Oracle® Applications

Maintenance Utilities

Release 11*i* (11.5.9)

Part No. B10640-01

June 2003

ORACLE®

Oracle Applications, Maintenance Utilities, Release 11*i* (11.5.9)

Part No. B10640-01

Copyright © 2000, 2003 Oracle Corporation. All rights reserved.

Primary Authors: Carole Eubanks, Michael Fiore, Yun Shaw

Contributors: Gary Burch, Subash Chadalavada, Venkat Dameria, Suchi Dorbala, Miles Fender, Henriette Fux, William Greene, Venkat Hariharan My-Xuan Ho, Matthew Lobaugh, Rick Lotero, Kent Noble, Andrew Rist, Prashant Saddi, Roger Sanders, Alice Sun

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

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

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

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

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Oracle is a registered trademark, and ConText, JInitiator, Oracle Discoverer, Oracle*MetaLink*, Oracle Store, Oracle8, Oracle8*i*, Oracle9*i*, Oracle9*i*AS Discoverer, PL/SQL, SQL*Net, and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

6 Reports

Index

Send Us Your Comments

Oracle Applications, Maintenance Utilities, Release 11*i* (11.5.9)

Part No. B10640-01

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

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

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

Preface

Oracle Applications Maintenance Utilities and *Oracle Applications Maintenance Procedures* make up the *Maintaining Oracle Applications Documentation Set*. This manual provides information about the tools for installing, updating, patching, and upgrading Oracle Applications products. This manual is a companion to *Oracle Applications Maintenance Procedures*, which describes how to maintain the Oracle Applications file system and the database. Much of the information in this manual was contained in the *AD Utilities Guide* in earlier releases.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Structure

This book contains the following chapters:

- Chapter 1 provides general information about the utilities and instructions for setting the environment. Two important utilities, Rapid Install and the AD Controller, are discussed in greater detail here.
- Chapter 2 describes the utilities you use to configure your system. Configuration includes which Oracle Applications products are registered, installation of the technology stack that supports Oracle Applications, adding new products after installation, and conversion from one character set to another.
- Chapter 3 describes the utilities you use to maintain your system. The primary maintenance utility is AD Administration.
- Chapter 4 describes patching your system. This chapter explains the structure of a patch, merging two or more patches into one, and the principal patching utility: AutoPatch.
- Chapter 5 describes the AutoUpgrade utility, which you use when upgrading from an earlier release. Specific steps for completing an upgrade are in the *Upgrading Oracle Applications* manual.
- Chapter 6 describes reporting utilities. The patch history database, new to Release 11*i*, provides information about which patches have been applied on your system. Reporting utilities can tell you which patches have been applied, which patches should be applied, and other system information not related to patching.

Related Documents

All Release 11*i* documentation is included on the *Oracle Applications Document Library CD*, which is supplied in the Release 11*i* Update CD Pack. You can download some soft-copy documentation from Oracle Docs Online at

http://otn.oracle.com/documentation. You can also purchase hard-copy documentation from the Oracle Store at http://oraclestore.oracle.com.

If you are looking for...	See these documents...
Additional information	*Oracle Applications Maintenance Procedures* *Upgrading Oracle Applications* *Installing Oracle Applications* *Oracle Applications Concepts* *Oracle Applications Installation Update Notes** *Oracle Applications Release Notes** *Oracle Applications NLS Release Notes** *Oracle Applications System Administrator's Guide* *Oracle Self-Service Web Applications Implementation Manual* *Oracle Workflow Administrator's Guide* *Oracle Workflow Developer's Guide* *Oracle Application Object Library/Workflow Technical Reference Manual*
Application-specific features	Oracle Applications user's guides Oracle Applications implementation manuals *Multiple Organizations in Oracle Applications* *Multiple Reporting Currencies in Oracle Applications* *Oracle Applications CRM Supplemental Installation Guide*
Information about custom development	*Oracle Applications User Interface Standards for Forms-based Products* *Oracle Applications Developers' Guide*
Database information	*Oracle Concepts* *Oracle Backup and Recovery Guide* *Oracle Reference* *Oracle Designing and Tuning for Performance* *Oracle National Language Support Guide*
Information on new features in this release	Oracle*MetaLink* document 210326.1*

*Available only on Oracle*MetaLink*

Update or patch readme files may contain information about new documentation.

> **Note:** Documentation associated with this release was current as of the time it was released. Oracle*MetaLink* contains the most up-to-date information.

Training and Support

Oracle offers a complete set of training courses and multi-level support services.

Training

You can attend training courses at any Oracle Education Center, arrange for trainers to teach at your facility, or use Oracle Learning Network (OLN) — Oracle University's online education utility. Oracle training professionals can also develop custom courses using your organization structure, terminology, and data as examples.

Support

The Oracle support team includes your Technical Representative and Account Manager. It also includes Oracle consultants and support specialists who have expertise in your business area, and in managing an Oracle RDBMS server and your hardware and software environment.

Oracle*MetaLink* is a self-service, web-based support connection, which is maintained by Oracle Support Services 24 hours a day, 7 days a week. Use it to obtain information and advice from technical libraries and forums, download patches, look at bug details, and create or update TARs. Register at http://metalink.oracle.com, and check for updates and information before you install or upgrade your Oracle Applications. The *Start Here* CD also contains links to the various resources on Oracle*MetaLink*.

Conventions

The following conventions are used in this book:

Convention	Meaning
UNIX: **Windows:**	Indicates platform-specific information. This guide contains information for both UNIX and Windows platforms. All instructions for UNIX platforms also apply to Linux platforms, unless otherwise noted.
$ or C:\>	Represents the platform-specific command prompt. Your prompt may differ.
`Monospace text`	Represents command line text. Type this text exactly as shown.
< >	Text enclosed in angle brackets represents a variable. Substitute a value for the variable text. Do not type the brackets.
[ ]	Encloses optional items or indicate a function key. Do not type the brackets.
|	Represents an *or* option among several options. You must enter only one of the options. Do not type the vertical bar.

Convention	Meaning
\	In examples of commands you type online, a backslash at the end of a line signifies that you must type the entire command on one line. *Do not type the backslash*.
Special notes	Additional Information, Attention, Note, and Warning boxes alert you to particular information within the body of the book.

1

Getting Started

The Oracle Applications utilities are a set of tools to install, upgrade, patch, and maintain Oracle Applications products. This chapter contains the following topics:

What Are the Utilities?

Many of the utilities used for maintenance are called "AD utilities," as "AD" is an abbreviation for Applications DBA. The AD utilities have similar interfaces, operation, input and report format. Much of the discussion about how these utilities work is common to all the AD utilities.

In addition to the AD utilities, some maintenance tasks are run from the Oracle Applications Manager (OAM). OAM is a web-based application that provides a subset of System Administrator functions and many other features. See the *Oracle Applications System Administrator's Guide* for more information on the Oracle Applications Manager.

This book describes the interfaces for all utilities, both AD and others, in depth.

Rapid Install

Rapid Install installs Oracle Applications products at the latest available Maintenance Pack level. It provides a wizard for entering parameters specific to your installation or upgrade. Rapid Install stores the parameters you choose in a configuration file, and then uses that file to perform the installation or upgrade. After you define a configuration for your Oracle Applications system, Rapid Install installs the required technology stack and creates the Oracle Applications file system. It then sets up the database listeners, web listener, web server, forms server, and reports server.

Additional Information: See *Installing Oracle Applications*.

Configuration and Environment Files

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

Note: The <CONTEXT_NAME> variable defaults to <SID>_<hostname>. You may set it to <SID> or customize it to some other environment-specific name.

File name	Location	Description
adconfig.txt	APPL_TOP/admin	Contains environment information used by all AD utilities. *Do not* update this file manually.
adalldefaults.txt	APPL_TOP/admin	A template defaults file that contains entries for all defaults-enabled prompts in AD utilities. Can be copied to $APPL_TOP/admin/<SID>/<new_name>.txt and edited.
def.txt	APPL_TOP/admin/<SID>	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.
applprod.txt	APPL_TOP/admin	The AD utilities product description file, which is used to identify all products and product dependencies.
applterr.txt	APPL_TOP/admin	The AD utilities territory description file. It contains information on all supported territories and localizations.

File name	Location	Description
applora.txt	APPL_TOP/admin	Contains information about required init.ora parameters for runtime.
applorau.txt	APPL_TOP/admin	Contains information about required init.ora parameters for install and upgrade.
APPS<CONTEXT_NAME>.env (UNIX) APPS<CONTEXT_NAME>.cmd (Windows)	APPL_TOP	Calls the appropriate <CONTEXT_NAME>.env (UNIX) or <CONTEXT_NAME>.cmd (Windows) file to set up the Oracle Applications (APPL_TOP) and the Applications technology stack (8.0.6). (Called APPSORA in earlier releases.)
<CONTEXT_NAME>.env (UNIX) <CONTEXT_NAME>.cmd (Windows)	APPL_TOP	The APPL_TOP environment file used to configure the environment to run Applications. It is created by AutoConfig.
<CONTEXT_NAME>.env (UNIX) <CONTEXT_NAME>.cmd (Windows)	RDBMS	Used to perform maintenance operations on the database.
<CONTEXT_NAME>.env (UNIX) <CONTEXT_NAME>.cmd (Windows)	iAS	Used to perform maintenance operations on the iAS ORACLE_HOME.
<CONTEXT_NAME>.env (UNIX) <CONTEXT_NAME>.cmd (Windows)	8.0.6	Used to perform maintenance operations on the 8.0.6 ORACLE_HOME.
adovars.env (UNIX) adovars.cmd (Windows)	APPL_TOP	Called by <CONTEXT_NAME>.env (UNIX) or <CONTEXT_NAME>.cmd (Windows) and is used to set environment variables for Java and HTML.
fndenv.env		Sets additional environment variables used by Oracle Application Object Library. The default values should be applicable for all customers.

Setting the Environment

Before you start any AD utility, you must first set the Applications environment:

> **Note:** See the *Oracle Applications Installation Update Notes* for any additional steps.

1. Log in as applmgr (Applications file system owner).
2. Run the environment or command file for the current APPL_TOP and database.

UNIX:

The environment file is typically APPS<CONTEXT_NAME>.env, and is located under APPL_TOP. From a Bourne or Korn shell, type the following:

```
$ . APPS<CONTEXT_NAME>.env
```

Windows:

Run %APPL_TOP%\envshell.cmd using either Windows Explorer or the Run command from the Start menu. This creates a Command Prompt window that contains the required environment settings for Oracle Applications. Run all subsequent commands in this Command Prompt window.

3. If you have made any changes to the environment, check that it is correctly set by typing the following commands:

   UNIX:

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

   Windows:

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

   ORACLE_HOME must be set to the proper database directory, and TWO_TASK or LOCAL must identify the correct database. On Windows, APPL_CONFIG must be set to <CONTEXT_NAME>.

4. Ensure that there is sufficient temporary disk space.

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

5. Shut down all concurrent managers if you plan to relink Oracle Applications product files or modify Oracle Applications database objects.

   Additional Information: See Administer Concurrent Managers in *Oracle Applications System Administrator's Guide.*

How the AD Utilities Work

There are three primary AD utilities: AD Administration, AutoPatch, and AutoUpgrade. Several other maintenance utilities were developed for specific Applications maintenance tasks. As one AD utility runs, it may automatically call one of the other utilities. However, you can also run a utility directly. The following is a brief description of each AD utility.

AD Utility	Executable Name	Action
AD Administration	adadmin	Performs maintenance tasks on Oracle Applications.
AutoPatch	adpatch	Applies patches, and adds new languages and products.
AutoUpgrade	adaimgr	Upgrades to the latest version of Oracle Applications.
AD Controller	adctrl	Manages parallel workers in AutoUpgrade, AD Administration, and AutoPatch.
AD Configuration	adutconf.sql	Reports standard information about the installed configuration of Oracle Applications.
AD File Identification	adident	Identifies the version and translation level of an Oracle Applications file.
AD Splicer	adsplice	Registers off-cycle products.
File Character Set Converter	adncnv	Converts a file from one character set to another.
AD Relink	adrelink	Relinks Oracle Applications executable programs with the Oracle server product libraries.
AD Merge Patch	admrgpch	Merges multiple patches into a single, integrated patch.
AutoConfig	adautocfg	Helps manage the various system configuration files.
AD Timing Report	adtimrpt.sql	Provides timing summary reports for jobs run by parallel workers.
License Manager	adlicmgr.sh	Licenses products, country-specific functionalities, or languages.

The AD utilities perform a variety of tasks, including generating files, upgrading your system, applying and merging patches, installing off-cycle products, and many others. The chapters in this book describe the types of tasks performed in detail.

Running the AD Utilities

To run an AD utility, type the utility's start command (such as *adpatch, adadmin* or *adaimgr*) and answer the prompts.

You can exit most utilities by entering *abort* at any prompt. Then, you can restart by typing the start command for that utility. When you restart, you can enter a new log file name or specify the log file from the previous session. 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 continued session as it generates them.

You can then do one of the following:

- Continue Session (the default)

  The utility restarts at the point where your last session stopped.

- 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.

