOP.

Additional Information: AD Merge Patch in Chapter 5

Patch Driver Files

The patch driver file lists the update actions that AutoPatch performs for 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<patchnum>.drv file and part of a g<patchnum>.drv file. Your driver files have the same form as the following example, but contain different information:

- Excerpt from c11235.drv:

```
begin aru c11235
  characterset us7ascii
  compatible release 11.5.0
  compatible parallel yes
  begin bug   po   10123
    ...
  end bug   po   10123
  begin bug   po   11235
    begin actions
      copy      po   reports    POXPOPAA.rdf   115.3
      copy      po   forms/US   POXSCVAR.fmb   115.6
      copy      po   resource   POXPOPOS.pll   115.16
    end actions
  end bug   po   11235
end aru c11235
```

- Excerpt from g11235.drv:

```
begin aru g11235
  characterset us7ascii
  compatible release 11.5.0
  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
      genfpll   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 patch 11235. If you are applying a maintenance pack (release update) using AutoPatch, you 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 maintenance pack (release update) per patch driver file.

The “characterset 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 11.5.0” statement indicates that this patch is compatible with any Oracle Applications 11.5.x release. For example, AutoPatch does not apply this patch to any Release 11.0 or 11.6 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 maintenance pack (release update). This separate feature is always active when you apply a maintenance pack.

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 file name 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 115.3, from the patch directory, with the version on-site in the reports directory under PO_TOP. If the on-site version is older or missing, AutoPatch copies

the newer file to this directory. Similarly, the second line instructs AutoPatch to compare POXSCVAR.fmb version 115.6 with the version on-site in the forms directory under the US subdirectory in AU_TOP, where all Applications form source files are located. If the on-site version is older or missing, AutoPatch copies the file to this directory. Note that AutoPatch automatically knows when certain file types, such as form source files 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 the one on-site and copy from patch directory to target subdirectory if target is older or missing. The main AutoPatch log file, adpatch.log records these actions.
exec	Run executable program. The main AutoPatch log file and worker log files record these actions.
exectier	Run executable program on specific list of tiers. 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 main AutoPatch log file, adpatch.log records these actions.
genform	Generate Oracle Forms form.
genfpll	Generate Oracle Forms PL/SQL library file.
genmenu	Generate Oracle Forms menu file.
genmesg	Generate Oracle Applications message files.
genogn	Generate Oracle Graphics files.
genrep	Generate Oracle Reports report.
genrpll	Generate the specified Oracle Reports PL/SQL library file.
genwfmsg	Generate Workflow resource files.
jcopy	Update Java class files.
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.

makedir	Create an empty directory.
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/<db_name> directory, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. NT users will find this file in %APPL_TOP%\admin\<db_name>, where <db_name> is the value of ORACLE_SID or LOCAL. It updates the header and appends information to this file each subsequent time it runs. This file records the following information about your AutoPatch session:

- Patches not applied and why
- Patches applied and the following information about each:
  - All actions executed for the patch.
  - All actions 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 11.5.A
CURRENT_RELEASE: 11.5.0
RELEASE_CHANGE_DATE: Thu Feb 17 2000 16:45:37
FILE_CHANGE_DATE: Fri Feb 18 2000 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 is run:

```
#
# Run of AutoPatch, Thu Feb 17 2000 18:44:09
#   ORACLE_SID/TWO_TASK: apptest
#           ORACLE_HOME: /oracle/db/8.0.6
begin aru c11235
  characterset 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   115.6
      copy     po   resource   POXPOPOS.pll   115.16
#
# AutoPatch did not execute the following actions:
#
      copy      po   reports    POXPOPAA.rdf   115.3
    end actions
  end bug po 11235
end aru c11235
#
# End run of AutoPatch, Thu Feb 17 2000 18:44:10
#
```

Patch Summary File

The patch summary file (applpsum.txt) records all patches successfully applied to an APPL_TOP or database in a succinct manner. Information such as when and to which databases a patch was applied is recorded.

> **Note:** Bugs within an ARU that are not applied or an ARU that has no actions will not be recorded.

Patch information in this file is first grouped by ORACLE_HOME and IDENTIFIER (SID or TWO_TASK on UNIX, LOCAL on NT), and then by the session the bug was applied. The most recent patch applied is listed first in each section. The file records the following information:

- ORACLE_HOME
- ORACLE_IDENTIFIER which will be an SID or TWO_TASK on UNIX, or LOCAL on NT
- The time of AutoPatch run (identical information to applptch.txt)
- The bug applied and what product it was applied to

The patch summary file is located in the admin directory under APPL_TOP. If the file does not exist, it is created by AutoPatch.

Adding Translations

You use AutoPatch to add translations. Translations are delivered as patch(es) and can be applied in the same way a regular patch is applied. The patch readme files contain detailed information.

Applying Java Patches

Applying a Java patch is similar to applying a standard patch. The Java patch is made up of a Java source file and the resulting Java executable files. These files, known as a Zipped Resource Unit (ZRU), are zipped into a patch archive along with the necessary driver files and readme files. The ZRU is in the format j<patchnum>.zip, where patchnum is the patch (or bug) number in the ARU system. Once the patch archive is unzipped and the PATCH_TOP directory created, you run the copy driver (c<patchnum>.drv) with AutoPatch just like a standard patch. You may have multiple j<patchnum>.zip files included in your patch, each of which is referenced in a jcopy command in the copy driver.

Creating a Digital Certificate

Before a Java patch can be applied, a digital certificate must exist. The digital certificate is usually created as a post-upgrade step or during the post-installation phase of an Applications install. If it does not exist, you must create one before Java files can be patched.

The AD Java Key Generation utility (adjkey) located in the bin directory under AD_TOP can be used to create and set up the signing entity, the digital key pair, the digital certificates, and the adsign.txt file required by AutoPatch and other AD Utilities to sign JAR files. Adjkey is a utility built to expand the functionality of javakey. It has one command line argument, - *initialize*.

To run adjkey to create a digital certificate:

- Go to the master web server and set up your environment pointing to the web server's APPL_TOP.
- Enter the following command:

  For UNIX users:

  ```
  $ adjkey -initialize
  ```

  For NT users:

  ```
  C:\> adjkey -initialize
  ```

- You are prompted for an entity name and an organization name.

Adjkey takes the following actions:

- Creates a certificate (.cer file) in the admin directory under your APPL_TOP
- Imports this certificate into the identitydb.obj file (in the user's home directory, default is applmgr).
- Creates adsign.txt in the admin directory of your APPL_TOP. This file is used to pass arguments to JRI for signing JAR files.

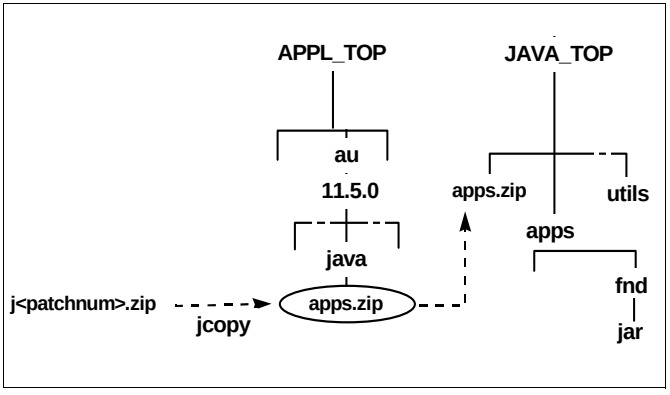
Once this process is complete, copy identitydb.obj and adsign.txt to *all* web servers.

> **Note:** There must only be one signature, which is copied to all web servers in the environment.

> **Additional Information:** Finishing Your Installation, *Installing Oracle Applications*

Patching Java Files

The copy driver file for a Java patch contains one or more jcopy commands. The jcopy command takes the contents of the Java patch file, j<patchnum>.zip, and merges them with the apps.zip file located in the java directory under AU_TOP. This process replaces the old class files in the existing apps.zip and updates it with the new files in j<patchnum>.zip. It then updates the public apps.zip file located under JAVA_TOP.



apps.zip

Apps.zip is a patchable archive of all Java class files required by Oracle Applications. Individual Java class files are never present in the file system. Apps.zip is located in the java directory under AU_TOP and a public copy is stored under JAVA_TOP. The public copy is the one used in a web server environment, as the web server can see the apps.zip in JAVA_TOP but, for security reasons, cannot access the apps.zip in the APPL_TOP.

Regenerate JAR files

The generation portion of the Java patch process regenerates the JAR (Java ARchive) files in both the APPL_TOP and JAVA_TOP. The JAR files in the APPL_TOP are located in $<PROD>_TOP/java/jar (%<PROD>_TOP%\java\jar for NT) and the JAR files in the JAVA_TOP are located in JAVA_TOP/oracle/apps/<prod>/jar (JAVA_TOP\oracle\apps\<prod>\jar for NT). The following table lists some FND JAR files.

JAR File	Description
fndewt.jar	Packages all Java ewt UI classes needed for web forms and for hierarchy editors
fndform.jar	Packages the Java code for the forms client applet
fndlist.jar	Lists all JAR files in Oracle Applications and for each JAR file lists the class files contained in each. This file is used by JInitiator for dynamic JAR file downloads
fndaol.jar	Contains all AOL add-ons to the web forms UI (for example, flex, message dialogs, process navigator)
fndctx.jar	Provides AOL services to Hierarchy Editors
fndtcf.jar	Provides communication between client and middle-tier Java code for Hierarchy Editors
fndhier.jar	Contains implementation of the Hierarchy Editor applet

JAR files can be regenerated as a maintenance task anytime with the regenerate product JAR file option of AD Administration.

Additional Information: File System Tasks in Chapter 2

Signing JAR files

The final step of the Java patching process is to sign all JAR files with the customer's digital signature. This is an inherent part of the Java patching process,

and no user intervention is required. AutoPatch maintains JAR files on all servers, but only signs them on the web server. Signing all Oracle Applications JAR files can take considerably longer than generating the JAR files without signing. Fortunately, most Oracle Applications patches only affect a subset of all JAR files and AutoPatch only recreates and signs the JAR files affected by a given patch.

Java Release Infrastructure

The Java Release Infrastructure (JRI) is a framework used by Oracle Applications to develop, release, patch, and maintain Oracle Applications Java code. Much of the functionality of JRI is invisible. The only direct interaction you may have is during the Java patching process when AutoPatch uses the jcopy command to merge the Java archive patch file with the apps.zip file and during the maintaining process when AD Administration is used to regenerate JAR files.

5

Other AD Utilities

This chapter contains information about the other AD Utilities that AutoUpgrade, AD Administration, and AutoPatch rely on to manage installation, patching, upgrade, and maintenance tasks. It includes information about the following:

- AD Controller (adctrl)
- AD Configuration (adutconf.sql)
- AD File Identification (adident)
- AD Splicer (adsplice)
- File Character Set Conversion (adncnv)
- ODF Comparison (adodfcmp)
- AD Relink (adrelink.sh)
- AD Merge Patch (admrgpch)
- DataMerge (addmimp)
- AD Run SQL (adurs)
- AD Rebase (adrebase.exe)
- License Manager (LicenseMgr)

Review the Running AD Utilities section in Chapter 1 before running any of the utilities in this chapter.

> **Note:** To use these utilities on Windows NT, you will need the MKS Toolkit version 6.1a or higher (http://www.mks.com)

AD Controller (adctrl)

By using AD Controller, you can determine the status of AutoUpgrade, AD Administration, or AutoPatch workers and restart failed tasks. Before you start the utility, perform these preparation steps:

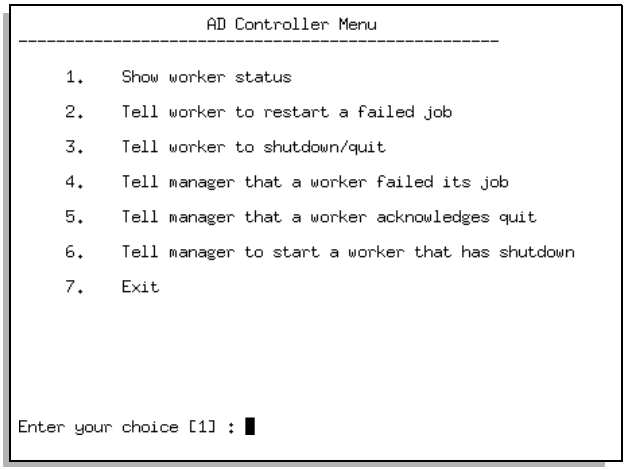
> **Note:** AD Controller should be run in its own window, not in the same window that AutoUpgrade, AD Administration, or AutoPatch is running in.

1. Log in as applmgr.
2. Run the product group's environment file for UNIX. Set APPL_CONFIG to the name of the product group registry subkey for NT.
3. Start AD Controller with this command by typing *adctrl* at the prompt.

You will be prompted to 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/<db_name>/log, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. For NT, the file is located in %APPL_TOP%\admin\<db_name>\log, where <db_name> is the value of ORACLE_SID or LOCAL.
- Supply the Oracle Application Object Library username and password of the product group.

AD Controller displays the main menu.



Type a 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 AutoUpgrade in Chapter 3

Log File

If an error occurred or if you are unsure of messages returned by AD Controller, review the adctrl.log file. 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. If you delete the file, AD Controller creates a new one 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 R115          afasdfa.dat  Running
     2   Run       Installing at R115          aftxt.drv    Failed
     3   Run       Installing at R115          afatsaf2.sql Running
     4   Run       Installing at R115                       Wait
     5   Run       Installing at R115                       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.

Status	Meaning
Assigned	The manager assigned a job to the worker, and the worker has not started.
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.

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 from the main menu 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:** Changes to the environment variables made while AutoUpgrade is running do not take effect until AutoUpgrade 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:

1. Verify from the operating system that the worker processes has exited (no longer exists).
2. Choose Option 1 to show worker status. The Status column will show the worker as Running or Restarted instead of Failed.
3. Choose Option 4 to indicate that the worker has failed.
4. Choose Option 6 to restart the worker. When prompted, enter the number of the worker that failed.

> **Note:** Do not restart a worker process that has not failed. Doing so will create duplicate worker processes with the same worker ID and may cause incorrect results.

Reactivating the Manager

A restarted worker resumes the failed job immediately as long as the manager is active. The other workers 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 screen 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:
Restarting all failed workers... (restart # 1)
```

When the manager becomes idle, AD Controller will automatically attempt to restart the failed workers. If the workers fail again on the same jobs, you will see a message like the following:

```
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.
```

You will need to determine the cause of the error and fix it before manually restarting the workers.

To manually restart a worker after the manager has stopped, set the worker to restart (option 2 from the AD Controller main menu). AutoUpgrade automatically reactivates, then restarts the worker processes, displaying a message like this:

```
Restarted: file aftwf01.sql  on worker  3 for product fnd username APPLSYS.
```

Shutting Down the Managers

Perform the following steps if you must stop AutoUpgrade, AD Administration, or AutoPatch 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. The following command is an example — note that the command arguments may vary on different platforms.

For UNIX users:

```
$ ps -a | grep adworker
```

3. When all workers have shut down, the utility will exit.

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 cannot, however, add additional workers, if you find that your operation might be more beneficial with additional parallel processing. 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 Configuration (adutconf.sql)

This utility is a SQL script 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 MultiOrg is installed
- whether Multiple Reporting Currency (MRC) is installed
- information about all installed products, including shared and dependent products
- information on all registered schemas
- the base language and other installed languages
- NLS environment variables
- rollback segment information
- SQL*Plus PAUSE and NEWPAGE settings
- useful information on referential integrity issues

Use the following command to run this script:

For UNIX users:

```
$ cd $AD_TOP/sql
$ sqlplus <APPS schema username>/<APPS schema password> @adutconf.sql
```

For NT users:

```
C:\> cd %AD_TOP%\sql
C:\> sqlplus <APPS schema username>/<APPS schema password> @adutconf.sql
```

The output file adutconf.lst will be placed in your current working directory. You may need this information for debugging or to document the status of your installation when you contact Oracle Support Services.

AD File Identification (adident)

With AD File Identification, you can 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. This utility is also useful for collecting information about your site when contacting Oracle Support Services.