> **Attention:** We recommend that you choose Continue Session. Some actions from the first session may be voided or duplicated by the new session.

Command Line Arguments

AD utilities accept arguments on the start command line that modify the actions performed by the utility. Arguments may be either options or flags.

The AD command line arguments listed in this chapter may be required for normal operation of an AD utility. Not all AD command line arguments are documented here as some should only be used when instructed by Oracle Support Services.

Command line arguments are in the "token=value" format. Enter AD command line arguments in lowercase. (The "token" portion is converted to lower-case, but the "value" is not.) In some cases, "value" is a comma-separated list. Note that AD command line arguments cannot contain embedded whitespace characters.

Examples of valid command line arguments:

```
adpatch options=nocopyportion,nogenerateportion
adpatch printdebug=y
```

Examples of invalid command line arguments:

```
adpatch options=nocopyportion, nogenerateportion

adpatch OPTIONS=NOCOPYPORTION,NOGENERATEPORTION
```

The first invalid syntax here contains a whitespace. In the second, the token ("OPTIONS") will be converted to lowercase, but the values (NOCOPYPORTION,NOGENERATION) are not recognized by the utility.

Example of using multiple command line arguments:

```
adpatch printdebug=y options=validate flags=hidepw
```

The following options can be used by the AD Utilities:

printdebug	
Used by	All AD Utilities
Purpose	Tells AD programs to display extra debugging information. In some cases, the amount of extra debugging information is substantial.
Values	y or n
Default	n
Example	adpatch printdebug=y

interactive	
Used by	AD Administration, AutoPatch
Purpose	Tells AD utilities whether to run in interactive mode or non-interactive mode.
Values	y or n
Default	y, meaning that the utility runs in interactive mode
Example	adpatch interactive=n

restart	
Used by	AD Administration, AutoPatch
Purpose	Tells AD utilities to restart an existing session in non-interactive mode. Only valid when interactive=n is also specified.
Values	y or n
Default	n, meaning that the utility run in non-interactive mode will expect to run a completely new session.
Example	adpatch interactive=n restart=y

logfile	
Used by	All AD Utilities
Purpose	Tells AD utilities what log file to use. Normally used in non-interactive mode.
Values	A file name (not a fully-qualified path name)
Default	None, meaning that the utility will prompt for the log file name.
Example	adpatch logfile=test.log

defaultsfile	
Used by	AD Administration, AutoPatch
Purpose	Specifies the defaults file which stores answers to interactive AD utility questions. Normally used in non-interactive mode.
Values	A fully-qualified filename. Must be under the $APPL_TOP/admin/<SID> directory.
Default	None, meaning that no defaults file is used.
Example	adpatch defaultsfile=/d1/apps/prodappl/admin/prod1/prod_def.txt

workers	
Used by	AD Administration, AutoPatch
Purpose	Specifies the number of workers to run. Normally used in non-interactive mode.

workers	
Values	1 to 99, inclusive
Default	None, meaning that the program will prompt for the number of workers to run.
Example	adpatch workers=6

flags	
Used by	All AD Utilities
Purpose	Generic flags passed to AD utilities. See the information that follows for more information on flags that are used.
Values	A comma-separated list of flags.
Default	None, meaning that no flags have been passed.
Example	adpatch flags=hidepw

The flags option may be followed by a list of flags. The valid flags are:

AD Utilities Flags	
hidepw	
Purpose	Directs the utilities to include or omit the 'HIDEPW:' comments in AD utility log files.
Default	nohidepw
Comments	Starting in AD Mini-pack H, lines in an AD utility log file containing passwords can be masked to not display the passwords. When nohidepw is specified, each line containing masked passwords is followed by a corresponding line prefixed with HIDEPW. The HIDEPW line shows original line with passwords. When hidepw is specified, the HIDEPW: line is not displayed. This flag defaults to include passwords in the AD program log file to aid debugging. Passwords can be stripped by other means, such as using the UNIX command grep -v 'HIDEPW:' file > file_no_pws. When security is more important, specify hidepw to prevent passwords from appearing in the AD utility log file.

AD Utilities Flags	
logging	
Purpose	Tells the AD Utility whether to create indexes using the logging or nologging mode.
Default	logging in AutoPatch; nologging in AD Administration.
Comments	Use of NOLOGGING when creating indexes may increase performance. However it also makes database media recovery incomplete and does not work with standby databases. LOGGING is the default in adpatch to support database media recovery and standby databases. We do not recommend using flags=nologging for production systems unless you make a complete backup both before and after running adpatch. NOLOGGING is the default in adaimgr because running adaimgr with an enabled standby database or in archive log mode has no benefits. If you make a complete database backup before and after running adaimgr, it is safe to take advantage of the extra performance gain provided by NOLOGGING. flags=nologging affects indexes created through ODF only, not SQL scripts.
trace	
Purpose	Tells the AD Utility whether to log all database operations to a trace file.
Default	notrace
Comments	RDBMS trace files created while running an AD utility may aid debugging. The flags=trace option creates multiple trace files for the AD utility and the AD workers. There is a new trace file each time the utility connects to the database. Note that flags=trace only traces database operations internal to the AD utility itself. Database operations in SQL scripts or external programs run by the AD utility are not recorded by flags=trace.

AD Utilities Features

AutoPatch, AutoUpgrade, and AD Administration, and some of the other utilities, employ certain common features. For example, they perform processing tasks in parallel, create log files, and make extensive use of prompts to gather the values for installation-specific information.

Attention: If a utility fails and you choose to continue past the error, you must correct the problem manually. If you do not correct the problem, the utility may produce incorrect results or not be able to continue. If you have difficulty resolving an error, contact Oracle Support Services.

AD Feature Versions

AD now has several major features that require the files on the file system and the tables in the database to match. To ensure this, each major feature now has both a file system and database version, and AD only enables the feature if the file system and database versions match.

AD Prompts

Many of the AD utilities prompt for the information needed to complete a task. Prompts typically include a description of the information needed, and may include a default answer (in square brackets). For example:

```
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] :
```

Press Return to accept the default value, or type a new value after the colon and press Return.

Attention: Read the prompts carefully to make sure you supply the correct information.

Parallel Processing

AutoPatch, AutoUpgrade, and AD Administration can perform some tasks using parallel processing. Parallel processing provides better throughput, better use of available resources, and overall reduction in the time it takes to complete tasks.

Parallel processing is typically used to:

- run database driver tasks, such as SQL scripts.
- generate various kinds files, such as forms, reports, messages, and graphics files.

Parallel processing is controlled by job *managers*, which, in turn, direct the actions of *worker* processes. The workers complete processing tasks assigned to them by the manager. Utilities that use parallel processing determine the list of tasks to be performed and prioritize them for execution. The utility also prompts for the number of workers you want to perform the tasks. For example, when AutoPatch is executing a database driver, it creates a list of database tasks and prompts for the number of workers that should run concurrently to execute these tasks.

AutoUpgrade, AutoPatch, and AD Administration are the manager processes in this model. The worker processes are instances of the adworker program. The adworker program can only be called my the manager processes and cannot be run stand-alone.

The manager creates the FND_INSTALL_PROCESSES table, assigns each worker a unique ID, and inserts a row for each worker. This table serves as a staging area for the job information, and as a way for the manager and the worker to communicate. Communication is accomplished using two columns: CONTROL_CODE and STATUS.

Managers

The manager updates the table with a subset of the list of jobs, one job per worker. For example, if there are five workers, then the table holds five jobs (even though there may be 100 or more jobs involved in the complete action). The manager starts the workers and uses the CONTROL_CODE and STATUS columns to assign tasks. It polls these two columns continuously, looking for updates from the workers. As a worker finishes its assignment, the manager updates each row with the next task in the list, and leaves another message for the worker.

Once all jobs are complete, the manager tells the workers to shut down, and then drops the FND_INSTALL_PROCESSES table (after it is sure all workers have actually shut down).

Workers

Each worker updates the STATUS column, giving the manager a report on its progress. As the jobs are completed, the manager updates the table with the next job in the queue, and updates the CONTROL_CODE and STATUS columns telling the worker to start processing. If there is a failure, the worker reports a failed status.

For certain tasks, some worker processes spawn other *child processes* that do the actual work. The spawned child process returns a status code to the worker that spawned it. The worker interprets the code to determine if the job has been completed successfully. Examples of child processes are SQL*Plus and FNDLOAD.

Deferred Jobs

The first time a job fails, the manager automatically defers it to the end of the current phase and assigns a new job to the worker. If the deferred job fails the second time it is run, the manager defers it again only if the total runtime of the job is less than ten minutes. If the deferred job fails a third time (or if the job's total runtime is not less than ten minutes the second time it is run) the job stays at failed status and the worker waits. At this point, you must address the cause of the failure, and then restart the job using AD Controller.

The deferred job feature uses the AD_DEFERRED_JOBS table. This table is created when the FND_INSTALL_PROCESSES table is created, and is dropped when the FND_INSTALL_PROCESSES table is dropped.

Controlling Managers and Workers

Use AD Controller to determine the status of workers and to control their operation. See AD Controller (adctrl) in this chapter for more details.

Log and Restart Files

All AD utilities record their processing actions and any errors in log files. Many utilities prompt you for the name of the log file that will record the processing session. The default file name is <utility name>.log. For example, for AD Administration, the default log file is adadmin.log, and for AutoUpgrade, the log file is adaimgr.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 three primary AD utilities place the log file in $APPL_TOP/admin/<SID>/log, where <SID> is the value of your ORACLE_SID or TWO_TASK variable (UNIX), or in %APPL_TOP%\admin\<SID>\log, where <SID> is the value of the LOCAL variable (Windows).

Other utilities may not prompt you for a log file name, and they may write their log file in the directory from which the utility was run. The discussion of each utility in this manual includes information on where the log file is written.

Worker log files

In addition to the information recorded in the <utility name>.log file, utilities that process jobs in parallel mode write details about any errors to worker log files. Review these adwork<number>.log files (adwork01.log, adwork02.log...) for information about any errors. These files are written to the $APPL_TOP/admin/<SID>/log directory, where <SID> is the value of the ORACLE_SID or TWO_TASK variable (UNIX), or in %APPL_TOP%\admin\<SID>\log, where <SID> is the value of ORACLE_SID or LOCAL (Windows).

Concurrent requests run by AutoPatch, AutoUpgrade, and AD Administration create their own log files.

> **Additional Information:** See Log and Output Filenames in Oracle *Applications System Administrator's Guide*.

Restart files

Restart files are used by many AD utilities to continue processing at the point where it stopped. Restart files reside in $APPL_TOP/admin/ <SID>/restart or in %APPL_TOP%\admin\<SID>\restart on Windows. The restart files contain information about what processing has already been completed, so that the utility can pick up where it left off.

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

By default, AD utilities delete their restart files when processing completes, but leave backup versions with the extensions .bak, .bk2, or .bk3.

Each worker may also have a restart file called adwork*xx*.rf9. These files are stored in $APPL_TOP/admin/<SID>/restart or in %APPL_TOP%\admin\<SID> \restart on Windows. The worker creates the restart file when the manager assigns it a job, and deletes the restart file when it finishes the job.

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

Additional Information: The Troubleshooting chapter in *Oracle Applications Maintenance Procedures* discusses various error situations when running a utility and how to resolve them.

Manager and Worker Log Messages

AutoPatch, AutoUpgrade, and AD Administration act as managers that coordinate a number of workers and assign them jobs to run during the upgrade. When these utilities are running, you see messages like the following on the screen:

```
Starting phase 0 (A0): first

There are now 2 jobs remaining (current phase=A0):
    0 running, 2 ready to run and 0 waiting.

 Assigned: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
 Assigned: file ad_wait2sec.sql on worker  2 for product ad  username APPLSYS.
   FAILED: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
 Deferred: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
           (Defer number 1 for this job)
 Assigned: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
Completed: file ad_wait2sec.sql on worker  2 for product ad  username APPLSYS.
   FAILED: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
 Deferred: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
           (Defer number 2 for this job)
 Assigned: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
   FAILED: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.

ATTENTION: All workers either have failed or are waiting:

           FAILED: file ad_wait1sec.sql on worker  1.

ATTENTION: Please fix the above failed worker(s) so the manager can continue.

Restarted: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
Completed: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
```

These messages indicate what each worker is doing. The example shows two workers running SQL scripts for one product, which is identified by its abbreviation (ad).

Worker Failed Message

The manager displays a message if a worker cannot complete its job. For example:

```
FAILED: file ad_wait1sec.sql on worker  1 for product ad  username APPLSYS.
```

In this example, worker 1 failed on the file ad_wait1sec.sql for Applications DBA (ad). If the job can be deferred, the manager displays a message that the job is deferred and assigns another job to the worker. If the job cannot be deferred, 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 that are not dependent on the failed job to the other workers.

The AutoPatch, AutoUpgrade, or AD Administration session is not complete until all jobs run successfully. When a job fails, determine the cause of failure, fix the problem, and restart the job.

> **Additional Information:** See Troubleshooting in the *Oracle Applications Maintenance Procedures*.

AD Controller (adctrl)

AD Controller is a general maintenance utility you use to determine the status of AutoUpgrade, AD Administration, or AutoPatch workers and to control worker operation.

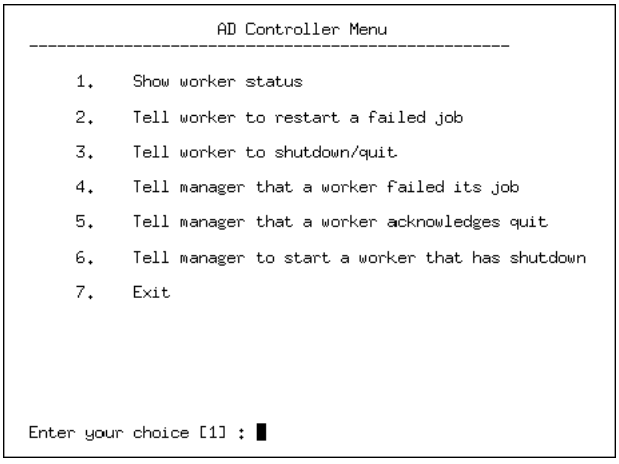
> **Note:** Run this utility in its own window, not in the same window as AutoUpgrade, AD Administration, or AutoPatch.

To start AD Controller:

Log in as applmgr and set the environment as described in Setting the Environment in Chapter 1. Start AD Controller with the *adctrl* command. It prompts you to:

- Confirm the value of APPL_TOP.
- Specify an AD Controller log file (the default is adctrl.log).
- Supply the Oracle Application Object Library username and password.

AD Controller displays the main menu.



Type a number to select an option. Press [Return] to return to the menu.

Additional Information: See Troubleshooting in *Oracle Applications Maintenance Procedures*.

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 Control Worker is the worker number; Code is the last instruction from the manager to this worker; Context is the general action the manager is executing; and Filename is the file the worker is running (if any). 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.

Note that the AD Controller log file is written in the current working directory.

2

Configuration

This chapter contains information about the following utilities:

- AutoConfig
- Context Editor
- License Manager
- AD Splicer (adsplice)
- File Character Set Converter (adncnv)

AutoConfig

AutoConfig is a tool for configuring an Applications instance using standard utilities. It manages your Release 11*i* configuration files, such as httpd.conf, jserv.properties, and appsweb.cfg.

All the information required for configuring an Applications instance is collected in a repository, called the Applications Context. AutoConfig uses information from the Applications Context file to generate all configuration files and update database profiles.

Rapid Install installs an environment compatible with AutoConfig. There is an Applications Context file for both the applications and the database tier. The applications tier Context file is an xml file in <APPL_TOP>/admin. In Release 11.5.9, this Applications Context file is named <CONTEXT_NAME>.xml. If you migrated to AutoConfig in a release earlier than Release 11.5.9, this Applications Context file is named <SID>.xml. The Applications Context file for the database tier is in <RDBMS ORACLE_HOME>/appsutil and is named <CONTEXT_NAME>.xml.

Note: The <CONTEXT_NAME> variable defaults to <SID>_<hostname>. You may set it to <SID> or customize it to some other environment-specific name.

There are two ways your configuration can be changed: run the Context Editor or apply a patch that contains a new version of a file controlled by AutoConfig. When you change the Applications Context file with the Context Editor, or when a patch changes any associated template, you can run AutoConfig to propagate the new values to the configuration files in both the APPL_TOP technology stack and in the ORACLE_HOME technology stack.

AutoConfig components include:

Component	Definition
Applications Context	An XML repository used to generate configuration information.
AutoConfig file templates	Files that include variable name tags that will be replaced with instance-specific information from the Applications Context when the files are copied to their destination (instantiated).
AutoConfig file drivers	File driver, similar to standard Applications file drivers, that list the AutoConfig file templates and their destination locations. There is one file driver per product.
AutoConfig scripts	A set of scripts that provide a simplified interface to the AutoConfig APIs.

Additional Information: Context Editor in this chapter and Managing Configuration Parameters in *Oracle Applications Maintenance Procedures*.

Running AutoConfig

AutoConfig generates the new configuration files for the APPL_TOP and the Oracle Homes in the associated techstack. AutoConfig uses the parameters stored in the Applications Context file and system configuration templates to create new process control scripts and update system profiles.

1. Log on as applmgr and set the environment.
2. Stop all server processes.

Additional Information: See Starting or Stopping All Server Processes in *Oracle Applications Maintenance Procedures*.

3. Start AutoConfig on the application tier.

   UNIX:

   AutoConfig prompts for the APPS password.

   ```
   $ <COMMON_TOP>/admin/scripts/<CONTEXT_NAME>/adautocfg.sh
   ```

   Windows:

   AutoConfig does not prompt for the APPS password, however, you must supply it in the command.

   ```
   C:\> <COMMON_TOP>\admin\scripts\<CONTEXT_NAME>\adautocfg.cmd \
        <APPS password>
   ```

   Or, start AutoConfig on the database tier.

   UNIX:

   AutoConfig prompts for the APPS password.

   ```
   $ <RDBMS ORACLE_HOME>/appsutil/scripts/<CONTEXT_NAME>/adautocfg.sh
   ```

   Windows:

   AutoConfig does not prompt for the APPS password, however, you must supply it in the command.

   ```
   C:\> <RDBMS ORACLE_HOME>\appsutil\scripts\<CONTEXT_NAME>\
   adautocfg.cmd <APPS password>
   ```

   AutoConfig generates new configuration files.

4. Start all server processes.

   Use the newly generated server process script to start all server processes.

AutoConfig Log Files and Rollback

After running AutoConfig, review the log files for any errors or warnings. There is only one log file per AutoConfig session, and it contains detailed information for every action performed.

Each execution of AutoConfig creates a rollback script in case you need to revert to the previous configuration settings. The script and all backup configuration files from each AutoConfig session are stored in the following directories:

Tier	Directory
Applications	<APPL_TOP>/admin/<CONTEXT_NAME>/out/<MMDDhhmm>
Database	<RDBMS ORACLE_HOME>/appsutil/out/<MMDDhhmm>

Navigate to the appropriate directory and use these commands to rollback the AutoConfig session:

UNIX

```
restore.sh
```

Windows

```
restore.cmd
```

Context Editor

The Context Editor is a wizard that modifies the Applications Context based on the changes you make on GUI interface screens. Run the Context Editor from OAM to perform configuration updates to your Applications Context.

Additional Information: Managing Configuration Parameters in *Oracle Applications Maintenance Procedures*.

License Manager

At various times throughout the life cycle of an Oracle Applications release, you may decide to add other products, country-specific functionalities, or languages. If you do, you must register the new products and functionalities. This registration process is necessary so that other utilities recognize that these new components are activated.

You use License Manager, an Oracle Applications Manager (OAM) utility, to register new Applications products, country-specific functionalities, and languages.

Note: License Manager does not set up license agreements or determine pricing. It only registers the products you have licensed or the country-specific functionalities and languages you have added. Use the Oracle Store to obtain new software or contact your Oracle sales representative to set up licensing agreements.

Additional Information: See Using License Manager in *Oracle Applications Maintenance Procedures*.

AD Splicer (adsplice)

Products introduced after a given release — not on the base Oracle Applications CD for that release — are called *off-cycle products*. AutoPatch installs groups of off-cycle products contained in Maintenance Packs. Off-cycle products not yet included in a Maintenance Pack are installed using the AD Splicer utility. AD Splicer modifies the APPL_TOP and database so that AutoPatch and AD Administration recognize the off-cycle product as a valid Oracle Applications product for the given release. AutoUpgrade ignores products for an existing release that have been added by AD Splicer.

Additional Information: See License Off-cycle Products in *Oracle Applications Maintenance Procedures.*

Before starting AD Splicer, log in as applmgr and complete the steps in the Setting the Environment section of Chapter 1. To run AD Splicer, navigate to the admin subdirectory under APPL_TOP and type *adsplice* at the command line.

Attention: Do not use AD Splicer to add custom products to Oracle Applications.

Splicing New Products

Patches that contain off-cycle products contain the AD Splicer files you need to splice in the new product, including control files and a readme file that describes how to install the new product. There are two kinds of AD Splicer control files: *product definition* and *product configuration*.

Product Definition Files

There are two Product Definition Files per spliced product. *Do not* edit these files.

File	Contents
<prod>prod.txt	Language-independent information for product <prod>
<prod>terr.txt	Language-dependent information for product <prod>

Product Configuration File

There is one Product Configuration File (newprods.txt) for each group of related spliced products. You *must* edit this file before you copy AD Splicer control files to any APPL_TOP. Each spliced product in newprods.txt has an entry similar to the following:

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

The newprods.txt file must contain all the entries shown in the example for each spliced product, and the entries must be in the exact order shown. Here is the basic information about each product entry line in newprods.txt.

Entry	Description
product=	Identifies the product being spliced. Cannot be modified. The product abbreviation <prod> is also used to name the <prod>prod.txt and <prod>terr.txt control files for this product. Most internal references use <prod>.
base_product_top=	Identifies the base directory that contains the product's files. The default value, *APPL_TOP*, means the product's files are written in the directory the APPL_TOP environment is set to. If you want the product files written to another directory, provide the full pathname here.
oracle_schema=	Identifies the Oracle schema where database objects for the product are created. The default Oracle schema is the same as the product abbreviation. You can change this if you want to put the product's database objects in a different schema. Moving a product's objects from one schema to another involves import/export and updates to internal Oracle Applications tables, so choose your initial schema carefully.

Entry	Description
sizing_factor=	Identifies the sizing factor Oracle Applications uses when creating tables and indexes for this product. The default value of 100 means 100%.The product's tables and indexes are created with the default sizes determined by Oracle. We recommend you accept the default sizing factor.
main_tspace=	Specifies the tablespace where this product's tables are created. To correctly follow OFA standards, create a new tablespace called <prod>D (where prod is the product abbreviation listed in the product= line) to hold the tables for this new product prior to running AD Splicer. Then set this value to <prod>D. AD Splicer fails if run with the default value (*Product_Name*D).
index_tspace=	Specifies the tablespace where this product's indexes are created. To correctly follow OFA standards, create a new tablespace called <prod>X (where prod is the product abbreviation listed in the product= line) to hold the indexes for this new product prior to running AD Splicer. Then set this value to <prod>X. AD Splicer fails if run with the default value (*Product_Name*X).
temp_tspace=	Specifies the tablespace 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. AD Splicer fails if you run it with the default value (*Temporary_Tablespace*).
default_tspace=	Specifies the default tablespace where this product's objects are 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. AD Splicer fails if run with the default value (*Product_Name*D).

File Character Set Converter (adncnv)

AD Administration, AutoPatch, and Rapid Install convert Oracle files from one character set to another automatically. If necessary, you can use the File Character Set Converter to manually convert files from other vendors. Before starting adncnv, log in as applmgr and complete the steps in the Setting the Environment section of Chapter 1. Use the following command to convert one file at a time:

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

The following parameters are required:

Parameter	Definition
source file	Path and file name of the file to convert.
source char set	Current character set.

Parameter	Definition
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:

UNIX:

```
$ cd $FND_TOP/sql
$ cp afcmstat.sql afcmstat.old
$ adncnv afcmstat.old we8dec afcmstat.sql we8hp
```

Windows:

```
C:\> cd %FND_TOP%\sql
C:\> copy afcmstat.sql afcmstat.old
C:\> adncnv afcmstat.old we8dec afcmstat.sql we8hp
```

3

Maintenance

You use AD Administration (adadmin) to perform maintenance tasks on an installed Oracle Applications system to ensure that it runs smoothly. This chapter contains these sections:

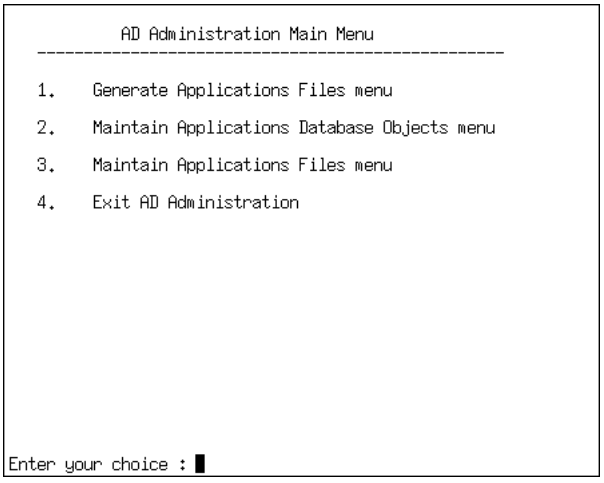
- AD Administration
- AD Relink (adrelink)

AD Administration

There are three types of AD Administration tasks: generating Applications files, maintaining the Applications database objects, and maintaining Applications files. You run all database and file system tasks by supplying the necessary information at the AD Administration prompts and then choosing an option from one of the AD Administration menus.

You can reduce the time it takes to respond to prompts by running some tasks *non-interactively*. This is useful for scheduling routine tasks that require little or no user intervention. AD Administration (like AutoPatch and AutoUpgrade) can run parallel workers for most database tasks and for some file system tasks.