Use the following command to run the program:

For UNIX users:

```
$ adident Header <file 1> [ <file 2> <file 3> ... ]
```

For NT users:

```
C:\> 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 for UNIX and .obj for NT), library file (extension .a for UNIX and .lib for NT), dynamic link library (.dll for NT), or executable program (.exe for NT). You may provide any number of file names as arguments. When you 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:

For UNIX users:

```
$ adident Header $FND_TOP/lib/wfload.o $FND_TOP/lib/libfnd.a

wfload.o:
  wfload.oc           115.5.1100.3
libfnd.a:
  fdacon.lc           115.0
  fdatat.lc           115.0
  fdastr.lc           115.0
```

```
  fdaupd.lc           115.0.1100.2
  fdahmi.lc           115.0.1100.2
  fdacv.lc            115.0
  fdfutl.lc           115.4
  ....
```

For NT users:

```
C:\> adident Header %FND_TOP%\lib\wfload.obj %FND_TOP%\lib\fndst.lib

wfload.obj:
  wfload.oc           115.5.1100.3
fndst.lib:
  fdacon.lc           115.0
  fdatat.lc           115.0
  fdastr.lc           115.0
  fdaupd.lc           115.0.1100.2
  fdahmi.lc           115.0.1100.2
  fdacv.lc            115.0
  fdfutl.lc           115.4
  ....
```

You can also use a '*' to identify all files in a directory (e.g. *.sql to identify all SQL scripts).

AD Splicer (adsplice)

Products introduced after a given release (not on the base Oracle Applications CD for that release) can be difficult to install or maintain because the AD Utilities do not recognize them as valid Oracle Applications products, therefore they may be ignored or fail. AD Splicer resolves this difficulty by modifying your APPL_TOP and database so that AutoPatch and AD Administration recognize the off-cycle product(s) as being a valid Oracle Applications product(s) for the given release.

Note that only the AutoPatch and AD Administration utilities recognize products added by AD Splicer. AutoUpgrade deliberately ignores products for the existing release that have been added by AD Splicer.

> **Attention:** Do not use AD Splicer to add custom products into your APPL_TOP.

Splicing New Products

Before using AD Splicer:

- Download the product's initial patch (includes AD Splicer files)
- Follow instructions in the patch readme file

The instructions in the readme file typically read as follows:

1. Apply prereq patches (if any).
2. Go to the admin subdirectory of the patch directory.
3. Edit newprods.txt (see Editing newprods.txt).
4. Manually copy AD Splicer control files (*.txt) to admin subdirectory under APPL_TOP.
5. Go to the admin subdirectory under APPL_TOP.
6. Run AD Splicer by typing `adsplice` on the command line.

   Use newprods.txt as AD Splicer control file. Make sure to create a new environment file (or update the registry).
7. Run adutconf.sql to verify that the product was spliced into the database.
8. **For Non-NT platforms only**: Integrate environment file created by AD Splicer with your existing environment file. If your existing environment file was not customized, you can just copy the new version on top of the existing one.
9. Log out and log back in so that you are using the new environment file (or registry entries) to set up your environment.
10. Verify that <PROD>_TOP registry and environment variables are set for your newly spliced products.
11. Run AutoPatch to install files and database objects for your new products.

AD Splicer must be run for each APPL_TOP and database combination in which you want the AD Utilities to recognize the spliced products as valid. It updates your database the first time you run it for each given APPL_TOP and database combination, regardless of which servers have been implemented in your APPL_TOP.

If you are applying the same patch with AD Splicer to multiple APPL_TOPs corresponding to multiple databases, we recommend you use *exactly* the same settings in newprods.txt for each APPL_TOP and database. This implies that you will use the same Oracle schemas, sizing factors, and tablespaces. It also implies that your spliced products will be located under APPL_TOP. If this is not possible, you must edit newprods.txt separately for each APPL_TOP and database

combination. If possible, use *exactly* the same set of AD Splicer control files for each APPL_TOP and database combination.

Editing newprod.txt

AD Splicer has two kinds of control files:

Product Definition Files

There are two Product Definition Files per spliced product and they must not be edited. The files are

- <prod>prod.txt: Language-independent info for product <prod>
- <prod>terr.txt: Language-dependent info for product <prod>

Product Configuration File

There is one Product Configuration File for each group of related spliced products. This file is called newprods.txt by default and must be edited by the customer prior to copying AD Splicer control files to any APPL_TOP. Each spliced product in newprods.txt has an entry like the following:

```
product=bis
base_product_top=*APPL_TOP*
oracle_schema=bis
sizing_factor=100
main_tspace=*Product_Name*D
index_tspace=*Product_Name*X
temp_tspace=*Temporary_Tablespace*
default_tspace=*Product_Name*D
```

For each spliced product, the newprods.txt file must contain all of the entries shown in the example. These entries must be listed in the exact order shown above. No entry may be omitted. Here is the basic information about each line of a product entry in newprods.txt.

Entry	Description
product=	Identifies the product being spliced. Cannot be modified. The product abbreviation <prod> found here is also used to name the <prod>prod.txt and <prod>terr.txt control files for this product. Most internal references to this product use <prod>.

Entry	Description
base_product_top=	Identifies the base directory under which the product's files will be located. The default value of *APPL_TOP* will put the product's files under the APPL_TOP directory. If you want the files for this product to be located somewhere other than APPL_TOP, you must specify the full pathname of the directory here.
oracle_schema=	Identifies the Oracle Schema in which database objects for the product will be created. The default oracle schema is named the same as the product abbreviation. You can change this if you prefer to put the product's database objects in a different schema. Note that it is complicated to move a product's objects from one schema to another, as this involves import/export as well as updates to internal Oracle Applications tables. Because of this, you should choose your initial schema carefully.
sizing_factor=	Identifies the Sizing Factor Oracle Applications will use when creating tables and indexes for this product. The default value of 100 means 100%. In other words, the product's tables and indexes will be created with the default sizes determined by the product's development team. Increasing to 200 will create the product's tables and indexes at 200% of the default settings, which is twice as large as the default settings. You can increase the sizing factor, but we do not recommend that you decrease it.
main_tspace=	Specifies the tablespace in which this product's tables will be created. To correctly follow the OFA standards, you should create a new tablespace called <PROD>D (where PROD is the product abbreviation listed in the product= line above) to hold the tables for this new product prior to running AD Splicer, then set this value to <PROD>D. If you run AD Splicer with the default value (`*Product_Name*D`), it will fail.
index_tspace=	Specifies the tablespace in which this product's indexes will be created. To correctly follow the OFA standards, you should create a new tablespace called <PROD>X (where PROD is the product abbreviation listed in the product= line above) to hold the indexes for this new product prior to running AD Splicer, then set this value to <PROD>X. If you run AD Splicer with the default value (`*Product_Name*X`), it will fail.
temp_tspace=	Specifies the tablespace that will be used by this product for creating temporary segments. In general, each Oracle Applications database should have a separate tablespace dedicated to temporary segments, and all Oracle Applications schemas (including the APPS schema) should use this tablespace for temporary segments. We recommend that you set this value to the temporary tablespace for the Oracle Applications database in which you are installing this product. If you run AD Splicer with the default value (`*Temporary_Tablespace*`), it will fail.
default_tspace=	Specifies the default tablespace in which this product's objects will be created. The default tablespace is used if scripts create tables or indexes for this product without explicitly specifying a tablespace. We recommend that you set this to the same value you used for the main_tspace= line. If you run AD Splicer with the default value (`*Product_Name*D`), it will fail.

File Character Set Conversion (adncnv)

The File Character Set Conversion is used to convert files from one character set to another. This may be required 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. In general, you do not need to run this utility manually because AutoPatch and the Rapid Install do all required character set conversion for you automatically.

Syntax

You can convert one file at a time with this command:

For UNIX users:

```
$ adncnv <source file> <source char set> <destination file> <dest char set>
```

For NT users:

```
C:\> 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 file name of the file to convert
source char set	Current character set
destination file	Path and file name for converted file
dest char set	New character set for converted file

The path and file name for the source and the destination files can be the same if the source file's directory and the APPLTMP directory are on the same file system. In general, it is simpler and safer to use different source and destination file names.

> **Suggestion:** If you cannot convert to the same file name, convert to a different file name or change APPLTMP to a directory on the same file system as the source file directory.

For example, to convert the file afcmstat.sql from the character set we8dec to the character set we8hp, you would type:

For UNIX users:

```
$ adncnv $FND_TOP/sql/afcmstat.sql we8dec $FND_TOP/sql/afcmstat.sql we8hp
```

For NT users:

```
C:\> 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.

ODF Comparison (adodfcmp)

ODF Comparison is used to compare the data model of a customer's database with the standard data model from the current release of Oracle Applications and optionally modify the customer's database objects to match the standard data model.

> **Attention:** You may not use ODF Comparison for custom development.

Each Oracle Applications product is composed of functional 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.

ODF Comparison compares a building block to its description file. A log file records any missing, extra, or incorrectly defined objects in the database. For example:

```
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 adodfcmp create missing objects, grants, and synonyms automatically. If you do, you will see this line after the log file record:

```
Statement Executed.
```

Running ODF Comparison

To run ODF Comparison, move to the directory that contains the appropriate object description file:

For UNIX users:
```
$ cd $<PROD>_TOP/admin/odf
```

For NT users:
```
C:\> cd %<PROD>_TOP%\admin\odf
```

For Example, to run the utility on a building block for Oracle General Ledger, move to this directory:

For UNIX users:
```
$ cd $GL_TOP/admin/odf
```

For NT users:
```
C:\> 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, they will be located in the $<PROD>_TOP/patch/115/odf directory for UNIX. Or, in the %<PROD>_TOP%\patch\115\odf directory for NT.

Syntax

Run adodfcmp by entering the following command:

For UNIX users:
```
$ adodfcmp <parameter>=<value> [<parameter>=<value> . . .]
```

For NT users:
```
C:\> adodfcmp <parameter>=<value> [<parameter>=<value> . . .]
```

You can see instructions about ODF Comparison syntax by typing adodfcmp at the prompt.

Parameters

The following entries explain ODF Comparison 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*i*. For example, use *changedb=yes* and *mode=tables* to create tables, indexes, and grants.

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
touser (required)	(no default)	Specifies the ORACLE username and password of the Oracle Applications product to grant to. In Release 11*i*, this is usually the list of APPS schemas. 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 Architecture installation.
priv_schema (Required)	(no default)	Specify a schema that has DBA privileges, along with its password. You may specify the APPS schema or some other schema with DBA privileges, such as SYSTEM.
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.

Parameter Name	Default Value	Description
userid (Required)	(no default)	Supply the ORACLE username and password for the Oracle Applications product's base schema. This is the schema where the product's tables, indexes, and sequences are located. For example, use userid=GL/GL when comparing a building block of Oracle General Ledger.
changedb (Optional)	No	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 do not exist in the ODF file. The default option, *changedb=No*, will not change database objects to match the definitions in their object description files.
alternext (Optional)	No	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)	Use to provide 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	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	Sets the sizing factor at which the utility creates missing database objects. The utility will create missing database objects if *changedb=Yes*. **Additional Information**: Sizing Factor, *Oracle Applications Concepts*
tspace (Optional)	SYSTEM	Sets the tablespace for tables created by the utility.
indextspace (Optional)	SYSTEM	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	Determines the number of rows to update at a time when populating new columns.

Parameter Name	Default Value	Description
oldviews (Optional)	replace	Use to rename, drop, or replace views that do not match their descriptions. Set *oldviews=rename* to change the names of non-matching views to R115_<*view name*> and create new views from the description file. 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 re-creating the views. **Additional Information**: CREATE VIEW command in *Oracle8i Server SQL Reference Manual.*

Examples

Note the following examples:

Compare Building Block and Create Missing Objects Compares and creates database objects.

For UNIX users:

```
$ adodfcmp odffile=<filename> userid=<username>/<password> 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>,...] changedb=Yes
```

For example:

```
$ adodfcmp odffile=glje.odf userid=GL/GL space=GLD indextspace=GLX \
  priv_schema=APPS/APPS mode=baseonly touser=APPS/APPS changedb=Yes
```

For NT users:

```
C:\> adodfcmp odffile=<filename> userid=<username>/<password> \
     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>,...] changedb=Yes
```

For example:

```
C:\> adodfcmp odffile=glje.odf userid=GL/GL space=GLD indextspace=GLX \
     priv_schema=APPS/APPS mode=baseonly touser=APPS/APPS changedb=Yes
```

Create Grants and Synonyms Creates missing grants and synonyms:

For UNIX users:

```
$ adodfcmp odffile=<filename> userid=<username>/<password> 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>,...] changedb=Yes
```

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 mode=grants \
  touser=APPS/APPS,APPS2/APPS priv_schema=APPS/APPS changedb=Yes
```

For NT users:

```
C:\> adodfcmp odffile=<filename> userid=<username>/<password> 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>,...] changedb=Yes
```

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.

```
C:\> adodfcmp odffile=glje.odf userid=GL/GL mode=grants \
    touser=APPS/APPS,APPS2/APPS priv_schema=APPS/APPS changedb=Yes
```

AD Relink (adrelink.sh)

AD Relink is used to relink Oracle Applications executable programs with the Oracle8 Server product libraries. Executable programs can also be relinked through the Relink Applications Programs task from the Maintain Applications File menu of AD Administration.

Attention: In Release 11*i*, you should use AD Administration to relink executables for most products. You must use adrelink.sh to relink AD executables because AD Administration does not relink AD executables.

Additional Information: Chapter 2 in this manual

If an error occurred during relinking, or if you are not sure that the relinking was successful, review the file adrelink.log. If adrelink.sh was run from AutoUpgrade,

AD Administration, or AutoPatch, the file is in $APPL_TOP/admin/<db_name>/log, where <db_name> is the value of your ORACLE_SID or TWO_TASK variable. NT users will find it in %APPL_TOP%\admin\<db_name>\log, where <db_name> is the value of ORACLE_SID or LOCAL. If you ran adrelink.sh from the command line, the file is in $APPL_TOP/admin/log for UNIX and %APPL_TOP%\admin\log for NT. As it runs, adrelink.sh 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. A new adrelink.log file is created the next time adrelink.sh runs.

Running adrelink.sh

Perform the following steps before you use adrelink.sh:

1. Verify that you have C development tools that are compatible with Berkeley Version 1.15 or higher for UNIX. Note that a C compiler is not needed unless you are doing custom development in C. See the Preface of this manual for the software requirements for NT.
2. Log in as applmgr and run the appropriate environment file. NT users should set APPL_CONFIG to the name of the product group registry subkey.
3. 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.

For NT users:

4. Generate environment variable files (apps.sh and apps.cmd) in %APPL_TOP% by running adregenv.exe in a MS-DOS command prompt window.

   ```
   C:\> adregenv.exe <APPL_CONFIG>
   ```

   For example:

   ```
   C:\> adregenv.exe p1152
   ```

5. Open Korn shell window (Start > Program > MKS Toolkit > Korn Shell) and change directory to %APPL_TOP%. Run apps.sh to set up all required environment variables. (Note that there is a space between the dots in the following command.)

   ```
   C:\> . ./apps.sh
   ```

6. Run the command:

```
C:\> export PLATFORM=WIN_NT
```

7. Change directory to %AD_TOP%/bin and relink the desired file using the following syntax:

```
C:\> sh adrelink.sh force=y{y | n} [<optional args>] <targets>
```

-or-

```
C:\> sh adrelink.sh force=y{y | n} [<optional args>]\
filelist=<file>
```

where <targets>={<product module> | [<product module>]. . .}

After relinking, restart the concurrent managers and allow Oracle Applications users to log on.