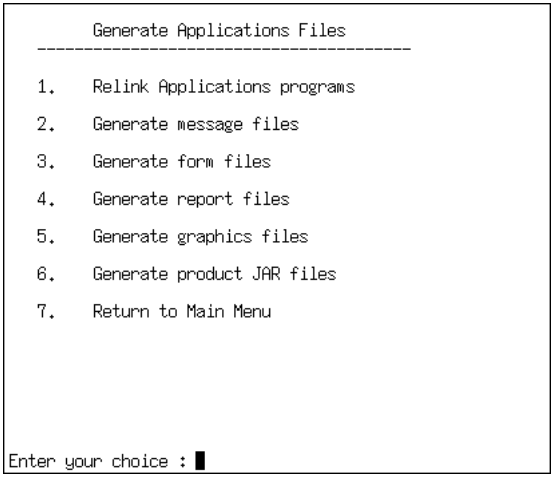
Before starting AD Administration, complete the steps in the Setting the Environment section of Chapter 1. Start the AD Administration utility with the *adadmin* command. AD Administration prompts you for the information it needs about your installation and then displays the Main Menu. You can choose to generate Applications files, maintain Applications database objects, maintain Applications files, or exit.



Note: The option names and numbers may differ depending on your configuration.

Generate Applications Files Tasks

From the AD Administration Main Menu, choose the Generate Applications Files menu option. This menu lists tasks related to the Oracle Applications product files.



Select the task number to perform a task. The generate file tasks may be performed on all servers, as needed. The following pages describe each task on this menu.

Relink Applications programs

Relinks Oracle Applications executable programs with the Oracle server libraries so that they function with the Oracle database. For each product, choose whether to link all executables or only specific ones.

You also have the option of relinking executables with debug information intact. Use this option only if requested to do so by Oracle Support Services. By default, AD Administration relinks all executables without debug information.

AD Administration does not link executables for the AD product. To relink AD executables, run the AD Relink utility.

> **Additional Information:** See Relinking AD Executables in *Oracle Applications Maintenance Procedures.*

Generate message files

Generates message binary files (extension .msb) from Oracle Application Object Library tables. Oracle Applications uses these files to display messages. Choose this task only when instructed to do so, such as in an update or a patch or by Oracle Support Services.

> **Additional Information:** See Generating Files in *Oracle Applications Maintenance Procedures.*

Generate forms files, reports files, and graphics files

These three menu options operate similarly and are explained together in this section.

Choose the Generate form files task to generate executable Oracle form files (extension **.**fmx) from the binary forms definition files (extension **.**fmb). The definition files are located under AU_TOP, and the executable files are stored under each product's directory.

Choose the Generate report files task to generate the binary Oracle Reports report files (extension .rdf).

Choose the Generate graphics files task to generate the Oracle Graphics files (extension .ogd) from the graphics definition files (extension .ogx).

> **Additional Information:** See Generating Product Files in *Oracle Applications Maintenance Procedures.*

All these tasks perform the same actions, except as noted:

- Prompt for the number of parallel workers (for generating files in parallel).
- Display the current character set (from NLS_LANG) and ask if you want to generate forms, reports, or graphics objects in this character set.
- (When generating forms), ask if you want to regenerate Oracle Forms PL/SQL library files, menu files, and executable files.
- Ask for the products for which you want to generate forms, reports, or graphics objects.
- Ask if you want to generate specific forms, reports, or graphics objects for each selected product.
- Display the current set of installed languages and ask if you want to generate forms, reports, or graphics files in these languages.
- Create a list of all objects to generate.
- Display the list of objects to be generated. You can generate specific objects or all objects.
- Generate all selected objects for all selected products in parallel.

> **Note:** If any forms, reports, or graphics objects did not generate successfully, AD Administration displays a list of warnings or errors, and asks if you want to continue as if successful. Review the AD Administration log file to determine if the problems require attention. If you choose not to continue and restart your session at a later time, AD Administration attempts to regenerate only the files that did not generate successfully.

Generate product JAR files

Run this task whenever you upgrade the Developer6*i* technology stack or when recommended by Oracle Support Services. This task prompts:

```
Do you wish to force generation off all jar files? [No]
```

If you choose No, it only generates JAR (Java archive) files that are missing or out-of-date. Choose Yes for this option when generating JAR files after upgrading the Developer 6*i* tech stack.

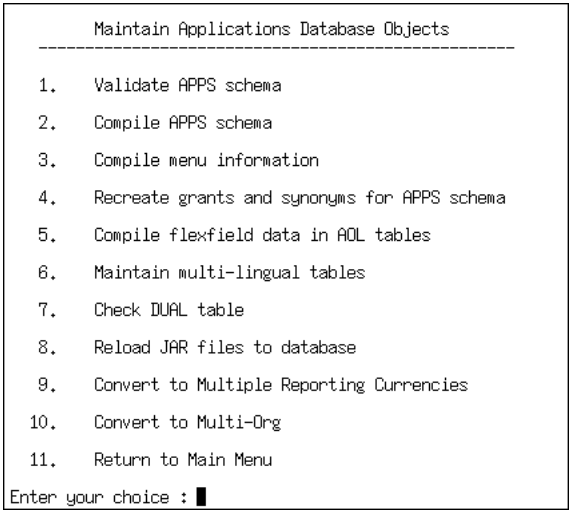
This task:

- generates product JAR files in JAVA_TOP and copies them to APPL_TOP.
- copies Oracle Forms registry file (Registry.dat) from ORACLE_HOME/forms60/java to JAVA_TOP/oracle/forms/registry.
- signs JAR files, if on the web server.
- recreates appsborg.zip and appsborg2.zip under APPL_TOP and JAVA_TOP.

This task fails if any Oracle product JAR files do not generate successfully. Review the AD Administration log file to determine whether the problems require additional attention. Restarting a failed AD Administration session attempts to generate only the Oracle product JAR files that did not generate successfully.

Maintain Applications Database Objects Tasks

From the Main Menu, choose the Maintain Applications Database Objects Menu option to see a list of tasks you can perform on installed Oracle Applications database objects. Run these tasks *only* on the admin server.



Select a task number to perform a task. The following pages describe each task on this menu.

Validate APPS schema

Runs a SQL script (advrfapp.sql) against the APPS schema to verify the integrity of the schema. It determines:

- Problems you MUST fix (not specific to the APPS schema)
- Problems you MUST fix (specific to the APPS schema)
- Issues you may want to address (specific to the APPS schema)

The problems and issues are described in separate sections in a report named <APPS schema name>.lst. This report is located in $APPL_TOP/admin/<SID>/out, where <SID> is the value of the ORACLE_SID or TWO_TASK variable (UNIX), or in %APPL_TOP%\admin\<SID>\out, where <SID> is the value of your LOCAL variable (Windows). Review the report, fix any problems, and rerun the task until no problems are listed.

> **Additional Information:** See Compiling Invalid Objects in *Oracle Applications Maintenance Procedures.*

> **Note:** You must fix all reported problems in the first two sections before running the Maintain Multiple Reporting Currencies task. Problems in the third section will not cause failure, but you should resolve issues in the third section when possible.

You can also run this task with SQL*Plus:

UNIX:

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

Windows:

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

Compile APPS schema

Spawns parallel workers to compile invalid database objects in the APPS schema.

> **Additional Information:** See Compiling Invalid Objects in *Oracle Applications Maintenance Procedures.*

AD Administration prompts:

```
Run Invoker's Rights processing in incremental mode [No] ?
```

When you type Yes at this prompt, Invoker Rights processing runs only on packages that have changed since Invoker Rights processing was last run.

> **Note:** Invoker Rights processing will not modify any of your custom packages, procedures, or functions unless you imbed RCS header information (in the identical format to those found in Oracle Applications PL/SQL files) in the first five lines of your PL/SQL source text.

> **Additional Information:** See the Oracle server documentation, and Invoker Rights in *Oracle Applications Concepts*.

Compile menu information

Compiles menu data structures. Choose this task after you have uploaded menu entries to the FND_MENU_ENTRIES table, or if Compile Security concurrent requests submitted from the Menus form (after changing menu entries) fail for any reason.

AD Administration asks if you want to force compilation of all menus. If you choose the default (No), only menus with changes are compiled. If you enter Yes, all menus are compiled. Compiling all menus is generally not advised.

Recreate grants and synonyms for APPS schema

Recreates grants and synonyms for the Oracle Applications public schema (APPLSYSPUB), recreates grants on some packages from SYSTEM to APPS, and spawns parallel workers to recreate grants and synonyms linking sequences and tables in the base schemas to the APPS schema.

To pro-actively verify that grants and synonyms are up to date, first run the Validate APPS Schema task. If you determine that grants and synonyms are missing, run this option to recreate them.

> **Note:** To set up grants and synonyms for the MRC schema, run the Maintain MRC task from the Database Objects menu after compiling the APPS schema and fixing any issues.

Compile flexfield data in AOL tables

Compiles flexfield data structures in Oracle Application Object Library (AOL) tables. Using this option after you modify flexfields for the first time improves performance at runtime.

Maintain multi-lingual tables

Calls PL/SQL routines to maintain multi-lingual tables. Run this task when adding a language. The task prompts you for the number of workers, then updates all multi-lingual tables in parallel.

Check DUAL table

Verifies that the DUAL table exists in the SYS schema, is accessible by Applications, and contains only one row. If the DUAL table does not exist, or if it does not contain exactly one row, Oracle Applications products that access this table will fail.

Reload JAR files to database

Use this option if all Oracle Applications Java classes have been removed from your database. (This can happen if your database becomes corrupt and the database Java Virtual Machine (JVM) is reloaded.)

This option runs the loadjava utility to reload all appropriate Oracle Applications JAR files into the database.

Maintain [or Convert to] Multiple Reporting Currencies

If you have installed Multiple Reporting Currencies (MRC) functionality, this menu option is called Maintain Multiple Reporting Currencies schema(s). If you have not, it is called Convert to Multiple Reporting Currencies option, which you use to install MRC.

Attention: Always run the Validate APPS Schema task before running the Maintain [Convert to] Multiple Reporting Currencies task.

MRC is implemented using an *adjunct schema*, which is an extra schema that 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.

MRC takes advantage of Invoker Rights, which eliminates the need to copy most packages from the APPS schema.

Additional Information: See Invoker Rights in *Oracle Applications Concepts*.

After upgrading an MRC-enabled database from Release 11.0 to Release 11*i*, choose the Maintain Multiple Reporting Currencies schema(s) task to synchronize the database objects in the MRC schema with those that may have changed in the APPS schema. It 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.
- Calls a PL/SQL procedure to maintain database objects in the MRC schema.
- Compiles all invalid objects in the MRC schema in parallel and runs Invoker Rights processing.
- Recreates MRC triggers.

Accept the default answers (Yes) to all questions when running MRC maintenance as a post-upgrade step.

Once AD Administration completes the process, check the log file (adadmin.log) for problems. Rerun this task until the report contains no problems.

Note: Shut down the concurrent managers before updating the MRC schema.

Additional Information: See Converting to Multiple Reporting Currencies in *Oracle Applications Maintenance Procedures.* See also *Multiple Reporting Currencies in Oracle Applications.*

Convert to Multi-Org

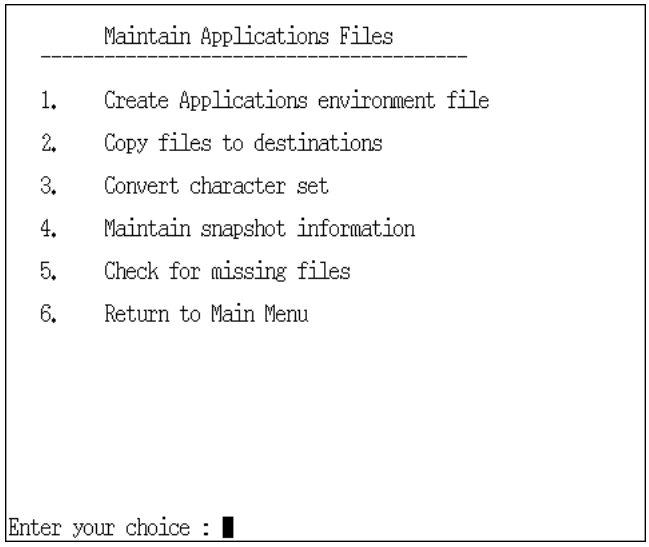
Appears as a menu choice only if Multi-Org is *not* installed in your database. Use it to convert a standard product group into a Multi-Org product group with one operating unit defined at the site level. The Convert to MultiOrg task does the following:

- Asks for the number of parallel workers (and dynamically uses multiple workers on any Multi-Org partitioned table that has more than one million rows).
- Disables all enabled triggers in the APPS schema.
- Converts seed data and transaction data to Multi-Org in parallel.
- Re-enables all previously disabled triggers in the APPS schema.

Additional Information: See *Multiple Organizations in Oracle Applications*. See also Converting to Multi-Org in *Oracle Applications Maintenance Procedures*.

Maintain Applications File Tasks

From the AD Administration Main Menu, choose the Maintain Applications Files menu option. This menu lists tasks related to maintaining the Oracle Applications product files.



Select the task number to perform a task. The file system tasks may be performed on all servers, as needed. The following pages describe each task on this menu.

Create Applications environment file (UNIX)

Choose this option to create an environment file that defines Oracle Applications environment variables. We recommend backing up the existing environment file before running this task.

Respond to the prompts for additional information. After the utility generates the environment file, you can make customizations in adovars.env and run the generated environment file as necessary.

Environment File Name AD Administration prompts for the file name to use when creating the environment file. The default is <SID>.env.

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

Note: If AutoConfig is enabled, AD Administration will automatically generate the environment file with asking any other questions. If AutoConfig is not enabled, AD Administration will ask for the following information before generating the environment file.

Parallel Concurrent Processing The task asks 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, and in the *System Administrator's Guide.*

DOS-Compatible File Names The task next asks 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 task prompts 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. This task sets 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 this is the only product group you have installed.

```
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] :
```

Directories for Temporary Files The task prompts 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 Directories for Temporary Files This task requires 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:
```

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.

> **Additional Information:** See Initialization Parameters the Oracle9*i* Release 2 documentation.

Web Server Oracle Applications forms and context-sensitive help are accessed from the desktop client through a connection to a web server. The task prompts 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.

Verify the New Environment File After you complete the final question, AD Administration creates the environment file. The following message appears:

```
Creating environment file...

Done.

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

The new environment file is placed directly under APPL_TOP.

Create Applications environment subkey in registry (Windows)

Choose this option to create an environment subkey to define Oracle Applications variables in the Windows registry. We recommend backing up the existing environment subkey before you begin.

At the prompt for the subkey name, enter any name (no extension is necessary). The default is <CONTEXT_NAME>. Respond to the prompts for additional information needed for the Applications environment. After the utility generates the environment subkey, you can customize it as necessary.

> **Note:** The <CONTEXT_NAME> variable defaults to <SID>_<hostname>. You may set it to <SID> or customize it to some other environment-specific name.

> **Note:** The Oracle Applications registry subkeys are created under the \\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\APPLICATIONS\11.5.0 subkey in the Windows registry.

Copy files to destinations

Copies files from each product area to central locations where they can be easily referenced by non-Applications programs. This option uses revision-based copy logic to ensure that the destination file versions are the same as, or higher than, the source file versions.

Attention: We recommend that you do not use the "force" option to overwrite existing files unless instructed by Oracle Support Services. Copying files with this option updates all JAR files. JInitiator then downloads required JAR files to each client again, causing runtime performance degradation.

The file types and their respective destinations are shown in the following table:

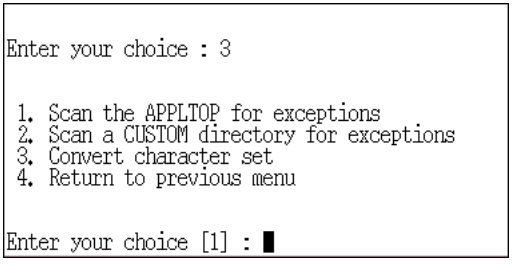
These files:	...are copied to (UNIX)	...are copied to (Windows)
Java files	$JAVA_TOP	%JAVA_TOP%
HTML files	$OAH_TOP	%OAH_TOP%
Media files	$OAM_TOP	%OAM_TOP%

The directories for the variables are specified in the adovars.env file (UNIX) or the adovars.cmd file (Windows).

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

Convert character set

Converts the character set of files in the APPL_TOP. The task displays the following submenu:



- Scan the APPLTOP for exceptions

  This task scans the APPL_TOP and creates three files in the admin\<SID>\out directory:

File	Contents
admanifest_excp.lst	Lists files which will not be converted because of lossy conversion. These files must be fixed before running the convert character set task (Task 3).
admanifest.lst	Lists files which can be converted.
admanifest_lossy.lst	Lists files which have lossy conversion along with details of those lines which are lossy.

You must fix any files that report lossy conversion. Run this until there are no entries in the admanifest_excp.lst report before running the conversion (task 3).

- Scan a CUSTOM directory for exceptions

This task is similar to the first task, except it scans custom Applications directories rather than the APPL_TOP directory.

3. Convert character set

To avoid lossy conversions, run this task only if admanifest_excp.lst has no entries. The task prompts you for the manifest file, which is the admanifest.lst file created by either task 1 or task 2.

Before the conversion, the utility backs up the product source files and the APPL_TOP/admin source files. It saves product files in the <PROD>_TOP directories in the format <prod>_s_<char_set>.zip. It saves admin source files in the APPL_TOP/admin directory in the format admin_s_<char_set>.zip

Maintain snapshot information

This task stores information about files, file versions, and bug fixes present in an APPL_TOP. Chose this option to record the current set of files and file versions in your APPL_TOP.

This task prompts you to manage snapshots within this Applications system, or to migrate snapshot information to or from another Applications system. (Choose to manage snapshots within the Applications system, as the other option is not currently implemented.) Choose whether to create a named snapshot or refresh the current view. The task prompts whether to refresh the snapshot with all bug fixes applied to all APPL_TOP directories. Accept the default answer (No) unless you believe the set of bug fixes listed for the current APPL_TOP is incomplete.

There are two types of APPL_TOP snapshots: current view snapshots and named snapshots. Current view snapshots are created once and updated when appropriate

to maintain a consistent view of the APPL_TOP contents. Named snapshots are created once and are not updated.

You can create as many named snapshots of each APPL_TOP as you want, or you can create or update the current view snapshot only. Creating a named snapshot for an APPL_TOP automatically creates or updates the current view snapshot for that APPL_TOP. The bug fix information for a named snapshot is copied from the current view snapshot for that APPL_TOP.

You must run this task once for each APPL_TOP before you can apply any patch that contains a "compatible feature prereq" line on that APPL_TOP. A complete snapshot is required for the automatic prerequisite patch checking feature to operate.

AutoPatch automatically updates the list of file versions and bug fixes in the current view snapshot for each APPL_TOP as patches are applied to that APPL_TOP. The combination of running AD Administration "Maintain snapshot information" once for an APPL_TOP and AutoPatch's incremental updates ensures that the current view snapshot for a given APPL_TOP contains an accurate picture of the current set of files and bug fixes present in that APPL_TOP.

APPL_TOP snapshot information is stored in the AD_SNAPSHOTS, AD_SNAPSHOT_FILES, and AD_SNAPSHOT_BUGFIXES tables.

Check for missing files

Verifies that all files needed to install, to upgrade, or to run Oracle Applications for the current configuration are in the current APPL_TOP. Choose this task if you suspect there are files missing in your APPL_TOP.

Using AD Administration Non-Interactively

AD Administration can run some file system and database tasks non-interactively. To use AD Administration non-interactively, you create a defaults file by running AD Administration through maintenance tasks that you would like to schedule in the future.

Once the defaults file is created, you run AD Administration from the command line, using the interactive=no parameter and specifying the defaults file. The utility runs the task using the parameters it finds in the defaults file. No other user intervention is necessary

> **Additional Information:** See Performing Tasks Non-interactively in *Oracle Applications Maintenance Procedures*.

Note: The Convert to Multiple Reporting Currencies, Create Applications environment file, Convert character set, and Convert to MultiOrg tasks are not supported when running non-interactively.

AD Relink (adrelink)

This utility relinks Oracle Applications executable programs with the Oracle server product libraries. For most products, you run this utility in AD Administration by choosing the Relink Applications Programs task from the Maintain Applications Files menu. However, because you cannot use AD Administration to relink AD executables, you must run AD Relink manually to link these programs.

Additional Information: See Relinking AD Executables in *Oracle Applications Maintenance Procedures.*

If an error occurs during relinking, or if you are not sure that the relinking was successful, review the AD Relink log file (adrelink.log). If AD Relink was run by AutoUpgrade, AD Administration, or AutoPatch, it is located in APPL_TOP/admin/<SID>/log, where <SID> is the value of your ORACLE_SID or TWO_TASK variable (UNIX) or the value of ORACLE_SID or LOCAL (Windows). If you run AD Relink from the command line, the file is in APPL_TOP/admin/log. As it runs, AD Relink appends information about the latest relink action to the end of the file.

To recover disk space, you can delete the adrelink.log file if you do not need the information. A new log file is created each time AD Relink runs.

AD Relink requires several parameters, including the "force" parameter and optional arguments. There is no default value for the "force" parameter: setting it to "n" means relink the executable program only if the dependent libraries or object files are more recent than the current executable program, and setting it to "y" means relink regardless of the status of the libraries or object files.

Type *adrelink.sh* to see online instructions about syntax for this utility. Type *adrelink.sh examples* to see more examples.

4

Patching

AutoPatch (adpatch) is a utility that applies patches and adds new products and languages to an Oracle Applications installation. This chapter describes the way AutoPatch works, and contains these sections:

Patches

Throughout the course of the Oracle Applications life cycle, patches are applied for a number of reasons, including:

- Updating to a higher maintenance level (maintenance pack)
- Applying the latest product enhancements (mini-pack)
- Adding a new feature or functionality
- Fixing an existing issue

All Oracle Applications patches update the file system, or the database, or both, and are available from Oracle*MetaLink*. The patch types are:

Patch Type	Description
Bug fix	Fixes an existing issue.
New feature	Adds new functionality.

Patch Type	Description
Interoperability	Contains Oracle Applications files and database objects to make the current version of Oracle Applications compatible with a newer version of the database or a technology stack component, for example, Oracle 9*i* interoperability with Oracle Applications Release 11.5.9.
Diagnostic	Released specifically to determine the source of an issue. A diagnostic patch does not fix the issue.
Translation	Contains Oracle Applications files that have been translated from English to another languages. A translation patch may also execute tasks to load or update data in the database.
Upgrade	Fixes a problem with an upgrade or improves the performance of an upgrade from a previous major release, such as 10.7 or 11.0.
Documentation	Updates Oracle Applications Online Help. When applying a product mini-pack or a stand-alone patch that adds a new feature, review the Features Summary Matrices on Oracle*MetaLink* for the coordinating documentation patch.

Patches are released in the following formats:

Patches	Description
Stand-alone	Fixes a specific issue or provides new functionality.
High-priority	Identified by Oracle Development as having an impact that is broad enough to merit application by all customers who have installed the affected product.
Mini-pack	A consolidation of all patches for a product. The naming convention is 11i.<product>.<mini-pack letter> such as 11i.AD.G. Subsequent mini-packs (those with higher <mini-pack letter>) supersede previous versions. Mini-packs are cumulative.
Family pack	A consolidation of a set of mini-packs and other patches for a product family. Subsequent family packs (those with higher number) supersede previous versions. Family packs are cumulative.
Family consolidated upgrade patch	All *upgrade-related*, high-priority patches consolidated from all the products within a product family. Family consolidated upgrade patches are released as needed. The *Oracle Applications Release Notes* lists the most recent family consolidated upgrade patches.

Patches	Description
Maintenance pack	A consolidation of all product mini-packs. A maintenance pack updates a system to a new point release of Oracle Applications, such as Release 11.5.8. Subsequent maintenance packs (those with higher number) supersede previous versions. Maintenance packs are cumulative.

Patch Structure

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 named <patchnum>, where <patchnum> is the number for the patch. The most important files in the top-level directory are: README.txt, README.html, c<patchnum>.drv, d<patchnum>.drv, and g<patchnum>.drv.

> **WARNING: Patches must always be applied in their entirety. If you apply a patch to update the file system, you must also apply the corresponding database and generation portions (if any).**

The README.txt or README.html file contains important information about the patch. It describes what the patch does, and may include a list of the files in the patch, which servers to run the patch on, and instructions for manual steps associated with applying the patch.

A patch may contain one or more of the following patch drivers: a copy driver, a database driver, and a generate driver.

Patch driver files must be applied in the following order: copy driver, database driver, and generate driver. If you apply the database driver to an APPL_TOP, you must apply the copy driver first. If you will apply the generate driver to an APPL_TOP, you must apply the copy driver (and perhaps the database driver) first.

Copy Driver

The copy driver is named c<patchnum>.drv, and contains commands to change Oracle Applications files. In a multi-node system, run the copy driver on all APPL_TOP directories containing one or more of the files being replaced by the patch. If in doubt, apply it on all APPL_TOP directories on all nodes.

> **See Also:** Applying Patches to a Multi-Node System in *Oracle Applications Maintenance Procedures*.