Additional Information: Concurrent Managers, *Oracle Applications System Administrator's Guide*

Syntax

To relink one or a few executable programs, run adrelink.sh with one of these two commands:

For UNIX users:

```
$ adrelink.sh force={y | n} [<optional args>] <targets>
```

or

```
$ adrelink.sh force={y | n} [<optional args>] filelist=<file>
```

For NT users:

```
C:\> adrelink force={y | n} [<optional args>] <targets>
```

or

```
C:\> 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: *none* (do not back up any executables) *all* (back up all executables) *file* (back up files in adlinkbk.txt) The default is *file*.

At the prompt, Type *adrelink.sh* to see online instructions about the syntax. Type *adrelink.sh* `examples` to see examples.

Parameters

The following entries describe adrelink.sh parameters. Because these parameters are required, there are no default values.

force *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.sh does not back up or remove executables.

product module Identifies the executable programs to relink. Enclose a product abbreviation and program name in quotation marks, as in **"**ad adadmin".

Examples

The following table contains examples of the way you can use adrelink.sh:

To...	Do this...

Relink a Single Executable Program	This command relinks a specific executable program:
	`$ adrelink force=y "<product> <module name>"`
	or
	`C:\> adrelink force=y "<product> <module name>"`
	To relink the Applications DBA program *adaimgr*:
	`$ adrelink force=y "ad adaimgr"`
	or
	`C:\> adrelink force=y "ad adaimgr"`

Relink Selected Executable Programs for Multiple Products	This command relinks the Applications DBA modules *adctrl* and *adworker*:
	`$ adrelink force=y "ad adctrl" "ad adworker"`
	or
	`C:\> adrelink force=y "ad adctrl" "ad adworker"`

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 information available. Usually, this is not very useful, since the debugging information 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.sh, specify the argument *link_debug=y* on the adrelink.sh command line after the force argument and before any other arguments. The following example shows the command to relink the Applications DBA module *adpatch* with debug:

For UNIX users:

```
$ adrelink force=y link_debug=y "ad adpatch"
```

For NT users:

```
C:\> adrelink force=y link_debug=y "ad adpatch"
```

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.sh 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.

For example, you might create a file that lists several executable programs for different products:

For UNIX users:

```
gl      bin     GLPPOS
gl      bin     GLCCON
fnd     bin     f60webmx
fnd     bin     FNDLIBR
```

For NT users:

```
<GL_TOP>/bin/GLPPOS.exe
<GL_TOP>/bin/GLCCON.exe
<FND_TOP>/bin/f60webmx.exe
<FND_TOP>/bin/FNDLIBR.exe
```

> **Note:** Although it is possible for you to relink several executables for several different products using an adrelink.sh target file, we recommend that you use AD Administration instead.

Run adrelink.sh

Run adrelink.sh with the following command:

For UNIX users:

```
$ adrelink.sh force=<y|n> filelist=<filename>
```

Here is an example:

```
$ adrelink.sh force=y filelist=/home/applmgr/exec.lst
```

For NT users:

```
C:\> adrelink force=<y|n> filelist=<filename>
```

Here is an example:

```
C:\> adrelink force=y filelist=/home/applmgr/exec.lst
```

Backup Options for adrelink.sh

The adrelink.sh 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).

force=n relinks out-of-date executables and does not back them up or remove them prior to relinking.

force=y and **backup_mode=none** removes existing executables prior to relinking.

force=y and **backup_mode=all** renames each existing executable prior to relinking.

force=y and **backup_mode=file** (the default) acts based on the instructions in $APPL_TOP/admin/adlinkbk.txt. As shipped, adlinkbk.txt tells adrelink.sh 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.sh 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.sh backs up.

Because AD programs call adrelink.sh 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.sh backs up an executable prior to relinking it, it renames the existing executable. If there is already a backup copy of the executable, adrelink.sh first renames the backup copy, then renames the executable. For example, if you are relinking FNDLIBR, adrelink.sh will rename it to FNDLIBR.sav prior to relinking. If both FNDLIBR and FNDLIBR.sav exist, adrelink.sh first renames FNDLIBR.sav to FNDLIBR.<pid> (where <pid> is the process ID of the current adrelink.sh 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.sh 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.

AD Merge Patch (admrgpch)

AD Merge Patch is a utility that is designed to merge multiple AutoPatch compatible patches into a single integrated patch. It is an executable located in the bin directory of AD_TOP. To merge two or more patches into a single integrated patch, run admrgpch with the following arguments:

For UNIX users:

```
$ admrgpch <source directory> <destination directory>
```

For NT users:

```
C:\> admrgpch <source directory> <destination directory>
```

The <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 of Chapter 4. The <destination directory> is the directory in which admrgpch will create the merged patch.

This utility reads the c<patchnum>.drv, readme.txt, d<patchnum>.drv, and g<patchnum>.drv for each patch in the source directory and merges them to create cmerged.drv, readme.txt, dbmerged.drv, and gmerged.drv files in the destination directory. It also merges the set of files contained in the individual 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 to the destination directory.

Actions in the merged patches are grouped by product then by patch number. Comments in the merged readme.txt are also ordered by product then by patch number.

After admrgpch runs, you should check the admrgpch.log file for errors. The file is located in your APPL_TOP in the log directory under admin. If you do not find any

errors, look in the readme.txt file in the destination directory for instructions on applying the merged patch using AutoPatch.

> **Note:** AD Merge Patch will not merge patches of different releases, different platforms, or different parallel modes.

DataMerge (addmimp)

DataMerge is similar to the import utility. It allows you to import data into the Applications database. Sometimes Oracle Support Services will ask you to run DataMerge to work around problems encountered while upgrading your Oracle Applications database. However, in general, you should not run DataMerge manually unless instructed to do so by Oracle Support Services.

If you are running DataMerge to work around or reproduce an upgrade problem, go to the import directory under admin in the <PROD>_TOP of the product that owns the DataMerge files that failed. Run the same database command that you found in your AD worker log file. You may wish to specify different file names for the DataMerge log and summary file as follows:

For UNIX users:

```
$ cd $<PROD>_TOP/admin/import
```

For NT users:

```
C:\> cd %<PROD>_TOP%\admin\import
```

Where <PROD> is the product short name of the product that owns the DataMerge file you want to run.

For UNIX users:

```
$ addmimp <parameter>=<value> [<parameter>=<value> . . .]
```

For NT users:

```
C:\> addmimp <parameter>=<value> [<parameter>=<value> . . .]
```

You can see instructions about DataMerge syntax by typing addmimp at the prompt.

Parameters

You can specify parameters in any order on the command line. You must specify values for target, export, control, parameter, log, and summary. The default value applies if you do not specify an optional parameter.

Parameter Name	Default Value	Description
target (required)	(no default)	Target account ORACLE username/password
export (Required)	(no default)	AD export file name (.exp)
control (Required)	(no default)	Control file name (.ctl)
parameter (Required)	(no default)	Parameter file name (.dat)
log (Required)	(no default)	Log file name (.log)
summary (Required)	(no default)	Summary report file name (.sum)
starttable (Optional)	(no default)	First table to process
changedb (Optional)	Yes	Update Your Database Or Not
trace (Optional)	No	Set SQL trace on
numofset (Optional)	1000000	Offset value for NUMBER columns
upkchar (Optional)	#	Offset character for CHARACTER columns
srcbufsize (Optional)	524288	Size of Source Fetch Buffer in bytes
tarbufsize (Optional)	524288	Size of Target Fetch Buffer in bytes
misbufsize (Optional)	524288	Size of Mismatched Rows Buffer
newbufsize (Optional)	524288	Size of New Rows Buffer in bytes
oldbufsize (Optional)	524288	Size of Out of Date Rows Buffer
obsbufsize (Optional)	524288	Size of Obsolete Rows Buffer in bytes
actbufsize (Optional)	524288	Size of Action Buffer in bytes
lngbufsize (Optional)	524288	Size of Long Buffer in bytes

DataMerge reads three data files. It expects that any specified file is in your current directory, unless you specify a full path name for the file.

export (.exp)	The data (or export) file is a binary file that contains data exported from a control database at Oracle.
control (.ctl)	The metadata (or control) file is an odf file that contains descriptions of table structure, indexes, primary keys, and so on.

parameter (.dat)	The parameter file is a text file that contains the rules that tell DataMerge what to do for each table.

DataMerge creates two log files, a detailed log file and a summary log file. Provide names for these log files with the log and summary parameters. To determine the effect of a DataMerge session without actually changing the data in the database, set the changedb parameter to No. Check the log and summary files to determine the changes DataMerge would have made had the changedb parameter been set to Yes.

Example

If you want to run DataMerge for the PO (Purchasing) Lookups building block against the ORACLE account PO/PO:

For UNIX users:

```
$ cd $PO_TOP/admin/import
$ addmimp target=APPS/APPS log=polk.log summary=polk.sum \
  export=polk.exp control=polk.ctl parameter=polk.dat
```

For NT users:

```
C:\> cd %PO_TOP%\admin\import
C:\> addmimp target=APPS/APPS log=polk.log summary=polk.sum \
  export=polk.exp control=polk.ctl parameter=polk.dat
```

AD Run SQL (adurs)

AD Run SQL is a utility that allows you to run a specified file in AutoUpgrade (AutoPatch) SQL mode. SQL mode is an interpreter that reads in your SQL script and then executes the statements it understands directly. SQL mode provides automatic error handling that is more robust than that of SQL*Plus.

When AutoUpgrade or AutoPatch run SQL scripts, they use AD SQL mode for most SQL scripts unless:

- the SQL script includes PL/SQL
- the SQL script includes SQL*Plus commands, for example, column X new_value, start or spool.
- the SQL script includes a CREATE TABLE or INDEX statement and you do not want the object to be autosized

If a SQL script being run by AutoUpgrade or AutoPatch in AD SQL mode fails, you can run it manually in AD SQL mode using AD Run SQL (adurs). In general, you

should use AD Run SQL to manually run SQL scripts that were run by AutoUpgrade or AutoPatch in AD SQL mode because these scripts may fail if run from SQL*Plus.

Run AD Run SQL with the following command:

For UNIX users:

```
$ adurs keyword=value [,keyword=value,...]
```

For NT users:

```
C:\> adurs keyword=value [,keyword=value,...]
```

Valid Keywords:

Keywords	Description
userid=	ORACLE username/password to run the SQL file in
sqlfile=	File name of SQL script to run
logfile=	File name of output log file (Default adurs.log)
sizingfactor=	Sizing factor to apply (Default 100)
args=	List of evaluated arguments [if any]

For example:

For UNIX users:

```
$ adurs userid=apps/apps sqlfile=file1.sql
```

For NT users:

```
C:\> adurs userid=apps/apps sqlfile=file1.sql
```

Where the username is apps, the password is apps, and the file name is file1.sql.

AD Rebase (adrebase.exe)

The AD Rebase utility is an NT-only utility that optimizes memory utilization of Applications, RDBMS and Tools executable programs. Applications executables are automatically optimized when delivered by way of Rapid Install, AutoPatch, or adrelink.sh. You may wish to rerun the AD Rebase utility after upgrading the Oracle tools or the RDBMS. To run AD Rebase:

1. Shut down all Oracle processes running on the system, such as Database services, Forms/Metric servers, Concurrent manager services, etc.

2. `C:\> cd %AD_TOP%\bin`
3. Run ADREBASE.exe utility
4. Examine %APPL_TOP%\admin\<APPL_CONFIG>\ADREBASE.log to check for failures.

Failure to rebase a DLL will not cause any functional problems. Warnings that the utility cannot find an expected DLL should be ignored. You will be able to restart and run your Applications regardless of the results of the rebase command.

License Manager (LicenseMgr)

When you want to add additional products or languages to your Oracle Applications installation, you can use the Oracle Applications License Manager to accomplish these tasks. Once you have contacted your Oracle sales representative, or set up your new license agreements online through the OracleStore, you are ready to "turn on" your new products and languages.

To begin, go to:

For UNIX users:

```
$ cd COMMON_TOP/admin/assistants/licmgr
$  LicenseMgr
```

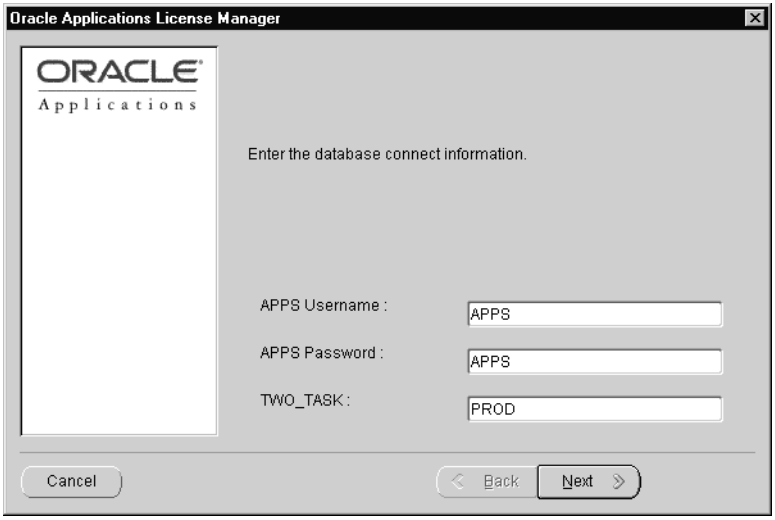
The location of COMMON_TOP was defined during the Rapid Install process.

> **Note:** Ensure that your DISPLAY environment variable is set correctly.

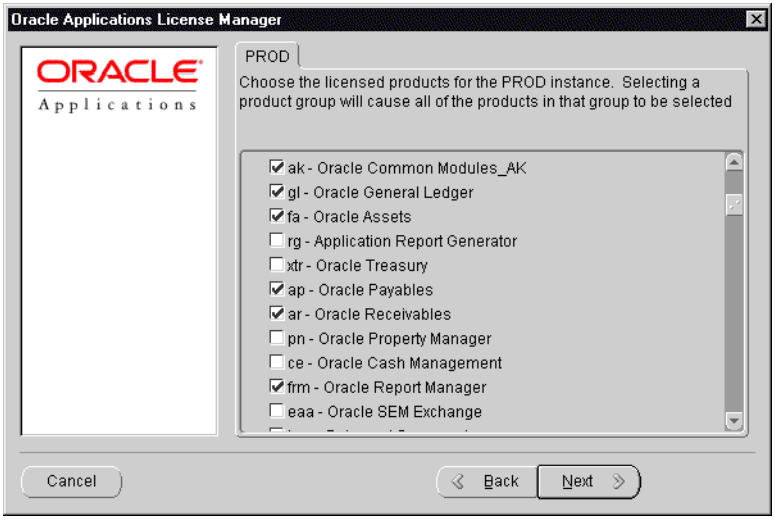
For NT users:

```
C:\> cd COMMON_TOP\admin\assistant\licmgr
C:\> LicenseMgr.cmd
```

License Manager displays the login screen for License Manager.



Enter your APPS username, APPS password, and the value of TWO_TASK (LOCAL for NT users). Click Next.

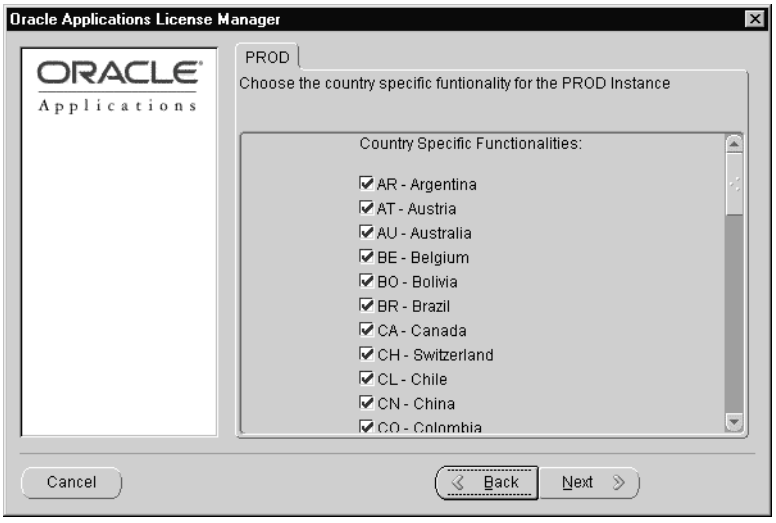


License Manager displays the individual products. As you scroll through the list, you will see that your currently installed products are checked. Click on the additional products you want to add to the list. If you plan to license products by

product families, license the products according to Table 5-1. Notice that products currently licensed in your database are checked and grayed out. You cannot remove products from the list of licensed products using License Manager.

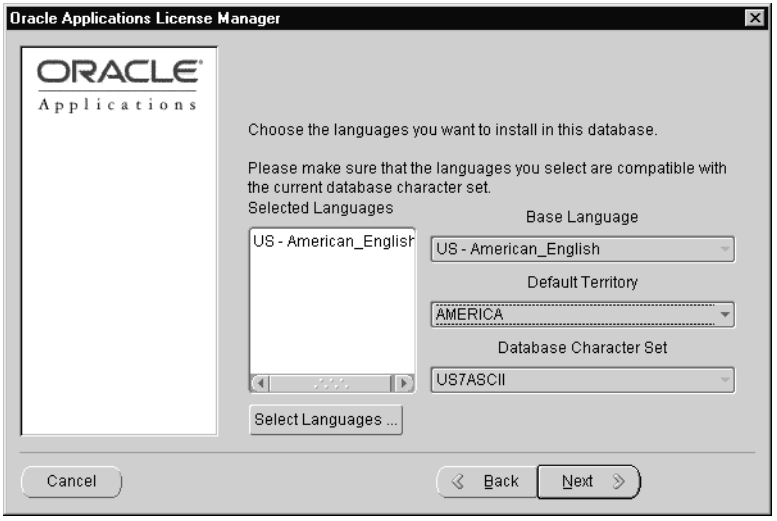
> **Note:** License Manager accommodates both new products and new languages. If you will be licensing only products or only languages, just click Next to omit the screens that you don't need.

When you have finished selecting the additional products, click Next. License Manager displays the Choose the Country Specific Functionalities screen.

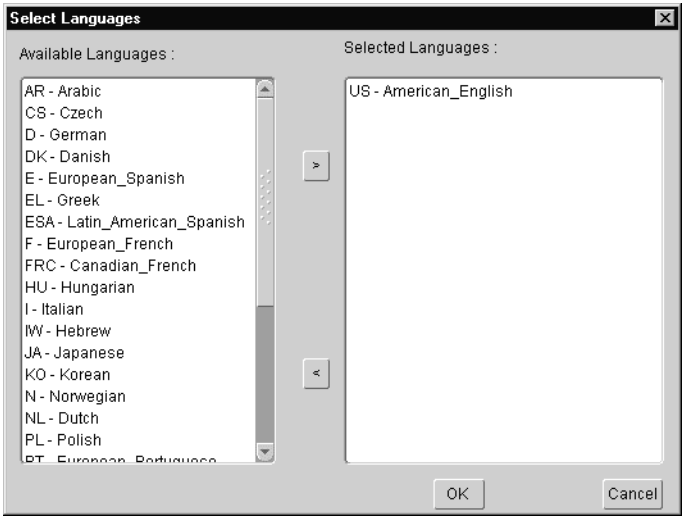


This screen displays the list of country specific functionalities (localizations) that are available in Release 11*i*. Like the product selection screen, License Manager indicates the country specific functionalities that are currently licensed in your database. Scroll through the list and select the new country specific functionalities (you cannot remove country specific functionalities from your database using License Manager). When you are finished, click Next.

The next screen displays the languages that you currently have licensed, along with your base language(s) and the character set of your database. Unlike adding additional products and country specific functionalities, you can change any of the selections displayed on this screen, as long as they are compatible.



To add a new language, use the Select Languages... list box to see a list of available languages.

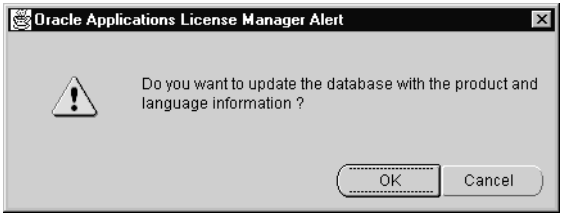


Highlight each new language and click the > to move it to the list of installed languages. When you are finished, click OK.

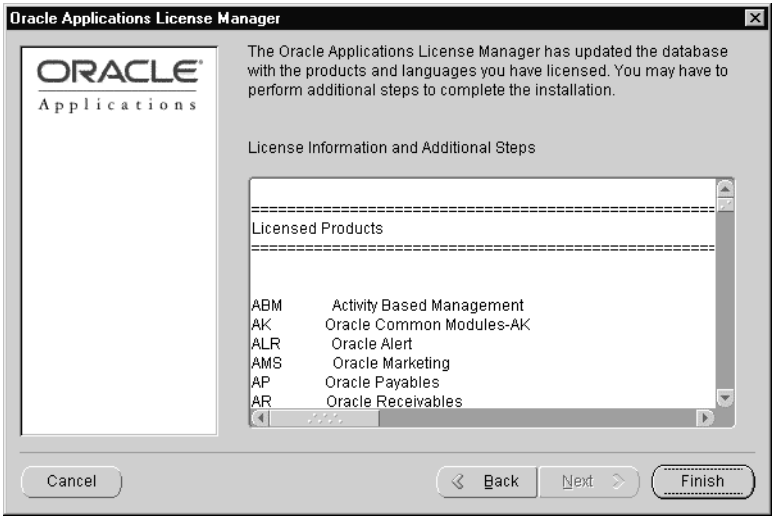
You can change the base language or specify the default end user territory. The character set of the database is displayed for reference. You must make sure that the languages you enable are compatible with the database character set.

You can use License Manager to disable previously licensed languages so that they are ignored. However, License Manager will not remove existing language files.

Once you have completed your selections for additional products and/or languages, click Next.



License Manager displays a screen that lists the licensed products, territories, and languages.



Click Finish. License Manager begins to update your database with the licensing information.

When your new products are licensed, you will need to recreate your Applications environment files on all relevant nodes to update the APPLFULL and APPSHAR variables according to your new installation configuration. Use the Create Applications environment files option of AD Administration to recreate your environment files.

Additional Information: Running AD Administration Interactively in Chapter 4.

Run AutoPatch to add the translated language components or to apply the latest mini-pack(s) for the product(s) you have just licensed.

Additional Information: Running AutoPatch in Chapter 4.

There may be product-specific implementation steps that you will need to perform before using the new product.

Additional Information: *Oracle Applications Implementation Manuals*

Products can be licensed individually or by product families. When licensing by product families, license all products within the product family.

Table 5–1 Licensing Product Families

To License the Product Family	License the Following Products
Basic (required)	0-aol / fnd, 3-au, 50-ad, 60-sht, 160-alr, 190-az, 265-frm, 601-ak
Service	
TeleService	170-cs, 511-csc, 514-css, 690-jtf
Service Online	513-csf, 690-jtf
Depot Repair	512-csd
Spares Management	523-csp
Advanced Scheduler	690-jtf, 698-csr
Contracts	510-okc, 515-oks, 524-okx, 777-oke
iSupport	672-ibu, 690-jtf
Customer Intelligence	191-bis, 518-bic

To License the Product Family	License the Following Products
Marketing	
Marketing Online	520-amv, 530-ams, 690-jtf
Marketing Intelligence	191-bis, 517-bim
Sales	
Sales Online	279-as, 522-asf, 676-bil
TeleSales	521-ast
iStore	670-iba, 671-ibe, 673-iby
iPayment	673-iby, 690-jtf
Sales Intelligence	191-bis, 676-bil
Incentive (Sales) Compensation	283-cn
Order Management	
Order Management	660-ont, 665-wsh
Release Management	662-rlm, 663-vea, 710-rla, 711-veh
Advanced Pricing	661-qp
Configurator	708-cz
Supply Chain Planning	
Advanced Supply Chain Planning	724-msc
Constraint Based Optimization	723-mso
Global ATP Server	724-msc
Demand Planning	722-msd
Supply Chain Intelligence	454-isc
Procurement	
Purchasing	201-po, 202-chv
iProcurement	178-icx, 201-po
Purchasing Intelligence	191-bis, 452-poa
iSupplier Portal	178-icx, 201-po
Manufacturing	
Discrete Manufacturing	250-qa, 401-inv, 700-mfg, 702-bom, 703-eng, 704-mrp, 705-crp, 706-wip, 712-pjm
Flow Manufacturing	714-flm

To License the Product Family	License the Following Products
Manufacturing Scheduling	388-wps
Process Manufacturing	550-gma, 551-gmi, 552-gmd, 553-gme, 554-gmp, 555-gmf, 556-gml, 557-gr, 558-pmi
Manufacturing Intelligence	451-opi, 558-pmi
Shop Floor Management	410-wsm
Warehouse Management	385-wms
Projects	
Project Costing	275-pa
Project Billing	275-pa
Internet Time	275-pa
Financials	
Financials	101-gl, 140-fa, 168-rg, 200-ap, 222-ar, 240-pn, 260-ce, 600-ax, 602-xla, 7000-ja, 7002-je, 7003-jg, 7004-jl
Treasury	185-xtr
Internet Expenses	178-icx, 200-ap
Internet Receivables	178-icx, 222-ar
Financials Intelligence	450-fii
SEM Exchange	270-eaa, 274-fem
Activity Based Management (Activa)	272-evm, 274-fem
Balanced Scorecard	271-bsc, 274-fem
Human Resources	
Human Resources	800-per, 802-ff, 803-dt, 804-ssp
Self-Service HR	178-icx, 800-per
Payroll	801-pay
Time Management	808-hxt, 809-hxc
Advanced Benefits	805-ben
HR Intelligence	453-hri
Training Administration	810-ota
Call Center	
Scripting	519-ies
Advanced Inbound	172-cct, 539-ieo, 690-jtf, 699-ieb

To License the Product Family	License the Following Products
eMail Center	680-iem
Call Center Intelligence	191-bis, 677-bix
e-Hub Software	
Exchange Marketplace	298-pom
Supply Chain Exchange	298-pom
Product Development Exchange	298-pom
Communications/Utility Industry	
CRL Financial Management	531-xnm, 532-xnc, 533-xns, 8721-ipa, 8724-cuf, 8731-cua
Network Logistics (Network Asset Tracking)	8722-cui, 8723-cup, 8727-cus, 8729-cun
Provisioning	535-xdp
Number Portability	534-xnp
Public Sector	
Public Sector Financials	101-gl, 200-ap, 222-ar, 8450-psa
U.S. Federal Financials	101-gl, 200-ap, 222-ar, 8450-psa, 8901-fv
Public Sector HR	800-per, 802-ff, 803-dt, 804-ssp, 8302-pgh
Government HR	800-per, 802-ff, 803-dt, 804-ssp, 8301-ghr
Public Sector Payroll	801-pay, 8303-pqp, 8450-psa
Public Sector Budgeting	8401-psb, 8450-psa
Grants Management	8402-gms, 275-pa, 8450-psa
Grants Proposal	8404-igw, 8450-psa
Labor Distribution	8403-pqp, 8450-psa
Commitment Administration	8407igc, 8450-psa
University	
Student Information	8405-igs, 8406-igf
Other	
EDI Gateway	175-ec
XML Gateway	174-ecx
CRM Gateway for Mobile Devices	689-asg
Contracts	510-okc, 515-oks, 524-okx
Mobile Applications	405-mwa

6

Maintaining the Applications Database

This chapter contains information about maintaining various aspects of the Oracle Applications database. It includes information about the following:

- Analyzing the Database
- Pinning Packages and Sequences
- Changing Oracle Applications Passwords
- Migrating the Oracle Applications Database
- Database Initialization Parameters
- Rollback Segments
- Product Tablespace Requirements

Analyzing the Database

In prior releases, the Oracle Applications initialization parameters set the ORACLE Server to use rule-based optimization (RBO) by default when executing a SQL statement. Oracle Applications Release 11*i* is run in the cost-based optimization mode (CBO). To properly configure the Oracle8i Server to use cost-based optimization, you must analyze the database by gathering statistics about Oracle Applications database objects. You should analyze the database and gather statistics as follows:

- For a new system after it has a significant amount of transaction data stored
- Before an upgrade (as either a Category 1 or Category 3 step)
- During an upgrade (as a Category 4 step)
- As part of system maintenance, approximately once a month

We recommend that you analyze the database by using the Gather Schema Statistics concurrent program. To run this:

1. Log on to Oracle Applications with the System Administrator responsibility
2. Navigate to the Submit Request window (Request > Run)
3. Submit the Gather Schema Statistics program

If schema name is set to ALL, then statistics are gathered for all Oracle Applications schemas (having an entry in the FND_PRODUCT_INSTALLATIONS table). In addition to gathering index and table-level statistics, the procedure also gathers column-level histogram statistics for all columns listed in the FND_HISTOGRAM_COLS table.

> **Additional Information:** Cost-Based Optimization in Oracle Applications, *Oracle Applications System Administrator's Guide; AOL Users Guide*

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 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 packages or sequences are patched using AutoPatch (when the patch readme file specifies you to do so), or any time after objects are invalidated, either because of patching or customizations.

The ADXGNPIN.sql script pins packages and functions in the APPS schemas, while the ADXGNPNS.sql script pins sequences in the base product schemas. Both scripts take the name of a schema from which to pin objects as an argument, or '%' for all schemas. ADXGNPIN.sql generates another SQL file, ADXSPPIN.sql which it automatically invokes. Similarly, ADXGNPNS.sql generates ADXSPPNS.sql which it runs.

1. Create an admin subdirectory under the apps directory in the ORACLE_HOME on your database server (i.e. ORACLE_HOME/apps/admin).
2. Copy ADXGNPIN.sql and ADXGNPNS.sql from the AD_TOP/sql directory to this directory.

3. Set your environment to point to the ORACLE_HOME on your database server.You must access the database server directly. Do not attempt to run any of these scripts by way of Net8.
4. Go to the admin subdirectory under the apps directory in the ORACLE_HOME on your database server that you created in step 1.
5. Run ADXGNPIN.sql and ADXGNPNS.sql from SQL*Plus:

For UNIX users:

```
$ sqlplus SYS/<SYS password> @ADXGNPIN.sql <APPS schema name>

$ sqlplus SYS/<SYS password> @ADXGNPNS.sql <schema name>
```

For NT users:

```
C:\> sqlplus SYS/<SYS password> @ADXGNPIN.sql <APPS schema name>

C:\> sqlplus SYS/<SYS password> @ADXGNPNS.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 can run the pinning scripts automatically at startup by adding the scripts to the database startup script, which is typically dbstart in the bin directory under your ORACLE_HOME.

> **Attention:** Pinning all packages installed in the APPS schema requires 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 can be run at any time and shows the objects known to the SGA and the size that they consume in the SGA. The output file is ADXCKPIN.lst. You run the script with these commands:

For UNIX users:

```
$ cd $APPL_TOP/admin/<db_name>/out
$ sqlplus SYSTEM/<SYSTEM password> @$AD_TOP/sql/ADXCKPIN.sql
```

For NT users:

```
C:\> cd %APPL_TOP%\admin\<db_name>\out
C:\> sqlplus SYSTEM/<SYSTEM password> @%AD_TOP%\sql\ADXCKPIN.sql
```

Changing Oracle Applications Passwords

Changing passwords frequently helps to ensure database security.

Note: You cannot change a username, such as APPLSYS or GL, after you install a product.

You must perform all of the process steps detailed in this section in the order the steps appear.

Warning: Running the following steps out of order may lock you out of Oracle Applications. If this occurs, contact Oracle Support Services.

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 Navigator is displayed if you have only one responsibility. If you have more than one, select System Administrator from the Responsibilities window.

Perform the following steps, listed here with the default menu paths:

1. Navigate to the ORACLE Users form. (Security > ORACLE > Register)
2. Start a query (From the View menu option in the toolbar, Query by Example -> Enter).
3. Enter the Database User Name for which you want to change the password.
4. Run the query. (From the View menu option, Query by Example -> Run)

5. Enter a new password (The change is not displayed). Enter the password a second time to verify it.

6. Commit the changes. (File > Save)

> **Attention:** Passwords for APPLSYS and each APPS schema — including MRC and MLS schemas — must be the same. When you alter the user in SQL*Plus, you must do so for APPLSYS and every APPS schema. If you change the password for one, you must change it for all.

Change Password in SQL*Plus

Log on to SQL*Plus using the SYSTEM DBA account:

For UNIX users:

```
$ sqlplus SYSTEM/<SYSTEM password>
```

For NT users:

```
C:\> 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, you must restart all concurrent managers, 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 or are in the process of upgrading to Release 11*i* 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 why you may want to move your Applications database. For example, you may be:

- Moving the database to a different platform
- Upgrading the version of the ORACLE Server you are using with Applications
- Moving to a split configuration
- Upgrading Oracle Applications, and you require a higher version of the ORACLE Server
- Improving scalability or performance
- Re-organizing database objects into different tablespaces
- Changing your database block size
- Changing your database character set

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 the best way to upgrade the data in your Applications database.

> **Additional Information:** Migration Overview, *Oracle8i Server Migration*

You may need to migrate your database to improve scalability or performance. For example, you may have originally installed the database on the same node as your concurrent processing server, and now want to improve scalability by dividing the two functions across different nodes. 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 export and import the database in order to move or rebuild it.

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 *from*.

Target machine — The machine you are migrating *to*.

Preparing to Migrate the Database

Complete the tasks in this section before you export the database.

Create an Empty Database on the Target Machine

Create an empty ORACLE database on the target machine before you begin the migration. If you have not done so already, make sure you install the Oracle8i Server software on the target machine. For supported platforms, use the Rapid Install to create the ORACLE_HOME. Otherwise, use the native Oracle 8i installation utility.

Attention: Always be sure you are using a version of the Oracle8i Server that is certified for use with Oracle Applications Release 11*i*.

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: Oracle8i Server and Oracle Tools Requirements, *Oracle Applications Concepts*

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 Oracle8i 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, be sure that all the steps included in the import process 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, *Oracle8i Utilities*

Run the Preparatory Scripts for the Target Database

Five preparatory scripts are required. Three of these are run on the ORACLE_HOME of your database server, one on the WebDB 2.5 ORACLE_HOME, and the fifth is run from the Applications APPL_TOP. Run the following scripts if you are already on Release 11i or are upgrading to Release 11i and are performing this migration as part of the Catagory 3 pre-upgrade step in the *Upgrading Oracle Applications manual*.

Attention: If you are migrating your database as a pre-upgrade Category 1 step in preparation for an upgrade to Release 11i, do not run the five preparatory scripts listed below. Follow the instructions in the *Upgrading Oracle Applications manual* about applying the interoperability patches.

Additional Information: *Upgrading Oracle Applications*

Run preparatory scripts - database server We provide scripts that create objects on the database server that the RDBMS and other technology stack components require. The scripts are located in your APPL_ TOP in the admin directory. They must run in an environment that is different from the standard APPL_TOP, so you must copy them to a directory outside of APPL_ TOP and reset your environment before you run them. Each script creates one or more spool files and places them in the directory you ran the script from. Make sure you have permission to write to that directory. Check the spool file for errors after you run each script. Some warnings that you may see in the spool file are allowable, such as:

```
ORA-00955: name is already used by an existing object
ORA-00942: table or view does not exist
ORA-01430: column being added already exists in table
ORA-01434: private synonym to be dropped does not exist
```

The following scripts are provided. Run them in the ORACLE_HOME on the database server.

Script	Description
addbxxx.sql	RDBMS SYS schema setup script. Also installs objects for JAVA and Spatial options.
adsyxxx.sql	RDBMS SYSTEM schema setup script.
addbctx.sql	Oracle interMedia setup script.

Because we provide new versions of addbxxx.sql and adsysxxx.sql for each new RDBMS release (for example, addb816.sql and adsys816.sql for Release 8.1.6 of the Oracle Data Server), we refer to them as addbxxx.sql and adsyxxx.sql in the following instructions.

> **Attention:** Make sure you obtain the scripts for the version of the target database if it is different from the source database.

1. Create an admin subdirectory under the apps directory in the ORACLE_HOME on your database server.
2. Copy addbxxx.sql, adsyxxx.sql, and addbctx.sql to this directory.
3. Set your environment to point to the ORACLE_HOME on your database server. You must access the database server directly. Do not attempt to run any of these scripts by way of Net8.
4. Run addbxxx.sql and adsyxxx.sql from Server Manager:

For UNIX users:

```
$ svrmgrl
SVRMGR> connect / as sysdba
SVRMGR> @addbxxx

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

For NT users:

```
C:\apps\admin> svrmgrl
SVRMGR> connect / as sysdba
SVRMGR> @addbxxx
```

```
C:\apps\admin> svrmgrl
$ svrmgrl
SVRMGR> connect <SYSTEM username>/<SYSTEM password>
SVRMGR> @adsyxxx
```

5. Run addbctx.sql from SQL*Plus. UNIX users should connect as SYSTEM, and NT users should connect as INTERNAL.

```
<SYSTEM username>/<SYSTEM password @addbctx.sql <Remove ConText> \
<Default Tablespace> <Temporary Tablespace> <Intermedia Shared Library>
```

The arguments are as follows:

Argument	Description
Remove ConText	TRUE if you are upgrading from an earlier release and need to remove an existing installation of Oracle ConText. Otherwise, FALSE.
Default Tablespace	Default tablespace for the Oracle InterMedia schema (CTXSYS). If the CTXSYS tablespace does not exist, you need create it. The size required is 10 MB.
Temporary Tablespace	Temporary tablespace for the Oracle InterMedia schema (CTXSYS).
interMedia Shared Library	Full pathname of the Oracle interMedia shared library. On UNIX, for example, you would enter $ORACLE_HOME/ctx/lib/libctxx8.so.