When applying the copy driver, AutoPatch:

- Extracts the appropriate files from each product's C library.
- 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.
- 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 the patch directory. For example, if <patch_dir> is the patch directory, <system_name> is the Applications Environment name, <appl_top_name> is the APPL_TOP name, and <prod> is the name of the product being patched, 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>
  ```

  > **Note:** The Applications System name and the APPL_TOP name are determined during the Rapid Install process.

- Replaces each product's outdated files with newer files from the patch directory.
- Loads the new object modules into the C libraries.
- Relinks the Oracle Applications products with the operating system, Oracle server, and other Oracle products libraries.
- Applies changed Java class files and regenerates JAR files as needed.
- Copies any specified HTML or media files to their respective destinations.
- Compiles out-of-date Java Server Page (JSP) files, if any JSP files are included in the patch.

Database Driver

The database driver is named d<patchnum>.drv and contains commands to change Oracle Applications database objects. Run the database driver only on the APPL_TOP that implements the admin server.

When applying the database driver, AutoPatch:

- Gets a list of current invalid objects in the APPS schema.
- Determines whether the action was performed in a previous patch.
- Runs SQL scripts and exec commands, which change Oracle Applications database objects. By default, AutoPatch does this in parallel.
- Performs Invoker Rights processing if the patch contains a package command.

> **Note:** Invoker Rights processing will not modify any of your custom packages, procedures, or functions unless you imbed RCS header information (in the identical format to those found in Oracle Applications PL/SQL files) in the first five lines of your PL/SQL source text.

- Maintains the MRC schema (if MRC is enabled).
- Compiles invalid objects in the database.
- Gets a list of current invalid objects in the APPS schema.

Generate Driver

The generate driver is named is g<patchnum>.drv and contains commands to generate forms, reports, graphics, or message files. Run this driver file on all APPL_TOP directories containing one or more files being generated by the patch. If in doubt, apply it to all APPL_TOP directories on all nodes.

Unified Driver

Some Oracle Applications patches contain a single driver file named u<patchnum>.drv. This driver contains commands to change Oracle Applications files, change Oracle Applications database objects, and to generate forms, reports, graphics, or message files. The u<patchnum>.drv is called a unified driver.

The unified driver replaces the c<patchnum>.drv, d<patchnum>.drv, and g<patchnum>.drv drivers. You apply a unified driver to all APPL_TOPS.

AutoPatch only runs the actions in the unified driver that are required for the current APPL_TOP.

The Release 11.5.9 Maintenance Pack was a unified driver. Unified drivers will eventually be used by most Oracle Applications patches. AutoPatch and AD Merge support both patches with split drivers and patches with unified drivers.

AutoPatch

AutoPatch is an interactive utility for applying patches to the Oracle Applications file system or database. Patches are necessary to update your system, to add new functionality, and to fix issues in your system. When applying patches, AutoPatch:

- Prompts for required information about the patch.
- Unloads patch metadata and validates prerequisite patches have been applied.
- Uploads patch history information to the database (if needed).
- Reads and validates the patch driver file.
- Reads product file driver files.
- Extracts object modules from the product libraries (so it can compare version numbers on the object modules it extracts).
- Compares the version numbers of the existing files against the files in the patch (version checking).
- Backs up any existing files that will be changed.
- Copies files.
- Archives files in libraries
- Relinks executables.
- Generates Java archive (JAR) files.
- Compiles JSP files.
- Updates database objects.
- Maintains the MRC schema.
- Compiles invalid database objects.
- Generates forms, reports, message, and graphics files.
- Saves patch history information to the database.

AutoPatch takes no action if it finds a patch contains no new updates to your files or database objects. Before it makes any changes to existing files, AutoPatch first makes a backup copy.

Patches usually include three driver files: a copy driver, a database driver, and a generate driver. A patch may, however, contain only one or two drivers. For most patches, running the patch drivers with AutoPatch is the only action required.

> **See Also:** Non-interactive Patching in this chapter.

Some patches require other steps besides running driver files, such as applying prerequisite patches and performing manual steps. The readme file for each patch explains the steps required to completely apply the patch.

> **Note:** See Applying Patches in *Oracle Applications Maintenance Procedures* for more detail about applying patches and the patching process.

When running AutoPatch interactively, you respond to the all prompts for each driver that you run. For example, if a patch requires you to run a copy driver, a database driver and a generate driver, you run AutoPatch three times and respond to all prompts when applying the patch.

> **Attention:** Do not run multiple sessions of AutoPatch on the same Applications system at the same time.

AutoPatch Features

Certain AutoPatch features help automate the process of applying patches and verify that patches will be correctly and successfully applied.

Patch History Database

In earlier releases, patch history information was stored only in the file system in a file called applptch.txt. This file contained information about all patches successfully applied to an Oracle Applications system. The new AutoPatch patch history database feature moves the contents of this file to Oracle Applications Manager (OAM) database tables. Using the OAM patch history interface, you can access all patch history information from the database through a set of query options and reports.

With the patch history database feature, you can determine:

- Which patches have been applied to a system.
- When patches were applied.
- What bug fixes were included in a patch.
- When a file was last patched.

> **Additional Information:** See Chapter 6, "Reports" in this book, and Analyzing Your Patching History in *Oracle Applications Maintenance Procedures*.

Patch History File System Changes

In this release, when AutoPatch saves patch history information to the file system, it writes it to the adpsv<timestamp>.txt and javaupdates<timestamp>.txt files. AutoPatch saves patch history information to the file system when:

- you run AutoPatch in pre-install mode
- its attempt to save patch history to the database fails
- there is a mismatch between the AutoPatch executable and the patch history tables in the database

In addition to applptch.txt, AutoPatch uploads patch history information from adpsv<timestamp>.txt and javaupdates<timestamp>.txt. When the information from any of these files is successfully uploaded to the database, the file is removed from the file system. If the upload fails, the file remains on the file system.

> **Additional Information:** See Patch History Database and AutoPatch Modes in Chapter 6.

Prerequisite Patch Checking

AutoPatch checks to see if all pre-requisite patches have been applied before applying a patch. If all prerequisite patches have been applied, AutoPatch applies the patch. If all prerequisite patches have not been applied, AutoPatch lists the missing prerequisite patches and does not apply the patch.

Recent Release 11*i* patches include an .ldt file at the top level that contains metadata about the patch. The prerequisite patch checking feature loads the patch metadata .ldt file and then checks to see if all prerequisite patches have been applied. Besides the .ldt file, patches that use the automatic prerequisite patch checking feature also include the line "compatible feature prereq" in the patch driver files. This line

prevents these patches from being applied using older versions of AutoPatch that do not support the automatic prerequisite patch checking feature.

> **Note:** You should continue to check the readme of each patch you apply to see if any prerequisite patches are required. You can use the Patch History feature of Oracle Applications Manager to see if you have already applied a given patch.

Prerequisite patch checking is performed when the copy driver for a patch is applied. It is not performed when the database or generate drivers for a patch are applied.

The prerequisite patch checking feature prevents you from applying a translation patch if you have not first applied the base (US) version of the patch. If a patch has already been applied, AutoPatch asks if you want to re-apply it.

Checkfile Feature

The AutoPatch checkfile feature reduces patch application downtime by checking whether a database action has been performed previously for the associated file contained in the patch. If a database action has been performed using the current version or a higher version of the file, AutoPatch omits the action.

When you run a checkfile-enabled database driver file, AutoPatch:

- Gets the version information from the specified file.
- Checks the AD_CHECK_FILES table to determine whether this version of the file and its arguments have been run previously.
- Determines whether the action is really redundant.
- Performs the action if the file has not previously been run.
- After running all database actions, updates the AD_CHECK_FILES table with information about the database actions that ran.

Checkfile-enabled patch driver files contain the line "compatible feature checkfile". In addition, all SQL and EXEC commands in the patch driver file have an argument that starts with "checkfile". The checkfile feature specifically looks at the SQL and EXEC commands in the patch driver file and performs a version check of the files in the command. It also distinguishes whether a specific action on a file has been done. For example, an SQL script may run twice, each type with a different set of

arguments, and checkfile can determine that these are not redundant actions and will perform both.

Applying Integrated Patches

You can merge several compatible patches using the AD Merge Patch utility, rather than applying each patch individually. This removes duplication and helps automate the patching process. For example, if two or more patches contain the same action, the integrated patch will only perform this action once. If two patches contain two different revisions of a file, the integrated patch only contains the later revision of the file. Also, when applying the integrated patch, you only need to run AutoPatch once for each driver in the integrated patch, rather than once for each driver in all individual patches. Applying integrated patches reduces the amount of time to apply patches.

> **Additional Information:** See AD Merge Patch (admrgpch) in this chapter, and Merging Patches in *Oracle Applications Maintenance Procedures*.

Non-interactive Patching

Non-interactive patching is a way to avoid some of the prompts and automate the patching process. First create a defaults file by running AutoPatch interactively with a specific command line option. Alternatively, you can copy $APPL_TOP/admin/adalldefaults.txt to $APPL_TOP/admin/<SID>/<new_file>.txt and edit it as needed. Then run AutoPatch non-interactively, providing the name of the defaults file you created plus other command line options. After AutoPatch completes, perform any post-AutoPatch steps listed in the patch readme file.

> **Additional Information:** See Non-interactive Patching in *Oracle Applications Maintenance Procedures*.

AutoPatch Modes

AutoPatch can apply patches in two specialized modes: test mode and pre-install mode. The patch readme file or documentation instructs you when to use one of these modes.

Test Mode

With test mode, you can see the effects applying a patch will have on your production system before applying the patch. In test mode, AutoPatch doesn't apply the patch. Instead, it lists each file it would have copied, relinked, executed,

or generated and shows exactly what actions it would have performed had it applied the patch.

Applying a patch in test mode works like applying a patch interactively, with the following exceptions. It does *not*:

- Copy any files from the patch directory to the installation area.
- Archive any object modules into the 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 patch information in the database.
- Update the release version in the database.

To run AutoPatch in test mode, you must include *apply=no* on the AutoPatch command line.

Because test mode does not copy over new file driver files, you may get inaccurate results when applying a patch in test mode.

> **Additional Information:** See Testing a Patch Before Applying It in *Oracle Applications Maintenance Procedures.*

Pre-install Mode

Pre-install mode is generally used during the upgrade process to update AD utilities, apply family consolidated upgrade patches, or to work around other patching issues. AutoPatch asks all normal start-up questions except those relating to the database. The patch readme file will instruct you when to use pre-install mode.

> **Note:** Run AutoPatch in pre-install mode *only* if the patch requires it.

Applying a patch in pre-install mode performs the following actions only:

- Version checking.
- File copy actions

- Relink FND and AD executables
- Save patch history information to the file system

> **Note:** Because AutoPatch does not read driver files in pre-install mode, it copies all product files in the patch to the APPL_TOP directory, even if they should not exist on this node. For example, it will copy forms files to the APPL_TOP that only implements the admin server.
>
> Additionally, if a file in the patch should in APPL_TOP and in another directory (such as in $OA_HTML), AutoPatch will only copy the file to APPL_TOP in pre-install mode.

To run AutoPatch in pre-install mode, include the command line argument *preinstall=y* on the AutoPatch command line.

Running AutoPatch

AutoPatch is located in the AD_TOP/bin directory. You usually start AutoPatch using the *adpatch* command from the directory that contains the unzipped patch files. You can exit AutoPatch by entering *abort* at any prompt. You can then restart AutoPatch from where the session ended or from the beginning.

> **Additional Information:** See Applying a Patch in *Oracle Applications Maintenance Procedures*.

In addition to the standard prompts common to most AD utilities, AutoPatch also asks for the following information specific to the patching process:

AutoPatch log file

The default log file is adpatch.log. This file is in $APPL_TOP/admin/<SID>/log, where <SID> is the value of your ORACLE_SID or TWO_TASK variable (UNIX), or in %APPL_TOP%\admin \<SID>\log, where <SID> is the value of ORACLE_SID or LOCAL (Windows).

> **Note:** When applying a patch, we recommend you change the log file name. Use the associated driver file name with a .log extension, such as d123456.log.

AutoPatch Password Hider

Lines in the AutoPatch log file containing passwords are automatically masked to not display the passwords. By default, each line containing masked passwords is followed by a corresponding line prefixed with HIDEPW, which contains the original line with the passwords included. When options=hidepw is specified on the AutoPatch command line, the additional HIDEPW line is omitted.

The intent of this feature is to include passwords protected by HIDEPW comments in the AutoPatch log file by default so that debugging is easier and stripping the passwords is easy. For example, strip the passwords from a log file:

UNIX

```
$ grep -v 'HIDEPW:' file > file_no_pws
```

Windows

```
C:\> grep -v 'HIDEPW:' file > file_no_pws
```

In cases where security is more important than ease of debugging, use options=hidepw to prevent passwords from appearing in the AutoPatch log file.

SYSTEM and AOL user passwords

After you enter the SYSTEM and AOL user passwords, AutoPatch optionally validates the password information for all Oracle Applications schemas and displays this information on the screen. By default, it does not validate the password information for Oracle Applications schemas. You can make AutoPatch validate schemas using options=validate on the AutoPatch command line.

Patch directory

AutoPatch asks you to specify the directory where the patch files have been unzipped. The default is the directory from which you started AutoPatch. If necessary, specify the full path name to the directory where you unzipped the patch files. The operating system user running AutoPatch must have write permissions on the directory where the patch files have been unzipped.

Patch driver file

AutoPatch prompts for the patch driver file. After you enter the file name, AutoPatch optionally checks the integrity of the patch driver file and determines which patches in the file need to be applied. By default, AutoPatch does not check the integrity of the patch, as Oracle Applications patches are tested to make sure

they contain the correct files before they are released. You can make AutoPatch check the integrity of the patch using options=integrity on the AutoPatch command line.

Number of parallel workers

In parallel mode, tasks are assigned to workers, the workers run the tasks to completion, and AutoPatch assigns new tasks. Use the AD Controller utility to monitor and alter AutoPatch parallel worker status.

> **Additional Information:** See AD Controller (adctrl) in Chapter 1 and Parallel Processing in Chapter 1.

> **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.

By default, AutoPatch runs database updates and file generation commands in parallel mode and prompts you for the number of parallel workers:

```
Enter the number of parallel workers [3] :
```

AutoPatch reviews the contents of the applcust.txt file to determine if any registered customized files will be replaced by the patch. If so, it displays a message listing the customized files it will replace.

> **Additional Information:** See Customization Standards in *Oracle Applications Developer's Guide.*

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...
```

Restarting AutoPatch

If you aborted an AutoPatch session or it did not run to completion, you can restart AutoPatch. AutoPatch prompts for the name of the log file. If you provide the log file name from the previous session, it appends to that log file. AutoPatch then asks if you want to continue the previous session. If you answer Yes to the restart prompt, AutoPatch restarts where the previous session stopped.

If you answer No, AutoPatch asks you to confirm your choice and then starts a new AutoPatch session. If the FND_INSTALL_PROCESSES table already exists, AutoPatch asks if you want to drop the table. You must determine if AutoPatch, or any other AD Utility, is running in another session, or whether a previous patch session did not run to completion. If any AD utility is running in another session or on another node, wait until that session is complete before you start a new AutoPatch session in the current environment. If a previous patch session did not complete, resume applying that patch before you continue with a new one.

Understanding AutoPatch Messages

Monitor AutoPatch to check for error messages while it is running. Informational messages are written to an informational log file in the same directory as the AutoPatch log file. This log file has the same base file name, but with an .lgi extension instead of a .log extension. For example, if the AutoPatch log file is named d123456.log, the AutoPatch informational log file is named d123456.lgi.

The following example message indicates that AutoPatch will not update the files listed because they are up-to-date.

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

Error Messages

When AutoPatch is not running jobs in parallel mode and an error occurs, AutoPatch asks if it should continue. We recommend you do not continue. After AutoPatch exits, review the log files to determine the source of the error, and restart

AutoPatch once the error is resolved. If an error or a problem cannot be resolved, you should:

- Verify that all steps in the readme file were completed
- Check Oracle*MetaLink* for additional information regarding the patch being applied

When AutoPatch is running jobs in parallel mode and a worker fails its job, you do not need to wait until the other workers and the manager (AutoPatch) stop. You can fix the problem and restart the worker while the manager is running. Some failed jobs are deferred by the manager. These deferred jobs do not cause the manager or other workers to stop.

> **Additional Information:** See Troubleshooting in the *Oracle Applications Maintenance Procedures*.

Successful Completion Message

AutoPatch displays messages like the following once it runs to completion:

```
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.
```

If you do not see the "AutoPatch is complete" message at the end of your AutoPatch session, AutoPatch did not complete successfully and you should investigate the cause of the AutoPatch failure.

Log Files

Review the log files when AutoPatch has finished successfully. 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 restart files ($APPL_TOP/admin/<SID>/restart/adpprod.*) record the passwords to your Oracle Applications products. Restrict access to these files. Depending on whether you are running AutoPatch with options=hidepw, the AutoPatch log files may also contain passwords on lines prefixed with the HIDEPW comment.

If you have converted to MRC, AutoPatch automatically maintains your MRC schema after applying patches that update the database. We recommend reviewing the AutoPatch log file, however, to verify that the MRC portion ran successfully. If it did not, correct the error and run Maintain MRC manually from AD Administration.

AutoPatch Command Line Arguments

In addition to the command line arguments listed in Chapter 1, the following arguments can be used by AutoPatch:

preinstall	
Purpose	Tells adpatch whether to run in pre-install mode. Pre-install mode is used to update AD utilities before an upgrade and to apply family consolidated upgrade patches.
Values	y or n
Default	n, meaning that adpatch does not run in pre-install mode.
Example	adpatch preinstall=y

apply	
Purpose	Tells adpatch whether to run in test mode.
Values	y or n
Default	y, meaning that adpatch does not run in test mode
Example	adpatch apply=n

order	
Purpose	Tells adpatch how to organize its list of jobs to run in a phase. This does not affect the order in which phases or sub-phases are run.
Values	forward or backward
Default	forward, meaning that adpatch organizes its list of jobs to run in a phase in the order they are listed in the patch driver file.
Example	adpatch order=backward

patchtop	
Purpose	Tells adpatch the top-level directory for the current patch. This is normally used in non-interactive mode.
Values	A fully-qualified directory name.
Default	None, meaning that adpatch prompts for the patch directory.
Example	adpatch patchtop=/d1/apps/patches/2344175

driver	
Purpose	Tells adpatch the name of the patch driver file. This is usually used in non-interactive mode. It is only valid when the patchtop option is also used. Note that this option is not recommended and may be desupported in future releases.
Values	A driver file name, or comma-separated list of patch driver file names.
Default	None, meaning that adpatch prompts for the patch driver file name.
Example	adpatch patchtop=/d1/apps/patches/2344175 driver=c2344175.drv

options	
Purpose	Generic options passed to adpatch. See the information that follows for more information on options that are used.
Values	A comma-separated list of options.
Default	None, meaning that no options have been passed.
Example	adpatch options=integrity

The options argument may be followed by a list of options. The valid options are:

AutoPatch Options	
check_exclusive	
Purpose	If the Concurrent Sessions feature is enabled, tells adpatch whether to print a warning message when another adpatch session is currently running against this APPL_TOP. Use check_exclusive to check for other running adpatch sessions.
Default	nocheck_exclusive
Comments	Use check_exclusive to avoid unintended attempts to run multiple patches on the same APPL_TOP simultaneously when the Concurrent Sessions feature is enabled.
checkfile	

AutoPatch Options	
Purpose	Tells adpatch to either skip running exec, SQL, and exectier commands if they are recorded as already run, or to record them as having run after running them. Use nocheckfile to turn off the checkfile feature.
Default	checkfile
Comments	checkfile provides significant performance benefits.
compiledb	
Purpose	Tells adpatch to automatically compile invalid objects in the database after running actions normally found in the database driver.
Default	compiledb for standard patches; nocompiledb for standard patch translations, documentation patches, and documentation patch translations.
Comments	In cases where multiple non-merged patches are applied in a maintenance window, you can use nocompiledb to save time. However, merging multiple patches and applying a single merged patch is usually a better strategy.
compilejsp	
Purpose	Tells adpatch whether to automatically compile out-of-date JSP files. JSP files are only compiled if the patch contains copy actions for at least one JSP file.
Default	compilejsp for standard patches; nocompilejsp for standard patch translations, documentation patches, and documentation patch translations.
Comments	In cases where multiple non-merged patches are applied in a maintenance window, you can use nocompilejsp to save time. However, merging multiple patches and applying a single merged patch is usually a better strategy.
copyportion	
Purpose	Tells adpatch whether to run commands normally found in a copy driver. Use nocopyportion to tell adpatch not to perform copy driver actions.
Default	copyportion
Comments	Useful mostly with unified drivers.
databaseportion	

AutoPatch Options	
Purpose	Tells adpatch whether to run commands normally found in a database driver. Use nodatabaseportion to tell adpatch not to perform database driver actions.
Default	databaseportion
Comments	Useful mostly with unified drivers.
generateportion	
Purpose	Tells adpatch whether to run commands normally found in a generate driver. Use nogenerateportion to tell adpatch not to perform generate driver actions.
Default	generateportion
Comments	Useful mostly with unified drivers.
integrity	
Purpose	Tells adpatch whether to perform patch integrity checking, which verifies that the version of each file referenced in a copy action matches the version present in the patch.
Default	nointegrity
Comments	As the integrity of Oracle Applications patches is checked before they are released, the default of nointegrity is safe and avoids some adpatch overhead.
maintainmrc	
Purpose	Tells adpatch whether to automatically maintain the MRC schema after running actions that update the database. The MRC schema is only maintained if you have converted to MRC.
Default	maintainmrc for standard patches; nomaintainmrc for standard patch translations, documentation patches, and documentation patch translations.
Comments	When multiple non-merged patches are applied sequentially, you can use nomaintainmrc to save time when applying the initial patches, then maintainmrc only when applying the final patch. Merging multiple patches and applying a single merged patch, however, is a better performance strategy.
parallel	
Purpose	Tells adpatch whether to run actions that update the database in parallel (like sql) and actions that generate files in parallel (like genform). Use noparallel to tell adpatch to run actions serially.

AutoPatch Options	
Default	parallel
Comments	We do not recommend using noparallel. Oracle Applications patches are tested using parallel mode, and this option may be desupported in a later Oracle Applications release.
prereq	
Purpose	Tells adpatch whether to perform prerequisite patch checking prior to running patch driver files that contain actions normally found in the copy driver. Use noprereq to turn off prerequisite patch checking.
Default	prereq
Comments	We do not recommend using noprereq, as prereq prevents you from applying a patch without first applying all required prerequisite patches.
revcache	
Purpose	Tells adpatch whether to load the package revision cache (if adpatch determines it is needed). Use norevcache to tell adpatch not to load the package revision cache.
Default	revcache
Comments	With AD Mini-pack G, adpatch only loads the package revision cache when package commands without checkfile syntax are present in the patch driver file, there is not much reason to use norevcache. Specifying norevcache will cause all package commands without checkfile syntax to run.
validate	
Purpose	Tells adpatch whether to connect to all registered Oracle Applications schemas at the start of the patch.
Default	novalidate
Comments	Useful for finding problems with incorrectly-registered Oracle Applications schemas or schemas with invalid passwords.

AD Merge Patch (admrgpch)

Each time AutoPatch starts, it prompts for information and attempts to connect to the Oracle Applications system. In addition, there may be duplicate link, generate, and database tasks in a collection of patches. When patches are applied individually, these tasks are performed multiple times. Applying a patch driver that

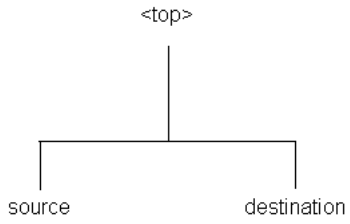
contains integrated patches reduces the patch application time by eliminating redundant system verification and duplicate patch tasks.

AD Merge Patch is an executable located in AD_TOP/bin that merges multiple AutoPatch compatible patches into a single, integrated patch. The command for merging patches is admrgpch. See Merging Patches in *Oracle Applications Maintenance Procedures* for complete instructions.

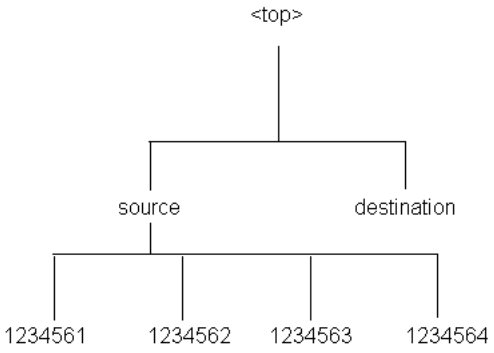
AD Merge Patch reads the patch driver files for each patch in the source directory and merges them to create patch driver files in the destination directory. AD Merge Patch supports both split patch driver files (c<patchnum>.drv, d<patchnum>.drv, and g<patchnum>.drv), and unified patch driver files (u<patchnum>.drv). If all the source patches have split driver files, the merged patch has split driver files. If any of the source patches has a unified driver file, the merged patch has a unified driver file.

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 exists in more than one source patch, only the highest revision of the file is copied to the destination directory.

The source and the destination directories cannot be child or parent directories of each other. We recommend that you run AD Merge Patch from the parent directory of the source directory, and that the destination directory also be located in the same parent directory. For example, if you run AD Merge Patch a directory named <top>, both the source and destination directories should be subdirectories of <top>.



The source directory must have all patches to be merged as immediate child directories. The patch directories cannot be in a lower directory under the source directory. For example, if four patches are to be merged, they must be in a directory structure similar to the following:



To run AD Merge Patch, log in as applmgr and set the environment as described in Setting the Environment in Chapter 1. From the <top> directory, run AD Merge Patch with the following command:

```
admrgpch -s <source directory> -d <destination directory> \
-merge_name <name>
```

The source directory where the patches to be merged have been unloaded is <source directory>; the destination directory where the integrated patch will be created is <destination directory>; and the name of the integrated patch is <name>.

> **Note:** The -merged_name parameter is optional. If you do not provide this parameter, the name of the merged patch will be "merged".

For example, when merging four patches called 1234561, 1234562, 1234563, and 1234564 located in the source directory /d01/patch_merge/source to the destination directory /d01/patch_merge/destination, and using "merge99" for the name of the integrated patch, use the following commands:

UNIX:

```
$ cd /d01/patch_merge
$ admrgpch -s source -d destination -merged_name merge99
```

Windows:

```
C:\> CD \d01\patch_merge
C:\> admrgpch -s source -d destination -merged_name merge99
```

5

Upgrading

You run AutoUpgrade to upgrade Oracle Applications products from an earlier release to the base version of the new release. This chapter describes how to run AutoUpgrade. It contains these sections:

AutoUpgrade

A complete upgrade of an Oracle Applications installation requires several AD utilities, including Rapid Install, AutoUpgrade, and AutoPatch. Once you complete the pre-upgrade tasks (including running Rapid Install to create the new file systems and install the required technology stack components), you run AutoUpgrade to upgrade products from an earlier release of Oracle Applications to the base version of the latest release. You may also need to run AutoPatch to apply patches associated with the upgrade.