Note: If you run this script inside SQL*Plus instead of from the SQL*Plus command line, you must enter the full pathname for the interMedia Shared Library parameter. SQL*Plus does not evaluate the ORACLE_HOME variable when used with a parameter.

Additional Information: *Oracle8i interMedia, Spatial, Time Series, and Visual Information Retrieval Options*

Install database objects for the Oracle HTTP server We provide a script to create objects on the database server required by the Oracle HTTP server. The script is admodpls.sql, located in your APPL_TOP in the admin directory.

1. In your iAS ORACLE_HOME, create an apps directory and an admin subdirectory under it..
2. Copy admodpls.sql to apps/admin.
3. Set your environment to point to your iAS ORACLE_HOME. You must access the database server using Net8.
4. Go to apps/admin and run admodpls.sql from SQL*Plus:

For UNIX users:

```
$ sqlplus <SYS username>/<SYS password @admodpls.sql <SYS password>\
<Default Tablespace> <Temporary Tablespace> <Connect String>
```

For NT users:

```
C:\> sqlplus <SYS username>/<SYS password @admodpls.sql <SYS password>\
 <Default Tablespace> <Temporary Tablespace> <Connect String>
```

The arguments are as follows:

Argument	Description
SYS password	Password for your SYS schema.
Default Tablespace	Default tablespace for the public web applications schema (OWAPUB). If the OWAPUB tablespace does not exist, you need create it. The size required is 10 MB
Temporary Tablespace	Temporary tablespace for the public web applications schema (OWAPUB)
Connect String	The Net8 connect string required to connect to your Oracle Applications database. You must provide a correct value for this argument even if you have LOCAL/TWO_TASK set.

Run adsysapp.sql The adsysapp.sql script creates Applications-specific objects in the SYSTEM schema. Run it in SQL*Plus as SYSTEM, *after* the other two scripts have run to completion. You *must* be in the admin directory of your APPL_TOP when you run this script. It takes one argument, the password for the SYSTEM user:

For UNIX users:

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

For NT users:

```
C:\> cd %APPL_TOP%\admin
C:\> sqlplus <SYSTEM username>/<SYSTEM password> @adsysapp.sql <SYSTEM password>
```

The script creates a spool file and places it in the directory you ran the script from. Make sure you have permission to write to that directory. Check the spool file for errors after you run this script.

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 Oracle8i Server Export utility to export the entire database from the source machine. You can export the database as a whole or in pieces.

Additional Information: Export, *Oracle 8i 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 or exceeds any platform-specific file system limitation.

Additional Information: *Oracle 8i Platform Specific Documentation*

To export the entire database, move to the directory where you want to create the export file and execute the following command. Here, <SYS password> is the password for the SYS ORACLE user:

For UNIX users:

```
$ exp sys/<SYS password> file=full.dmp \
  full=y buffer=1000000 grants=y log=<xxxx>.log
```

For NT users:

```
C:\> exp sys/<SYS password> file=full.dmp \
     full=y buffer=1000000 grants=y log=<xxxx>.log
```

When the export is finished, examine the log file for errors.

Exporting the Database By Schema

If a full database export file will not fit on one disk or in one file, or you want to reduce downtime, then export the database to multiple export files. You first export the database structure, then export data for 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, <SYS password> is the password for the SYS ORACLE user:

   For UNIX users:

   ```
   $ exp sys/<SYS password> file=fullnorows.dmp \
     buffer=1000000 grants=y rows=n log=<xxxx>.log full=y
   ```

   For NT users:

   ```
   C:\> exp sys/<SYS password> file=fullnorows.dmp \
        buffer=1000000 grants=y rows=n log=<xxxx>.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 Applications products, as well as the Oracle Applications Public Access schema, and any custom schemas. 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:

   For UNIX users:

   ```
   $ exp <ORACLE username>/<ORACLE password> \
     file=<ORACLE username>.dmp buffer=1000000 log=<xxxx>.log
   ```

   For NT users:

   ```
   C:\> exp <ORACLE username>/<ORACLE password> \
        file=<ORACLE username>.dmp buffer=1000000 log=<xxxx>.log
   ```

Suggestion: If you have enough hardware capacity, 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 there were errors in the export and you are not certain how to resolve them. Contact *Oracle Support Services* if this occurs.

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, *Oracle 8i 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:

For UNIX users:

```
$ imp SYSTEM/<SYSTEM password> file=full.dmp buffer=1000000 \
  commit=y full=y ignore=y log=<xxxx>.log
```

For NT users:

```
C:\> imp SYSTEM/<SYSTEM password> file=full.dmp buffer=1000000 \
     commit=y full=y ignore=y log=<xxxx>.log
```

When the import finishes, examine the log file for errors.

Importing the Database By Schema

If you exported the database in pieces, recreate the database structure from the full database export with no data. Then, restore the data by exporting each ORACLE schema.

1. Import database structure with no data.

Enter the following command to import the full database export with no data:

For UNIX users:

```
$ imp system/< SYSTEM password> file=fullnorows.dmp \
buffer=1000000 commit=y full=y indexes=n ignore=y log=<xxxx>.log
```

For NT users:

```
C:\> 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.

   Move to the directory that holds the ORACLE schema export files and enter the following command for each file:

   For UNIX users:

   ```
   $ imp <ORACLE username>/<ORACLE password> file=<ORACLE username>.dmp \
   buffer=1000000 commit=y indexes=y ignore=y log=<xxxx>.log
   ```

   For NT users:

   ```
   C:\> imp <ORACLE username>/<ORACLE password> file=<ORACLE username>.dmp \
   buffer=1000000 commit=y indexes=y ignore=y log=<xxxx>.log
   ```

   Suggestion: If 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 there were errors in the import 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. These missing objects and grants will be created during the subsequent step of running AD Administration against the imported database.

> **Note:** If you have products other than Oracle Applications installed in your database that access objects in SYS and 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, update 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 must update this information.

1. 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 original tablespace no longer exists, you must update the value of these columns to an existing tablespace.

   First look up the ORACLE_ID for the schema with the following command:

   For UNIX users:

```
$ sqlplus <APPS username>/<APPS password>
SQL> select ORACLE_ID
     2> from FND_ORACLE_USERID
```

```
     3> where ORACLE_USERNAME='<schema whose objects were moved>';
```

For NT users:

```
C:\> sqlplus <APPS username>/<APPS password>
SQL> select ORACLE_ID
      2> from FND_ORACLE_USERID
       3> where ORACLE_USERNAME='<schema whose objects were moved>';
```

Then issue the following SQL statement in SQL*Plus, connected to the APPS schema:

For UNIX users:

```
$ sqlplus <APPS username>/<APPS password>
SQL> update FND_PRODUCT_INSTALLATIONS
      2> set TABLESPACE='<new ablespace>', INDEX_TABLESPACE='<new index tablespace>'
     3> where ORACLE_ID='<ORACLE_ID from above>';
```

For NT users:

```
C:\> sqlplus <APPS username>/<APPS password>
SQL> update FND_PRODUCT_INSTALLATIONS
      2> set TABLESPACE='<new ablespace>', INDEX_TABLESPACE='<new index tablespace>'
     3> where ORACLE_ID='<ORACLE_ID from above>';
```

For example, if you imported all tables and indexes owned by the PO2 schema into the PO2DNEW and PO2XNEW tablespaces respectively, you would first enter the following SQL command to determine the ORACLE_ID:

```
SQL> select ORACLE_ID
     2> from FND_ORACLE_USERID
     3> where ORACLE_USERNAME='PO2';
```

This command will return a value for ORACLE_ID. Let's assume the value is '10037'. Use this value for the second SQL command to update the table.

```
SQL> update FND_PRODUCT_INSTALLATIONS
     2> set TABLESPACE='PO2DNEW', INDEX_TABLESPACE='PO2XNEW'
     3> where ORACLE_ID='10037';
```

2. If you want to change the tablespace where new tables or indexes are created for a product.

If you want to separate 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 PAY created in PAYD, issue the following update statement:

```
SQL> update FND_PRODUCT_INSTALLATIONS
     2> set TABLESPACE='PAYD' where APPLICATION_ID=801;
```

Application IDs can be found in the view FND_APPLICATION_VL.

> **Additional Information:** *Oracle Application Object Library Technical Reference Manual*

GL_STORAGE_PARAMETERS In the GL_STORAGE_PARAMETERS table, 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, issue 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 from your administration server and connect to the *new* database you just finished importing to perform the following tasks *in the order shown* here:

- Check SYS.DUAL table
- Recreate grants and synonyms for APPS schema(s)
- Compile APPS schema(s)

If the export or import processes failed to recreate some derived Applications objects or grants properly, you may want to perform these additional steps:

- Maintain Multiple Reporting Currency schema(s)

  Additional Information: Chapter 2 of this manual

- Compile flexfield data in AOL tables or platform-specific data recreation

Perform Post-installation Steps on the Target Machine

If the database name or Net8i database alias for your target database differs from your source database, you must perform some database-specific post-installation steps. You may also need to perform the following steps, depending on the components you have installed.

Additional Information: Finishing Your Installation, *Installing Oracle Applications*

- Create an Oracle Web Server Database Access Descriptor (DAD)
- Define Oracle Forms Cartridge parameters
- Configure Developer Metrics Server
- Modify the initial HTML signon file

It may be easier to use the Rapid Install Reconfigure option rather than performing the above steps manually.

Additional Information: Reconfigure, *Installing Oracle Applications*

Additionally, if you are specifying the ORACLE_SID or TWO_TASK (ORACLE_SID or LOCAL for NT) 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 chapter

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.

Database Initialization Parameters

This section lists the database initialization parameters for Oracle Applications Release 11*i* certified with Oracle Data Server 8*i* and the init.ora parameters required to use the cost-based optimizer (CBO). The init.ora is a reference file that contains the major initialization parameters used in Oracle Applications. Most of these parameters are defaulted or set during the course of an installation or upgrade. Some of the parameters will vary according to the configuration of the Oracle Applications environment at individual sites.

> **Additional Information:** *Oracle8i Parallel Server Setup and Configuration Guide*

General init.ora Parameters

The following section provides a guide to the required parameters. The actual values chosen are specific to each customer site.

Database parameters

The database parameters define the name of the database and the names of the control files. The database name is established when the database is built, and for most customers, matches the instance name. It should not normally be necessary to change the database name, except for database cloning.

There should be at least two control files, preferably three, located on different disks. The control files can dynamically grow, so allow at least 20M per file for growth.

```
db_name                         = TST115
control_files                   = /d01/TST115_DB/cntrltst115.dbf,
                                  /d02/TST115_DB/cntrltst115_2.dbf
```

Database block size

The standard block size for Oracle Applications is 8K. However, this can vary, depending on the O/S block and stripe size. For Oracle Parallel Server, a 4K block size is recommended.

Existing databases with 2K or 4K block sizes that are not experiencing performance problems need not be rebuilt. Databases experiencing performance problems (other than those associated with the network, client, or application) should be rebuilt when chaining becomes a significant component of I/O.

```
db_block_size                   = 8192
```

Compatible

Compatibility should normally be the current release number. Some new RDBMS features used by Oracle Applications function only if compatible is set to 8.1.6 or above.

```
compatible                      = 8.1.6
```

Required parameters

These parameters are required for Oracle Applications and MUST NOT be changed. Do not set the fixed_date parameter.

```
row_locking                     = always
nls_date_format                 = DD-MON-RR
```

NLS and character sets

These parameters are specific to the default language the ORACLE server uses, however nls_sort and nls_numeric_characters normally should not be changed.

```
nls_numeric_characters          = "."
nls_sort                        = binary
nls_language                    = american
nls_territory                   = america
```

Auditing and Security

Logon auditing is very useful in determining the I/O profile of batch (concurrent manager) processes. This information will be available in FND_CONCURRENT_ REQUESTS in a later release of Oracle Applications.

The cost of logon auditing is minimal, and the only additional requirement is for a housekeeping procedure to periodically purge the SYS.AUD$ table. Statement level auditing is not recommended.

Some products require max_enabled_roles to be set. It should be set to a minimum of 40, although higher values are quite acceptable.

```
#audit_trail                   = true          # if you want auditing.
max_enabled_roles               = 40            # Some modules depend on
                                                # this feature.
```

Dump parameters

These specify the destination of the trace and core files, and typically point to the appropriate OFA-compliant trace directories. The maximum size of a dump file can

be changed at the session level, and prevents a trace file from using an excessive amount of disk space.

```
user_dump_dest                 = ?/rdbms/log
background_dump_dest           = ?/rdbms/log
core_dump_dest                 = ?/rdbms/log
max_dump_file_size             = 10240 # trace file size
```

Timed statistics

On most platforms, enabling timed statistics has minimal effect on performance. There are a handful of exceptions. It can be enabled / disabled dynamically at both the system and session level.

This information is used by many options, including SQL_TRACE, Oracle Trace, statspack and Oracle Enterprise Manager.

```
timed_statistics               = true
```

Trace parameters

_trace_files_public Setting this parameter to TRUE makes RDBMS trace files publicly readable. Otherwise, only the operating system user who owns the RDBMS files will be able to read the trace files.

```
_trace_files_public            = TRUE
```

sql_trace Should be set at the session level, not for the entire database.

Oracle Trace Future versions of Oracle Applications will use Oracle Trace as part of end-to-end performance analysis. The required parameters will be documented at a later date.

Rollback segments

Rollback segments can be brought online or taken offline on demand. However, the normal rollback configuration, typically OLTP, should be included here. The number of rollback segments is dependent on the number of concurrent OLTP transactions, but 6 to 12 is a typical profile.

If rollback contention is seen in the bstat report, add more rollback segments. It is generally recommended to create private rather than public rollback segments.

```
rollback_segments              = (rbs1,rbs2,rbs3,rbs4,rbs5,rbs6,rbs7,rbs8,rbs9)
```

PL/SQL parameters

The utl_file_dir must be set per the installation manual.

```
#utl_file_dir = <dir1>,<dir2> ...
utl_file_dir  = sqlcom/out
```

Advanced Queuing (AQ)

AQ requires the Transaction Manager (TM) process to handle delayed messages. A number of Application modules use AQ, including workflow.

```
aq_tm_processes                 = 1
```

Archiving

Log archiving is part of a good backup strategy, and therefore highly recommended. Archiving parameters, including destination (optionally multiple destinations in 8*i*)

```
# log_archive_start             = true      # if you want automatic archiving
```

Events

Events are used by Oracle Support and Development. They should only be set as requested. Many exist for historical reasons. The events in the following list *are not* needed for Release 11*i*.

```
#event                 = "10210 trace name context forever, level 2"
#event                 = "10211 trace name context forever, level 2"
#event                 = "10235 trace name context forever, level 1"
#event                 = "10246 trace name context forever, level 2"
#event                 = "7267 trace name errorstack level 4"
#event                 = "3106 trace name errorstack level 4"
#event                 = "10076 trace name context forever, level 1"
```

Platform-specific parameters

The spin_count parameter is used on SMP platforms. It determines how long to spin trying to acquire exclusive access to low-level SGA data structures. With Oracle 8*i*, this parameter is undocumented and *should never* be set except at the request of Oracle Support.

Fixed SGA init.ora Parameters

The fixed SGA parameters represent resources that have their size fixed on startup. If the maximum size is reached (for example, the number of sessions), then the resource is unavailable until freed by the instance.

Processes/sessions

Oracle Applications NCA users require two sessions per database process. Either explicitly set sessions to twice processes, or double the number of processes.

The other parameters depend on the specific installation, but the values given are typical of many Oracle Applications customers.

```
processes                = 75    # Max. no. of users x 2
db_files                 = 500   # Max. no. of database files
dml_locks                = 500
open_cursors             = 300   # Consumes process memory, unless
                                 # using MTS.
enqueue_resources        = 5000  # Max. no of concurrent database locks.
```

Buffer Cache

The buffer cache requires (db_block_size x db_block_buffers) bytes within the SGA. Its sizing can have a significant effect on performance. Values less than 5000 are unrealistic for most customers, and can be increased as memory permits.

You can also divide the buffer cache into pools, for objects (data blocks) that should be kept in the cache and those that should be recycled. If the keep buffer pool is configured, it caches only those objects (tables) marked for the keep buffer pool.

```
db_block_buffers         = 5000
```

Log Writer

The log writer parameters control the size of the log buffer within the SGA and how frequently the redo logs are checkpointed (all dirty buffers written to disk to create a new recovery point).

The log buffer can be a megabyte or more but given the commit frequency in an OLTP environment, little benefit is achieved above 5M. It must be a multiple of redo block size, normally 512 bytes.

The checkpoint interval and timeout control the frequency of checkpoints. Time-based checkpointing is normally the best approach.

```
log_checkpoint_timeout   = 72000 # Checkpoint at least every 20 mins.
log_checkpoint_interval  = 100000
log_buffer               = 1048576
```

Sort Area

The sort area is allocated from process memory, unless using Multi-threaded Server (MTS). This parameter can have a dramatic effect on performance. Set too low, users

will suffer with excessive disk sorts; set too high, there is a real chance of significant paging.

The recommended values for OLTP range from 256K to 2M, depending on available memory. The parameter can be changed dynamically for specific batch jobs, such as index rebuild and analyze.

```
sort_area_size                  = 256000
```

Hash Area Size

The hash area is allocated as per sort_area_size, and the default is 2 x sort_area_size. This should be fine for most customers, but can be changed dynamically for specific jobs. It can potentially have the same dramatic effect on memory as sort_area_size.

Shared Pool

The shared pool size is key to controlling contention between SQL and PL/SQL objects. 300M is a reasonable estimate for Release 11*i*, with 30M for the reserved area (10%). In versions 8.0.6 and 8.1.6 of the server, the min_alloc can be set to 4100, which can help with PL/SQL allocations. In most circumstances, the default of 5000 is acceptable.

In 8.1.6 the shared_pool_reserved_min_alloc is undocumented.

```
shared_pool_size                 = 300000000
shared_pool_reserved_size        = 30000000
#_shared_pool_reserved_min_alloc = 4100
```

cursor_space_for_time Reduces contention within the shared pool but requires at least a 50% increase in the shared pool. Set this parameter only on the advice of Oracle Support.

Java Pool

In Release 11*i*, certain products use Java Stored Procedures. If your installation uses these products, an initial setting for the Java Pool is 50M, but this may need to be increased as required. Java Stored Procedures are invoked by way of MTS, so MTS and the large_pool must also be configured.

```
#java_pool_size                 = 50000000
#large_pool_size                = 50000000
```

Cost-Based Optimization (CBO) init.ora Parameters

Because Oracle Applications Release 11*i* runs in cost-based optimization mode, the init.ora parameters in this section must be set. Most, if not all, of the parameters are dynamic, and can be set at the session level.

If you are an online user, there are profile options to change the parameter values. If you are a batch user, the concurrent request form allows you to specify an optimizer_mode. Other parameters are set by using profile options.

optimizer_features_enable

Typically, must be set to the current release, as Oracle Applications will rely on optimizer fixes and new features.

```
optimizer_features_enable       = 8.1.6
```

optimizer_mode

Prior to Release 11*i*, optimizer_mode was always set to *rule*. Now, *choose* is mandatory. Although Applications modules will set the optimizer mode to either first_rows or all_rows, according to whether they are online or batch, an Applications database *must be started* with the optimizer mode set to *choose*.

In general, the profile options will ensure that online users use first_rows and batch jobs use all_rows.

```
optimizer_mode                  = choose
```

> **Attention:** CBO requires accurate table and index statistics. FND_STATS should be run regularly. See Analyzing the Database in this chapter for details.

optimizer_undo_changes

This parameter is *not* dynamic and for RBO compatibility must remain as TRUE. For Release 11*i*, you should set it to FALSE. If custom code continues to use the RBO and experiences performance problems, setting this to TRUE should not affect CBO queries. This will be removed in a future release.

```
_optimizer_undo_changes         = false
```

_optimizer_mode_force

Must be set to TRUE. Forces recursive SQL (packaged SQL) to use the optimizer_ mode from the current environment.

```
_optimizer_mode_force             = true
```

db_file_multiblock_read_count

Many APPS customers have multiblock read count set at 16 or 32, depending on block size. For Release 11*i*, the recommended value is now 8, as this provides the best value for the CBO.

This parameter can be set at the session level, so specific batch jobs, index rebuilds, and analyze can take full advantage of the maximum available multiblock I/O.

```
db_file_multiblock_read_count    = 8
```

optimizer_max_permutations

The default (80000) can cause excessive parse times in some circumstances. The parameter must be set to 79000 or less in order to allow the optimizer to consider more than the starting table. The default of 80000 limits the number of starting tables that the optimizer considers.

```
optimizer_max_permutations       = 79000
```

_complex_view_merging

Enables the complex view merging feature which allows certain types of complex views (such as the Apps KFV) to be merged. This parameter is disabled by default, so it must be explicitly set.

```
_complex_view_merging            = TRUE
```

_push_join_predicate

Enables the push join predicate feature which allows the optimizer to push join predicates inside non-mergable views. This helps eliminate full table scans against the adjoining table of a non-mergable view. Pushing the join predicate allows the optimizer to promote an index on the table inside the view and utilize a nested loop join to the outer referencing table. Push join predicate is disabled by default, so it must be explicitly enabled.

```
_push_join_predicate             = TRUE
```

_sort_elimination_cost_ratio

Fixes the FIRST_ROWS bug 780376 which optimizes for the order by case as opposed to the selective index filters in the WHERE clause. Setting it to five forces the optimizer to only eliminate the sort when it is 1/5th the cost of the index probe (or conversely the index probe is 5 times as costly as the sort).

```
_sort_elimination_cost_ratio     = 5
```

_use_column_stats_for_function

Allows the optimizer to utilize dictionary statistics for columns that are involved in no-op expressions such as [col + 0] and [col || ''']. If this parameter is disabled (FALSE), the optimizer will employ internal default statistics for such complex expressions which can result in higher parse times and more expensive execution plans.

```
_use_column_stats_for_function  = TRUE
```

_like_with_bind_as_equality

This parameter forces the optimizer to treat expressions of the form [indexed-column like :b1] similar to [indexed-column = :b1]. Oracle Apps has many queries which use the LIKE operator on indexed columns with binds. Since binds are involved, the CBO assigns internal default selectivity estimates for the LIKE operator (5%), and hence does not consider the index selective.

```
_like_with_bind_as_equality      = TRUE
```

_or_expand_nvl_predicate

Allows the optimizer to probe on an index for a column involved in an nvl() function as an r-value.

If enabled, the optimizer transforms expressions of the form [p.project_id = nvl(:b1,p.project_id)] into an OR expanded UNION where one-side of the UNION contains the predicate ((:b1 is not null) and (p.project_id =:b1)) and the second branch of the UNION contains the predicate (:b1 is null). Therefore, if a value for the project_id bind is supplied, the optimizer executes the first branch of the UNION and probes on the project_id index.

This improves performance for the Apps legacy code which employs the nvl() construct on indexed columns.

```
_or_expand_nvl_predicate        = TRUE
```

_push_join_union_view

Allows the optimizer to push join predicates inside non-mergable views which contain UNION ALL set operators. This improves query execution performance for queries joining to views which contain UNION ALL operators.

```
_push_join_union_view           = TRUE
```

_table_scan_cost_plus_one

Increases the cost of a full table scan by one in order to eliminate ties between a full table scan on a small lookup table and the cost of a unique or range index scan on the lookup table.

```
_table_scan_cost_plus_one       = TRUE
```

_fast_full_scan_enabled

Used to disable fast full scans.

```
_fast_full_scan_enabled         = FALSE
```

_ordered_nested_loop

Helps to reduce the cost of a nested loop join/index probe when the left side of the join input is being satisfied by way of an index or sort row source.

```
_ordered_nested_loop            = TRUE
```

optimizer_percent_parallel

Sets the amount of parallelism to include in the CBO cost function. The default is zero, and typically should not be changed. It is necessary to ensure that parallel query is not included in costing.

```
optimizer_percent_parallel      = 0
```

query_rewrite_enabled

Required for materialized views and function based indexes, which are used in some Release 11*i* products.

```
query_rewrite_enabled           = true
```

always_anti_join

Specifies the join method to be used for anti-joins. Anti-joins can be used for queries which require anti-join semantics such as queries which contain a NOT IN

sub-query. This parameter should not be altered from the default setting of NESTED_LOOPS.

```
always_anti_join                = NESTED_LOOPS
```

always_semi_join

Specifies the join method to be used for semi-joins. The optimizer can transform a query which consists of an explicit join into a semi-join provided that the semantics of the query permit such a transformation. The transformation of a regular equi-join into a semi-join is equivalent to replacing the join with an EXISTS sub-query. This parameter should not be altered from the default setting of NESTED_LOOPS.

```
always_semi_join                = NESTED_LOOPS
```

_sqlexec_progression_cost

Specifies the cost threshold for the progress meter. Non-zero values can prevent cursors from being shared when timed_statistics=TRUE.

```
_sqlexec_progression_cost       = 0
```

Special Features

Multi-threaded Server and Parallel Query are special Oracle8*i* features. The following are init.ora parameters related to these features.

Multi-threaded Server (MTS)

Most Oracle Applications customers *do not* need to use MTS, so the default is to leave it disabled. If MTS is used, it can have a dramatic effect on the SGA, as session memory, including sort and cursor areas, is taken from the SGA.

MTS is required for MVS, NT, and Java Stored Procedures (JSP). In the initial release of Release 11*i*, JSPs will not be widely used. When MTS is required for JSP, then only JSP calls need to go by way of MTS. Configuring MTS (and thus JSP) requires the large pool to be allocated. When using JSP, large_pool_size should be a minimum of 50M.

Parallel Query

Not usually required for OLTP systems. If enabled, tables / indexes *must not* have degree set. Use hints to enable parallel query.

To use the parallel option with DBMS_STATS (FND_STATS), parallel_max_servers must be set — it is not a dynamic parameter. It should be set based on the number

of available CPUs. The following examples may need to be altered. They indicate reasonable values for DBMS_STATS.

Parallel Query uses the large_pool for message buffers, so the large_pool_size may need to be specified.

```
#parallel_max_servers       = 8
#parallel_min_servers       = 0
```

Rollback Segments

The following table 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. 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.

Action	Number of Segments	Extent Size	Minimum Size
Upgrading	1 per each worker	1 MB	60 MB
Normal Use	1 for each 25 users plus 1 for each concurrent manager	256 K	13 MB

Rollback segment tablespaces must be locally managed with uniform extent size equal to 256K and minimum extent of 50.

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 AutoUpgrade 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.

Product Tablespace Requirements

The table below 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 upgrade requires a different sizing factor, we recommend you use a sizing factor of 100% for all products.

Rapid Install and AutoUpgrade install 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 AutoUpgrade Database Parameters screen. By default, all non-licensed products install at 100% sizing factor.

The total minimum tablespace for all product tables at 100% sizing is 10462 MB and for indexes is 11282 MB.

Table 6–1 Product Tablespace Requirements

Product Family	Abbreviation	Product Name	MOA/SOA	Table Size	Index Size
Applications Technology	AD	Applications DBA	SOA	469	341
	AK	Oracle Common Modules	SOA	44	17
	ALR	Oracle Alert	SOA	11	10
AD, AU, FND, SHT are in the APPLSYS schema (Sizing is stated in AD)	AU	Applications Utilities	SOA	-	-
	AZ	Application Implementation Wizard	SOA	2	3
AD, AU, FND, SHT are in the APPLSYS schema (Sizing is stated in AD)	FND	Application Object Library	SOA	-	-
	ICX	Self-Service Web Applications	SOA	42	37
AD, AU, FND, SHT are in the APPLSYS schema (Sizing is stated in AD)	SHT	Shared Technology	SOA	-	-
CRM	AMS	Marketing	MOA	134	305
	AMV	Marketing Encyclopedia System	MOA	97	128
in the OSM schema	AS	Oracle Sales and Marketing	MOA	40	100
	ASF	Field Sales	MOA	1	1
	ASG	Gateway for Mobile Devices	MOA	4	4
	ASO	Order Capture	MOA	12	24
	AST	TeleSales	MOA	1	1
	BIC	Customer Intelligence	MOA	30	30
	BIL	Sales Intelligence	MOA	2	2
	BIM	Marketing Intelligence	MOA	35	41
	BIX	Call Center Intelligence	MOA	3	1
	CCT	Telephony Manager	MOA	5	3

Product Family	Abbreviation	Product Name	MOA/SOA	Table Size	Index Size
	CN	Oracle Sales Compensation	MOA	174	176
	CS	Oracle Service	MOA	180	203
	CSC	Customer Care	MOA	5	8
	CSD	Depot Repair	MOA	193	3
	CSF	Field Service	MOA	916	1
	CSP	Spares Management	MOA	108	14
	CSR	Scheduler	MOA	97	1
	CSS	Support	MOA	9	1
	CUA	CRL Financials - Assets	MOA	1	1
	CUF	CRL Financials	MOA	1	1
	CUI	CRL SupplyChain - Inventory	MOA	1	1
	CUN	CRL SupplyChain - NATS	MOA	4	2
	CUP	CRL SupplyChain - Purchasing	MOA	1	1
	CUS	CRL SupplyChain	MOA	1	1
	FPT	TeleBusiness for Financial Services	MOA	531	94
	IBA	iMarketing	MOA	1	1
	IBE	iStore	MOA	41	3
	IBP	iBill and Pay	MOA	1	1
	IBU	iSupport	MOA	1	1
	IBY	iPayment	MOA	53	45
	IEB	Interaction Blending	MOA	3	3
	IEM	eMail Center	MOA	27	22
	IEO	Call Center Technology	MOA	3	1
	IES	Scripting	MOA	3	4
	IEU	Universal Work Queue	MOA	99	99
	IEX	Collections	MOA	7	5
	IPA	CRL Financials - Projects	MOA	33	32
	JTF	CRM Foundation	MOA	457	397
	ME	Maintenance, Repair, and Overhaul	MOA	1	1
	MWA	Mobile Applications	MOA	1	1
	OKC	Contracts Core	MOA	26	39
	OKS	Contracts Service Module	MOA	2	2

Product Family	Abbreviation	Product Name	MOA/SOA	Table Size	Index Size
	OKX	Contracts Integration	MOA	1	1
	OZF	Funds & Budgets	MOA	17	54
	OZP	Trade Planning	MOA	1	1
	OZS	iClaims	MOA	1	1
	XDP	SDP Provisioning	MOA	656	1219
	XNC	Sales for Communications	MOA	4	1
	XNM	Marketing for Communications	MOA	1	1
	XNP	SDP Number Portability	MOA	373	1196
	XNS	Service for Communications	MOA	97	1
Financials	ABM	Activity-based Management	SOA	1	1
	AP	Oracle Accounts Payable	MOA	113	212
	AR	Oracle Accounts Receivable	MOA	506	1477
	AX	Global Accounting Engine	SOA	39	73
	BSC	Balanced Scorecard	SOA	10	9
	CE	Oracle Cash Management	MOA	56	17
	EAA	SEM Exchange	MOA	1	1
	EVM	Value-based Management	SOA	1	1
	FA	Oracle Assets	SOA	119	93
	FEM	Strategic Enterprise Management	SOA	24	21
	FII	Financial Intelligence	MOA	1	1
	FRM	Report Manager	SOA	1	1
	FV	Federal Financials	MOA	1	1
	GL/RG	Oracle General Ledger	SOA	74	62
	PA	Oracle Projects	MOA	1117	1808
	PN	Property Manager	SOA	11	35
	RG	Report Generator	SOA	29	25
	XLA	Common Accounting Modules	SOA	1	1
	XTR	Treasury	SOA	178	195
Human Resources	BEN	Benefits	SOA	58	207
	DT	DateTrack (in HR schema)	SOA	-	-
	FF	FastFormula (in HR schema)	SOA	-	-

Product Family	Abbreviation	Product Name	MOA/SOA	Table Size	Index Size
	GHR	Government Human Resources (in HR schema)	SOA	-	-
	HRI	Human Resources Intelligence	MOA	1	3
	HXC	Time Capture	SOA	1	1
	HXT	Time Management	SOA	8	17
	OTA	Human Resources (Training)	SOA	3	10
DT, FF, GHR, PAY, PER, PQH, PQP in HR schema	PAY (default=HR)	Human Resources (Payroll) (in HR schema)	SOA	398	652
	PER (default=HR)	Oracle Human Resources (Personnel) (in HR schema)	SOA	-	-
	PQH	Public Sector HR (in HR schema)	SOA	-	-
	PQP	Public Sector Payroll (in HR schema)	SOA	-	-
	SSP	Oracle Statutory Sick Pay	SOA	3	5
Localizations	JA	Financials for Asia/Pacific	MOA	10	11
	JE/JG	Financials for Europe	MOA	7	9
	JG	Regional Financials		2	2
	JL	Oracle Financials for Latin America	MOA	21	29
Manufacturing / Distribution	BIS	Business Intelligence System	SOA	9	10
	BOM	Oracle Bills of Material	MOA	108	108
	CHV	Oracle Supplier Scheduling	MOA	1	1
	CRP	Oracle Capacity	MOA	6	9
in BOM schema	CST	Oracle Cost Management	MOA	-	-
	CZ	Oracle Product Configurator	MOA	10	22
	EC	Oracle e-Commerce Gateway	MOA	115	51
	ECX	Oracle XML Gateway	MOA	1	1
	ENG	Oracle Engineering	MOA	17	8
	ENI	Engineering Intelligence	MOA	1	1
	FLM	Flow Manufacturing	MOA	1	1
	INV	Oracle Inventory	MOA	145	208
	ISC	Supply Chain Intelligence	MOA	1	1
	MFG	Manufacturing Menu	MOA	1	1
	MRP	Master Scheduling	MOA	78	102

Product Family	Abbreviation	Product Name	MOA/SOA	Table Size	Index Size
	MSC	Supply Chain Planning	MOA	192	61
	MSD	Demand Planning	MOA	82	1
	MSO	Constraint Based Optimization	MOA	1	1
	OE	Order Entry	MOA	75	64
	OKE	Contracts for Projects	MOA	1	1
	ONT	Order Management	MOA	92	91
	OPI	Operations Intelligence	MOA	1	1
	PJM	Project Manufacturing	MOA	3	4
	PO	Oracle Purchasing	MOA	94	150
	POA	Purchasing Intelligence	MOA	1	1
	POM	Exchange	SOA	1	1
	QA	Quality	SOA	16	11
	QP	Advanced Pricing	MOA	30	53
	RHX	Advanced Planning Foundation	MOA	1	1
	RLA	Release Management	SOA	7	3
	RLM	Release Management	SOA	6	5
	VEA	Automotive	SOA	2	3
	VEH	Automotive	SOA	12	5
	WIP	Work in Process	MOA	107	100
	WMS	Warehouse Management Systems	MOA	20	45
	WPS	Manufacturing Scheduling	MOA	1	1
	WSH	Shipping Execution (Common)	MOA	27	57
	WSM	Shop Floor Management	MOA	13	1
Process Manufacturing	GMA	Process Manufacturing Systems	MOA	25	4
	GMD	Processing Manufacturing Product Development	MOA	22	12
	GME	Process Manufacturing Process Execution	MOA	34	11
	GMF	Process Manufacturing Financials	MOA	280	259
	GMI	Process manufacturing Inventory	MOA	106	58
	GML	Process Manufacturing Logistics	MOA	56	34
	GMP	Process Manufacturing Process Planning	MOA	455	55

Product Family	Abbreviation	Product Name	MOA/SOA	Table Size	Index Size
	GR	Process Regulatory Management	MOA	375	33
	PMI	Process Manufacturing Intelligence	MOA	1	1
Public Sector	GMS	Grants Management	MOA	1	1
	IGC	Commitment Administration	MOA	1	1
	IGF	Student Systems Financial Aid	MOA	1	1
	IGS	Oracle Student Systems	MOA	1	1
	IGW	Grants Proposal	MOA	1	1
	PSA	Public Sector Applications	MOA	1	1
	PSB	Public Sector Budgeting	MOA	1	1
	PSP	Labor Distribution	MOA	1	1

7

Maintaining the Applications File System

This chapter contains information about maintaining the Oracle Applications file system. It includes information about the following:

- Migrating the Oracle Applications File System
- Recovering Disk Space
- Server Processes
- Reserved Words

Migrating the Oracle Applications File System

This section lists the tasks you need to perform to relocate all or part of the Oracle Applications file system. These steps assume you are moving all product files to another machine, or moving all files for one or more Applications products to another file system 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. Also, you can move only the files in a product's subdirectory. The Applications environment structure does not directly support this, so in order to relocate files only in a subdirectory, 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 UNIX man page for the 'ln' command explains how to create symbolic links on your file system. NT does not support symbolic links.

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 Oracle Applications.

   After all users have logged off, 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 file systems. Use the *cp -r* command to move an entire directory tree at once. For example:

   For UNIX users:

   ```
   $ cp -r /d02/appl/115/gl /d05/appl2/115/gl
   ```

   For NT users:

   ```
   C:\> xcopy /s /e /i d:\appl\115\gl e:\appl2\115\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 <PROD>_TOP environment variables for the product files you moved to the new location. In the example, you would set the variable GL_TOP to the new location by doing the following:

   For UNIX users:

   ```
   $ GL_TOP=/d05/appl2/115/gl/11.5.0; export GL_TOP
   ```

   For NT users:

   ```
   C:\> set GL_TOP=D:\appl2\115\gl\11.5.0
   ```

   Do this only at the operating system prompt. Do not edit environment files yet.

4. Edit topfile.txt

The file topfile.txt in the admin directory under APPL_TOP contains an entry for each product. The entry consists of the product abbreviation and the product's base product top directory. The base product top directory is the directory under which the product directory is located. Often the base product top directory is APPL_TOP, but not always. For example, if GL_TOP is set to /d05/appl2/115/gl/11.5.0, the base product directory for GL is /d05/appl2/115.

Since you have changed the base product directory for one or more products, you must now edit topfile.txt to reflect this change.

Open the topfile.txt in an editor. Find the product short name(s) for the product(s) you moved. Edit the corresponding product top directory entry to match the new product top base directory for the product. Save your changes to topfile.txt and exit the editor.

Note: A careful naming strategy for database files and product directories eliminates data administration problems. We recommend adherence to the Optimal Flexible Architecture (OFA) rules and recommendations published in *The OFA Standard: Oracle8i for Open Systems.*

---

5. Recreate the main Applications environment file.

   Run AD Administration and choose Create Applications Environment File from the Maintain Applications Files menu. After answering several questions, AD Administration recreates your main environment file to reflect the new file system arrangement.

   Attention: If you previously customized your main Applications environment file, make a backup before running AD Administration so you can copy your customizations to the new file.

   Additional Information: Chapter 2 in this manual

6. Execute the 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 file system arrangement. Once you are

satisfied that everything is working properly, you can allow other users to access the system, and remove the files from the old directory to free up space.

Preparing to Migrate the File System to Another Machine

If you are moving your product files to a different machine, complete the following tasks before you move the file system.

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*i* on the source machine, obtain Release 11*i* 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

The preparation steps for this process are identical to the preparation steps required for Rapid Install. If the target machine has already been set up to use Oracle Applications, most of these steps will already be done.

> **Additional Information:** Starting Your Installation, *Installing Oracle Applications*

Unload Oracle Applications Product Files

You must use Rapid Install to unload the necessary files on the target machine.

> **Additional Information:** Upgrading Your Installation, *Installing Oracle Applications*

Completing the Migration

Follow these steps to complete the file migration process from one machine to another.

Perform Post-install Steps

The post-installation steps for this process are identical to the post-installation steps required for Rapid Install.

Additional Information: Finishing Your Installation, *Installing Oracle Applications*

Apply Patches

There may be patches you applied to the Oracle Applications file system that you are replacing. If this is the case, apply these patches to your new file system. If you migrated to a different platform, be sure that you apply these patches specific to the new platform. Also, apply any additional patches for that specific platform.

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: File System, *Oracle Applications Concepts*

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:

- For UNIX, the files in $APPL_TOP/admin/<db_name>/log and $APPL_TOP/admin/<db_name>/out. For NT, the files in %APPL_TOP%\admin\<db_name>\log and %APPL_TOP%\admin\<db_name>\out, where <db_name> is the name of the database in which the current product group is installed.

  These directories contain log and output files, respectively, for AutoUpgrade and other Oracle Applications utilities. You can back up and remove the files after you run AutoUpgrade 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:

- For UNIX, the files in $APPL_TOP/admin/preupg. For NT, the files in %APPL_TOP%\admin\preupg. This directory contains SQL*Plus and other files used to prepare products for an upgrade to the current software release.

> **Attention:** Do *not* remove any files under the <PROD>_TOP/admin directory. They 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. They 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 lib directories of your <PROD>_TOP. They are needed to relink programs if, for example, you update them using AutoPatch or upgrade the Oracle8-based technology stack. These files take approximately 40 MB of disk space on a system that has all products installed.

AutoPatch Backup Files

After you run AutoPatch to update Oracle Applications, you can delete old files that have been backed up in the patch subdirectory. If space permits, we recommend backing up and keeping these files. Delete them only after you have verified the patch was successful and fully tested the patched functionalities. The amount of space recovered depends on the number of files that were replaced.

Additional Information: After Running AutoPatch in Chapter 4

Executable Program Compression Utilities

Oracle Applications automatically uses certain utilities to reduce the size of executable programs. Therefore, you do *not* need to run the following 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.

Server Processes

Rapid Install sets up and configures the server processes for you during installation and stores a script for each one in the scripts subdirectory under the admin directory in the Common area of your file system. You use these scripts to start server processes when you use your Oracle Applications system. These scripts can be recreated at anytime by using the Reconfiguration option of Rapid Install.

Additional Information: *Installing Oracle Applications*

Starting Server Processes

When running Rapid Install, you chose the character set for each application tier server (such as the forms server, reports server, and the concurrent processing server). You must start these processes using the same character set you specified during Rapid Install. For example, if you installed the forms server with JA16SJIS, you must start the forms server in JA16SJIS. If you installed the reports server with JA16EUC, you must start it in JA16EUC.

Note: All scripts in this section contain environment-specific information. If you modify your environment from the Rapid Install defaults, the scripts may fail. You may need to edit the scripts before rerunning them.

Net8 Listener for Oracle8*i* Server (addlnctl.sh)

Rapid Install sets up and configures the Net8 database listener. It also provides a script so you can start / stop the listener, if necessary. This script is in the 8.1.6

ORACLE_HOME under appsutil/scripts. As the oracle user on the database server, type:

For UNIX users:
```
addlnctl.sh [start|stop] <SID>
```

For NT users:
```
addlnctl.sh [start|stop] <SID>
```

For example, the oracle user would start the Net8 listener with the following command:

For UNIX users:
```
addlnctl.sh start PROD
```

For NT users:
```
addlnctl.sh start PROD
```

Additional Information: *Oracle Net8 Administrator's Guide*

Forms Server Listener (adfroctl.sh)

The forms server listener defines the Oracle Forms Cartridge parameters. To start/stop the listener, log in to the system as the Applications login account on the forms server. Then, type:

For UNIX users:
```
adfroctl.sh [start|stop]
```

For NT users:
```
adfroctl.sh [start|stop]
```

Forms Metrics Server (adfmsctl.sh)

The Metrics Server keeps track of all forms servers in a given pool of forms servers and directs each form execution request to the least loaded server that is able to service requests in the given pool. To start/stop the Metrics Server, log in to the system as the Applications login account on the forms server. Then, type:

For UNIX users:
```
adfmsctl.sh [start|stop]
```

For NT users:
```
adfmsctl.sh [start|stop]
```

Forms Metrics Client (adfmcctl.sh)

The Metrics Client sends load information, such as the number of forms processes that are currently running on that machine, to the Metrics Server. The Metrics Client runs on each machine with a forms server. To start/stop the Metrics Client, log in to the system as the Applications login account on the forms server. Then, type:

For UNIX users:
```
adfmcctl.sh [start|stop]
```

For NT users:
```
adfmcctl.sh [start|stop]
```

Reports Server (adrepctl.sh)

The reports server processes requests to execute a report and returns the output of the report. To start/stop the reports server, log in as the Applications user on the concurrent processing server and type:

For UNIX users:
```
adrepctl.sh [start|stop]
```

For NT users:
```
adrepctl.sh [start|stop]
```

Report Review Agent (adalnctl.sh)

Oracle Applications uses the Report Review Agent to view concurrent processing files online. This utility is on the concurrent processing server and uses Net8 to communicate with the forms server. Rapid Install sets up and configures the Report Review Agent for you during installation. It names the Net8 listener APPS_<SID> and performs the following tasks:

- configures network files
- verifies file permissions

To start/stop the Report Review Agent, log in as the Applications user on the concurrent processing server and type:

For UNIX users:
```
adalnctl.sh [start|stop] <listener_name>
```

For NT users:
```
adalnctl.sh [start|stop] <listener_name>
```

The listener should always be the APPS listener name or the script may terminate another listener. For example, the oracle user would start/stop the Net8 listener on the concurrent processing server by adding the APPS_<SID> argument.

TCF SocketServer (adtcfctl.sh)

The Thin Client Framework (TCF) SocketServer is a middle-tier process that accepts connections from UI clients to their server components. The TCF architecture currently supports Hierarchy Editor applications such as the Object Navigator and the Function Security menu Viewer. The SocketServer process should always be running in a production installation.TCF SocketServer is configured by Rapid Install.

To start/stop the TCF SocketServer, log in as the Applications user on the web server and type:

For UNIX users:
```
adtcfctl.sh [start|stop]
```

For NT users:
```
adtcfctl.sh [start|stop]
```

Concurrent Managers (adcmctl.sh)

Rapid Install sets up and starts the concurrent managers for you. It also provides a script, so you can restart them later, if needed. You can use this script to start/stop the concurrent manager for a database. During the set up process, Rapid Install performs the following tasks:

- sets up electronic mail messaging
- creates common log and output directory and log and output files in each product top directory
- defines startup parameters
- restricts access

To run the script, log in as the Applications user on the concurrent processing server and type:

For UNIX users:
```
adcmctl.sh <APPS username>/<APPS password> [start|stop]
```

For NT users:
```
adcmctl.sh <APPS username>/<APPS password> [start|stop]
```

Additional Information: *Oracle Applications System Administrators Guide*

HTTP server Listener (adapcctl.sh)

The Oracle HTTP server listener processes URL requests to execute forms based Applications and Self-Service Web Applications. Rapid Install sets up and starts the HTTP server listener. It also provides a script so you can restart it later, if necessary. To start/stop the HTTP server listener, log in to the system as the Applications login account on the web server. Then, type:

For UNIX users:

```
adapcctl.sh [start|stop]
```

For NT users:

```
adapcctl.sh [start|stop]
```

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, ORACLE, or an Oracle Applications product short name, such as FND, AD, or GL. Also, do not use operating system-specific words such as the UNIX commands mv, cp, or ln.

- A - / - B -	ABM_TOP	AD	AD_TOP
ADJKYPRG	ADJREOPTS	ADJVAPRG	AK_TOP
ALR_TOP	AMS_TOP	AMV_TOP	AP_TOP
APPCPNAM	APPL_CONFIG	APPL_TOP	APPLBIN
APPLCFRM	APPLCSF	APPLDCP	APPLFENV
APPLFRM	APPLFULL	APPLHLP	APPLIMP
APPLINC	APPLINP	APPLINST	APPLLDR
APPLLIB	APPLLOG	APPLMAIL	APPLMSG
APPLMSK	APPLNQS	APPLORB	APPLORC
APPLOUT	APPLPLS	APPLPLUS	APPLPTMP
APPLREG	APPLREP	APPLRGT	APPLRMF
APPLRSC	APPLSAV	APPLSHAR	APPLSQL
APPLSUB	APPLTMP	APPLUSR	APPMLANG
AR_TOP	AS_TOP	ASF_TOP	ASG_TOP

ASO_TOP	AST_TOP	AU	AU_TOP
AX_TOP	AZ_TOP	BEN_TOP	BIC_TOP
BIL_TOP	BIM_TOP	BIS_TOP	BIX_TOP
BOM_TOP	BSC_TOP		

- C - / - D -	CC	CCT_TOP	CE_TOP
CFLAGS	CHMOD	CHV_TOP	CLASSPATH
CLIBS	CN_TOP	CP	CRP_TOP
CS_TOP	CSC_TOP	CSF_TOP	CSP_TOP
CSR_TOP	CSS_TOP	CUA_TOP	CUF_TOP
CUI_TOP	CUN_TOP	CUP_TOP	CUS_TOP
CZ_TOP	DPFBUG	DT_TOP	

- E - / - F -	EAA_TOP	EC_TOP	ECX _TOP
ENG_TOP	EVM_TOP	FA_TOP	FDBDMCHK
FDCPCHK	FDMSGCHK	FDOPGOAL	FDSQLCHK
FEM_TOP	FF_TOP	FII_TOP	FILE
FLM_TOP	FND_CRT	FND_RT	FND_RTCP
FND_TOP	FND_ZOOM	FNDFAPP	FNDLANG
FNDMAPP	FNDMENU	FNDNAM	FNDOTHERLIBS
FORMS60_APPSLIBS	FORMS60_MAPPING	FORMS60_OUTPUT	FORMS60_PATH
FORMS60_SESSION	FPT_TOP	FRM_TOP	FV_TOP

- G - / - H -	GHR_TOP	GL_TOP	GMA_TOP
GMD_TOP	GME_TOP	GMF_TOP	GMI_TOP
GML_TOP	GMP_TOP	GMS_TOP	GR_TOP
GRAPHICS60_PATH	GWYUID	HRI_TOP	HXC_TOP
HXT_TOP			

- I - / - J -	IBA_TOP	IBE_TOP	IBP_TOP
IBU_TOP	IBY_TOP	ICX_TOP	IEB_TOP
IEM_TOP	IEO_TOP	IES_TOP	IEU_TOP
IEX_TOP	IGC_TOP	IGF_TOP	IGS_TOP
IGW_TOP	INCLUDE_FLAGS	INV_TOP	IPA_TOP
ISC_TOP	JA_TOP	JAVA_TOP	JDK
JE_TOP	JG_TOP	JL_TOP	JRE
JRI	JTF_TOP		

- L - / - M -	LD_LIBRARY_PATH	LDFLAGS	LFNDLIB
LN	LUSRIAP	LUSRLIB	LUSRPRG
LUSRSRW	MAILLIB	MCS	ME_TOP
MFG_TOP	MRP_TOP	MSC_TOP	MSD_TOP
MSO_TOP	MV	MWA_TOP	

- N - / - O -	NETLIBS	NLS_DATE_FORMAT	NLS_LANG
NLS_NUMERIC_CHARACTERS	NLS_SORT	OA_DOC	OA_HTML
OA_JAVA	OA_JDK_TOP	OA_MEDIA	OAD_TOP
OAH_TOP	OAM_TOP	OE_TOP	OKC_TOP
OKE_TOP	OKS_TOP	OKX_TOP	ONT_TOP
OPI_TOP	ORACLE_LINK	ORALIBS	OTA_TOP
OTHERLIBS	OZF_TOP	OZP_TOP	OZS_TOP

- P - / - Q -	PA_TOP	PATH	PAY_TOP
PCC	PCCFLAGS	PCCINC	PER_TOP
PJM_TOP	PLATFORM	PMI_TOP	PN_TOP
PO_TOP	POA_TOP	POM_TOP	PQH_TOP
PQP_TOP	PRGLIB	PROLIBS	PRSAVOUT
PSA_TOP	PSB_TOP	PSP_TOP	QA_TOP
QP_TOP			

- R - / - S -	RANLIB	RELINK_LIBS	RELINKM
REPORTS_PATH	REPORTS_TMP	REPORTS60_PATH	REPORTS60_TMP
RG_TOP	RHX_TOP	RLA_TOP	RLM_TOP
RM	SCP	SHLIB_PATH	SPCCHANL
SQLLIB	SSP_TOP		

- V - / - W - / - X -	VEA_TOP	VEH_TOP	VNDARPL
VNDARSL	VNDPAYPL	VNDPAYSL	WIP_TOP
WMS_TOP	WPS_TOP	WSH_TOP	WSM_TOP
XDP_TOP	XITOBJ	XITSRW	XLA_TOP
XNC_TOP	XNM_TOP	XNP_TOP	XNS_TOP
XTR_TOP			

Index

A

AD Administration (adadmin)
  log file, 2-3
  main menu, 2-3
  non-interactive mode, 2-16
  running, 2-2
AD Configuration (adutconf.sql), 1-3
  running, 5-7
AD Controller (adctrl), 1-3
  log file, 5-3
  restarting a worker, 5-4
  restarting the manager, 5-6
  running, 5-2
  worker status, 5-3
AD File Identification (adident), 1-4
AD Java Key Generation, 4-32
AD Merge Patch (admrgpch), 1-4
  running, 5-26
AD Rebase (adrebase.exe), 1-5
  running, 5-30
AD Relink (adrelink.sh), 1-4
  running, 5-19
AD Run SQL (adurs), 1-5
  running, 5-29
AD Splicer (adsplice), 1-4
  running, 5-9
AD Timing Report, 1-20
AD Utilities
  before running, 1-6
  how they work, 1-5
  stopping and restarting, 1-8
  when to use, 1-5
adadmin.log, 1-10
adaimgr.log, 3-30
adalnctl.sh, 7-9
adapcctl.sh, 7-11
adctrl, 1-3
adctrl.log, 5-3
addbctx.sql, 6-9
addbxxx.sql, 6-9
addlnctl.sh, 7-8
addmimp, 1-5
adfmcctl.sh, 7-9
adfmsctl.sh, 7-8
adfroctl.sh, 7-8
adident, 1-4
adjkey, 4-32
adlibin.log, 4-15
adlibout.log, 4-15
admrgpch, 1-4, 4-27, 5-26
adncnv, 1-4, 5-13
adodfcmp, 1-4
adpatch, 4-1
adpatch.lgi, 4-15
adpatch.log, 4-15
adrebase.exe, 5-30
adrelink, 1-4
adrelink.log, 4-15
adrepctl.sh, 7-9
adsplice, 5-9
adsysapp.sql, 6-11
adsyxxx.sql, 6-9
adtcfctl.sh, 7-10
adurs, 1-5, 5-29
adutconf.sql, 1-3, 5-7
adworkxx.log, 3-31, 4-15
ADXGNPIN.sql, 4-16, 6-2

ADXGNPNS.sql, 4-16, 6-2
ADXSPPIN.sql, 6-2
ADXSPPNS.sql, 6-2
airfinit.rf9, 3-30
airfmain.rf9, 3-30
airfmod.rf9, 3-30
airfprod.rf9, 3-30
airftask.rf9, 3-30
airftodo.rf9, 3-31
apcmctl.sh, 7-10
APPL_TOP Name, 1-12
APPLCSF, 1-17
applcust.txt, 4-12
Applications Directory (APPL_TOP), 1-9
Applications Environment Name, 1-11
Applications files
  maintaining
    AD Administration, 2-10
APPLLOG, 1-17
APPLOUT, 1-17
applpsum.txt, 4-31
applptch.txt, 4-16, 4-30
appltape.txt, 5-14
APPS schema(s)
  compiling
    through AD Administration, 2-5
    through AutoUpgrade, 3-24
  validating
    through AD Administration, 2-4
APPSORA, 1-7
apps.zip, 4-33
Architecture
  See *Concepts*
AutoPatch, 1-3
  after running, 4-14
  backup files, 4-15
  driver file format, 4-27
  error messages, 4-11
  handling of customizations, 4-12
  log files, 4-15
  options parameter, 4-25
  patch history file, 4-30
  patch summary file, 4-31
  running, 4-7
  running in a multi-tier environment, 4-4
  running in non-interactive mode, 4-21
  running in pre-install mode, 4-20
  running in test mode, 4-19
  starting, 4-8
AutoUpgrade, 1-2
  Database Parameters screen, 3-18
  deleting products, 3-15
  error correction, 3-32
  error messages
    acceptable, 3-34
  how it works, 3-2
  log file, 3-31
  main menu, 3-7
  manager, 3-29
  manager log files, 3-30
  manager restart files, 3-30
  monitoring, 3-29
  number of worker processes, 3-27
  parallel mode, 3-27
  restart files, 3-31
  restarting, 3-5
  running, 3-3
  screens, using, 3-8
  Select Modules screen, 3-17
  Select Products screen, 3-9
  stopping, 3-4
  Tasks screen, 3-22
  upgrade actions, 3-11
  Worker Failed message, 3-29
  worker log files, 3-31
  worker restart files, 3-31
  workers in, 3-29

B

Batch Size, 1-11

C

cmerged.drv, 5-26
Concepts, xi
concurrent managers
  starting, 7-10
conventions
  typographical, xiv

Cost-Based Optimization (CBO), 6-1
parameters, 6-25
_complex_view_merging, 6-27
_fast_full_scan_enabled, 6-29
_like_with_bind_as_equality, 6-28
_optimizer_mode_force, 6-27
_or_expand_nvl_predicate, 6-28
_ordered_nested_loop, 6-29
_push_join_predicate, 6-27
_push_join_union_view, 6-29
_sort_elimination_cost_ratio, 6-28
_sqlexec_progression_cost, 6-30
_table_scan_cost_plus_one, 6-29
_use_column_stats_for_function, 6-28
always_anti_join, 6-29
always_semi_join, 6-30
db_file_multiblock_read_count, 6-27
optimizer_features_enable, 6-26
optimizer_max_permutations, 6-27
optimizer_mode, 6-26
optimizer_percent_parallel, 6-29
optimizer_undo_changes, 6-26
query_rewrite_enabled, 6-29

D

database
analyzing, 6-1
exporting, 6-12
importing, 6-14
migrating, 6-5
preparatory scripts, 6-8
preparing to migrate, 6-7
database administrator
duties of, xi
database initialization parameters (init.ora), 6-20
_shared_pool_reserved_min_alloc, 6-25
_trace_files_public, 6-22
Advanced Queuing (AQ), 6-23
aq_tm_processes, 6-23
archiving, 6-23
audit_trail, 6-21
auditing and security, 6-21
background_dump_dest, 6-22
buffer cache, 6-24
character sets, 6-21
compatible, 6-21
control_files, 6-20
core_dump_dest, 6-22
cursor_space_for_time, 6-25
database block size, 6-20
db_block_buffers, 6-24
db_block_size, 6-20
db_files, 6-24
db_name, 6-20
dml_locks, 6-24
dump, 6-21
enqueue_resources, 6-24
hash area size, 6-25
java_pool_size, 6-25
large_pool_size, 6-25
log writer, 6-24
log_archive_start, 6-23
log_buffer, 6-24
log_checkpoint_interval, 6-24
log_checkpoint_timeout, 6-24
max_dump_file_size, 6-22
max_enabled_roles, 6-21
multi-threaded server (MTS), 6-30
NLS, 6-21
nls_date_format, 6-21
nls_language, 6-21
nls_numeric_characters, 6-21
nls_sort, 6-21
nls_territory, 6-21
open_cursors, 6-24
parallel query, 6-30
parallel_max_servers, 6-31
parallel_min_servers, 6-31
PL/SQL parameters, 6-23
processes, 6-24
processes/sessions, 6-23
rollback_segments, 6-22
row_locking, 6-21
shared pool, 6-25
shared_pool_reserved_size, 6-25
shared_pool_size, 6-25
sort area, 6-24
sort_area_size, 6-25
timed_statistics, 6-22

trace, 6-22
user_dump_dest, 6-22
utl_file_dir, 6-23
database objects
maintaining, 2-3
upgrading, 3-24, 3-26
Database Parameters screen
using, 3-18
Database Tasks
AD Administration, 2-1
DataMerge (addmimp), 1-5
running, 5-27
dbmerged.drv, 5-26
Defaults File
creating
AD Administration, 2-16
AutoPatch, 4-21
dependent products, 3-12
ORACLE user ID, 3-14
digital certificate, 4-32
directories
for Log and Output Files, 1-17
disk space
recovering, 7-5
DUAL table
verifying
through AD Administration, 2-6

E

environment file
AutoUpgrade, 3-27
creating
through AD Administration, 2-11
specifying name, 3-23
verifying, 1-20
error messages
acceptable, 3-34
Export utility, 6-12

F

File Character Set Conversion (adncnv), 1-4
running, 5-13
file system
migrating, 7-1
moving on the same machine, 7-1
moving to another machine, 7-4
File System Tasks
AD Administration, 2-2
files
copying (to destinations), 2-12
generating
forms, 2-13
graphics, 2-15
JAR, 2-15
message, 2-12
report, 2-14
verify for runtime, 2-12
flexfield data
compiling
through AD Administration, 2-6
through AutoUpgrade, 3-24
FND_HISTOGRAM_COLS table, 6-2
FND_PRODUCT_INSTALLATIONS table, 6-16
fndaol.jar, 4-34
fndctx.jar, 4-34
fndewt.jar, 4-34
fndform.jar, 4-34
fndhier.jar, 4-34
fndlist.jar, 4-34
fndtcf.jar, 4-34
force argument (relinking), 5-22
form definition files, 7-6
form files
generating, 2-13
form generation
errors or warnings, 2-14
Forms Metric Client
starting, 7-9
Forms Metrics Server
starting, 7-8
Forms Server listener
starting, 7-8

G

GL_STORAGE_PARAMETERS table, 6-18
gmerged.drv, 5-26
grants

create missing, 6-16
recreating
through ODF Comparison, 5-16
grants and synonyms
recreating
through AD Administration, 2-5
graphics files
generating, 2-15

H

HTTP server
database objects, 6-10
starting, 7-11

I

initialization parameters
checking through AutoUpgrade, 3-23, 3-28
database, 6-20
init.ora parameters
fixed SGA, 6-23
general, 6-20
Installing Oracle Applications
See *Installing Oracle Applications manual*
interMedia
setup script, 6-9

J

JAR files
generating, 2-15
regenerating, 4-34
signing, 4-34
Java Release Infrastructure (JRI), 4-35

L

language installation
specifying language, 3-5
library files, 7-6
License Manager (LicenseMgr), 1-5
licensing a new language, 5-34
licensing country specific functionalities, 5-33
licensing product families, 5-36
licensing products, 5-32
running, 5-31
localizations
licensing screen, 3-17
log file, 1-10
AD Administration, 2-3
AD Controller (adctrl), 5-3
AutoPatch, 4-9
AutoUpgrade, 3-4
manager file, 3-30
worker files, 3-31
ODF Comparison (adodfcmp), 5-17

M

maintenance pack, 1-3
manager
log files, 3-30
messages, 3-29
restart files, 3-30
mcs, 7-7
message files
generating, 2-12, 3-24
mini-pack, 1-3
multi-lingual tables
maintaining, 2-6
Multi-Org
converting a standard database to, 2-9
Multiple Reporting Currencies (MRC)
converting to, 2-8
maintaining, 2-6, 4-16
multiple sets of books (MSOBA)
dependencies, 3-15
upgrading, 3-12

N

Net8 listener
naming, 7-9
starting, 7-7

O

object description file (ODF), 5-14
objects

create missing, 6-16
ODF Comparison (adodfcmp), 1-4
parameters, 5-15, 5-28
Oracle Applications passwords
changing, 6-4
Oracle Bug Database
patch numbers in, 4-5
Oracle Consulting Services, xv
Oracle Support Services, xv
ORACLE user ID
changing password, 3-20, 6-4
dependent product, changing, 3-14
product, 3-10, 3-20

P

Packages
pinning, 6-2
passwords
changing, 6-4
patch
AD Merge Patch utility, 4-27
driver files, 4-5
format, 4-5
integrated, 4-7
readme.txt file, 4-5
patch directory, 4-9
patch driver file, 4-9
patch set, 1-3
patch subdirectory, 4-15
pre-install mode
running AutoPatch in, 4-20
prerequisite fixes
applying, 4-7
product abbreviations, 6-32
product files
unloading, 7-4

R

related documents
list of, xvi
release update, 1-3
relinking
force argument, 5-22
programs, 2-11
target file, 5-23
report files
generating, 2-14
Report Review Agent
starting, 7-9
Reports Server
starting, 7-9
reserved words, 7-11
rollback segments, 6-31

S

Schema Statistics
gathering, 6-2
Select Modules screen
using, 3-17
Select Products screen
using, 3-9
Sequences
pinning, 6-2
server processes
starting, 7-7
sizing factor, 6-32
setting during upgrade, 3-21
strip, 7-7
synonyms
recreating
through AD Administration, 2-5
through ODF Comparison, 5-16
system administrator
duties of, xi

T

tablespaces
requirements for products, 6-31
specifying during upgrade, 3-21
TCF SocketServer
starting, 7-10
technical specialist
duties of, xii
temporary files
setting directory variables, 1-18
topfile.txt, 7-2

translations, 4-32

U

upgrade
running, 3-27
Upgrading Oracle Applications
See *Upgrading Oracle Applications manual*

V

Verify Apps (advrfapp.sql), 1-5
running, 2-5

W

Web Server
naming, 1-19
worker
failed message, 3-29
log files, 3-31
messages, 3-29
processes, 3-27
restart files, 3-31

Z

Zipped Resource Unit (ZRU), 4-32