> **Additional Information:** See Overview of an Upgrade in *Upgrading Oracle Applications.*

Starting and Stopping

Before starting AutoUpgrade, complete the steps in the Setting the Environment section of Chapter 1. Then, start AutoUpgrade from any directory by typing *adaimgr* at the command prompt. The utility starts and displays the first prompt in the series of questions it asks about the upgrade.

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

You can stop the upgrade process before it is complete by typing *abort* at any prompt. If you take this action, AutoUpgrade saves the operations it has performed to that point in restart files. It uses these files to restart the upgrade from where the last session ended. If an error occurs while you are upgrading the database, you can correct the error without stopping AutoUpgrade.

Additional Information: See Restarting AutoUpgrade in this chapter.

Responding to Prompts

The first action AutoUpgrade takes is to ask a series of questions about your installation and about the upgrade process itself. In addition to general questions about your installation, it asks the following questions that are specific to the upgrade process.

Software compatibility

AutoUpgrade prompts you to verify that your installation is made up of a certified combination of components:

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

If you are uncertain whether a release combination is certified, see Technology Stack Components in *Installing Oracle Applications* and the Certify web page for the latest certification information. Access Certify from Oracle*MetaLink* (Product Lifecycle > Certifications).

Additional Information: See Frequently Asked Questions about Certify, Oracle*MetaLink* Doc ID: 119139.1.

Applications System name

The Applications System name is set by Rapid Install during the pre-upgrade process. The default is <SID>. The following prompt appears when AutoUpgrade starts running. We recommend that you do not change the default system name.

```
Please enter the name of the Oracle Applications System that this APPL_TOP
```

```
belongs to.

The Applications System name must be unique across all Oracle
Applications Systems at your site, must be from 1 to 30 characters
long, may only contain alphanumeric and underscore characters,
and must start with a letter.

Sample Applications System names are: "prod", "test", and "demo" and
Development_2.

Applications System Name [prod] :
```

Files installed in APPL_TOP

The server configuration information is set by Rapid Install during the pre-upgrade process. The configuration determines the types of files that were unloaded by Rapid Install on a given node (such as Java files, HTML files, forms files, and concurrent programs files). AutoUpgrade prompts for answers pertaining to server configuration. You should not change the defaults unless instructed by Oracle Support Services.

```
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 system 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. We recommend that you do not change the APPL_TOP name.

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

The APPL_TOP name you select must be unique within an Oracle Applications
System, must be from 1 to 30 characters long, may only contain.
alphanumeric and underscore characters, and must start with a letter.

APPL_TOP Name [prod_all] :

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

AutoUpgrade log file name

The AutoUpgrade log file is where the AutoUpgrade manager stores information about actions taken during the upgrade. Enter a new name or accept the default (adaimgr.log). The file is located in $APPL_TOP/admin/<SID>/log, where <SID> is the value of the ORACLE_SID or TWO_TASK variable (UNIX), or in %APPL_TOP%\admin\<SID>\log, where <SID> is the value of the ORACLE_SID or LOCAL variable (Windows).

Identify your Organization Type

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

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

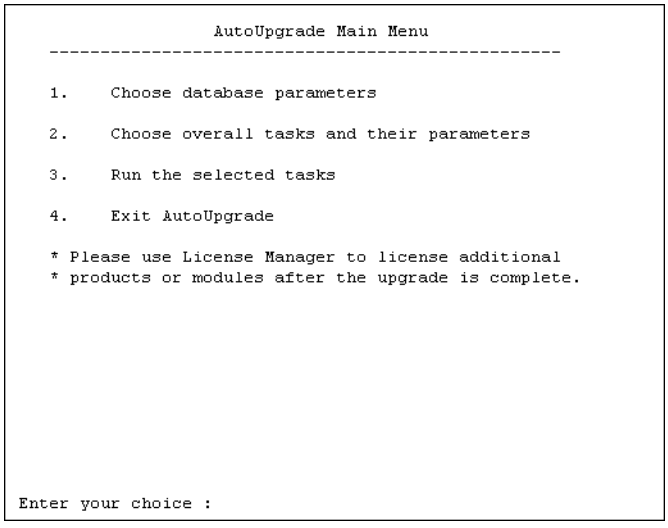
AutoUpgrade does not ask these questions if you have previously installed public sector, education, or not-for-profit products. Once not-for-profit products are installed, they *cannot* be converted to commercial products.

Running the Upgrade

After you supply answers to the initial AutoUpgrade prompts, the AutoUpgrade Main Menu appears.

Main Menu

The Main Menu presents the options you use to configure and run the upgrade. Use options 1 and 2 to configure the upgrade, then use option 3 to start the upgrade process.



Type the option number to choose an option. You usually choose the options in numeric order. You can choose to exit AutoUpgrade before you complete all the tasks on this menu. If you do, save the configuration information so you can run AutoUpgrade again without re-entering the information.

Choose Database Parameters

Type the option number for Choose Database Parameters. The Database Parameters screen appears.

```
                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  APPLSYSD   APPLSYSX   APPLSYSD
2 Application Utilities      APPLSYS     100  APPLSYSD   APPLSYSX   APPLSYSD
3 Applications DBA           APPLSYS     100  APPLSYSD   APPLSYSX   APPLSYSD
4 Oracle Alert               ALR         100  ALRD       ALRX       ALRD
5 Global Accounting Engi     AX          100  AXD        AXX        AXD
6 Oracle Common Modules-     AK          100  AKD        AKX        AKD
7 Oracle Common Accounti     XLA         100  XLAD       XLAX       XLAD
8 Oracle General Ledger      GL          100  GLD        GLX        GLD

     There are 179 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) :
```

Use this screen to:

- Change the default Oracle user ID (schema name) and password for each product.
- Set the sizing factor for new objects for a product (or for new products).
- Verify that tablespaces for existing products are set correctly.
- Specify the tablespaces for each new product you are licensing.

You can change a parameter for all products by entering A (All) instead of a product number, and then a letter, such as O, S, M, I, or D, to specify the parameter. For example, to change the sizing factor for all products, you would type *AS*.

The Action column indicates the action that AutoUpgrade will take during the upgrade. The action for all your currently licensed products is set to U (upgrade). A blank entry means that the product is not licensed. (Rapid Install installs all files for all products regardless of license status). If you plan to license additional products, use License Manager after the upgrade is complete. See Registering Applications Products in *Oracle Applications Maintenance Procedures*.

In the example, the screen lists eight Oracle Applications products, but indicates there is a total of 179 products. To display information about products 9 through 16, type D and press [Return] to scroll down the list. Type B and [Return] to scroll to the bottom of the list and display the last eight products.

Option O: Oracle User ID

This column lists the Oracle User ID (username/password) that owns each product's database objects. You cannot change the User ID for products that are currently installed and in use.

> **Attention:** We recommend you do not change the Oracle 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 after you run AutoUpgrade.

  > **Additional Information:** See Oracle Applications Schema Password Change Utility in *Oracle Applications System Administrator's Guide*

- 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 others.

To change the Oracle User ID, type <product number>O. AutoUpgrade prompts for the Oracle username. Accept the default or enter a new username (up to 30 characters). To change only the password, press [Return] when AutoUpgrade prompts for the username, and enter the new password at the next prompt.

Option S: Sizing Factor

This column shows the sizing factor that AutoUpgrade applies to new product tables and indexes. To change a sizing factor, type <product number>S 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.

> **Note:** Non-licensed products are installed with the sizing factor shown on the Database Parameters screen. We recommend that you leave the sizing factor at the default (100), so you can later license a new product and use it immediately. The 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 Oracle server tablespaces in which AutoUpgrade places product 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. 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. For example, GLD and GLX for the GL 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 you return 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 (All Main) or AI (All Index) will not change the tablespaces for products that are already installed in the database. You can change the tablespaces for these products by entering the product number explicitly, such as <product number>M or <product number>I.

When you are finished with this screen, press [Enter] at the prompt to return to the AutoUpgrade Main Menu.

Choose Overall Tasks and their Parameters

From the Main Menu, enter the option number for Choose Overall Tasks and their Parameters. The Tasks screen appears. This screen displays the tasks that

AutoUpgrade will perform during the upgrade processing. By default, AutoUpgrade performs all these tasks.

```
          AutoUpgrade - Choose overall tasks and their parameters

 # Task                                        Do it?   Parameters
-- ------------------------------------------  ------   --------------------
 1 Create Applications environment file         YES     vis115p.env
 2 Verify files necessary for install/upgrade   YES
 3 Install or upgrade database objects          YES


      There are 3 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) :
```

Review the options on this screen, and (if necessary) modify the default environment file name. To return to the Main Menu, press [Return] at the prompt on this screen. Descriptions of the options are as follows:

Create Applications environment file (Option 1)

During the upgrade, AutoUpgrade creates an environment file that defines Oracle Applications environment variables. The default name for the environment file is <SID>.env, where <SID> is the value of the ORACLE_SID or TWO_TASK variable (UNIX); or <SID>.cmd, where <SID> is the value of ORACLE_SID or LOCAL (Windows). The environment file name is listed in the Parameters column. To change the environment file name, first type the option number at the prompt and then enter a new file name. The name can contain up to 30 characters. Press [Return] to accept the default name.

Verify files necessary for install/upgrade (Option 2)

When you choose this option, AutoUpgrade verifies that all files necessary for the upgrade are present. You may want to run this task as a pre-upgrade step to identify missing files before you begin the upgrade process.

> **Additional Information:** See Overview of an Upgrade in *Upgrading Oracle Applications*.

Install or upgrade database objects (Option 3)

This option verifies files, upgrades database objects for existing product groups, and installs new database objects. Running this task always verifies that all required files are present.

Run the Selected Tasks

After you have configured the upgrade with the first two tasks on the Main Menu, type the option number for Run the Selected Tasks to start the upgrade. AutoUpgrade performs the tasks listed on the Tasks screen, and prompts for information based on those tasks. During the process, it takes these actions:

Environment File

Asks a series of questions related to the configuration stored in the environment file. For example, it asks questions about parallel concurrent processing, file-naming conventions, directory paths for log, output, and temporary files, and so on.

> **See Also:** Create Applications environment file (UNIX) and Create Applications environment subkey in registry (Windows) in Chapter 3, "Maintenance".

Number of Workers

Launches multiple worker processes to perform the upgrade processes in parallel. It automatically determines the default value for the number of workers by adding 2 to the number of CPUs on the machine where the database server is running. For example, the default is 3 on single-processor machines.

AutoUpgrade prompts for the number of workers that you want to use. Accept the default or enter a different value — up to 99 workers.

> **Note:** It may be inefficient to have more than two workers over the number of CPUs on the database server because the CPUs may waste time switching between the processes.

Verifying Files

As AutoUpgrade verifies files, it 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] :
```

The log file (adiuvf.lst) is stored in $APPL_TOP/admin/<SID>/out (UNIX), or %APPL_TOP\admin\<SID>\out (Windows). You can accept the default name or enter a new one at this prompt.

If AutoUpgrade finds that any files are missing, it displays a failure message and stops. It lists the missing files in the adaimgr.log file. Review the missing files, correct the problems, and restart AutoUpgrade. See Restarting AutoUpgrade in this chapter.

Installing and upgrading database objects

AutoUpgrade upgrades all previously installed and licensed products and installs database objects for all other products in Release 11i. After AutoUpgrade completes the last task, it returns to the Main Menu.

Exit AutoUpgrade

AutoUpgrade has run all the tasks to completion when the Main Menu appears again on the screen. To end the AutoUpgrade session, type the option number for Exit AutoUpgrade. AutoUpgrade exits, and this portion of the upgrade is complete. The next step is to perform the post-upgrade tasks as described in *Upgrading Oracle Applications*.

You can also choose the Exit AutoUpgrade option any time you have access to the Main Menu, for example, after AutoUpgrade completes all the actions for one of the tasks you selected on the Main Menu, or if you press [Return] at the prompt on one of the AutoUpgrade screens.

Any time you choose the Exit AutoUpgrade option, the utility exits and saves the actions taken up to that point in restart files. When you restart AutoUpgrade, choose Yes at the following prompt to restart from the point where you exited:

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

Acceptable Errors

If you encounter a failure in the worker log file 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 job with AD Controller. Contact Oracle Support Services if the worker encounters the same error while running the job again.

Additional Information: See AD Controller (adctrl) in Chapter 1.

Oracle Server Error Messages

The following Oracle server error messages indicate acceptable problems and do not require any action:

```
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
```

If you consistently receive ORA-1555 errors, the problem may be due to insufficient rollback space. Try increasing the available rollback space before restarting the workers.

Additional Information: See Managing Rollback Segments in *Oracle Administrator's Guide*.

Database Object Differences

Several different types of warnings may be reported when AutoUpgrade compares database objects in the upgrade to those from the previous release:

```
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, review these warnings after the upgrade to determine whether you need to modify your customizations.

Import Messages

AutoUpgrade may run the Oracle Import utility (IMP) when upgrading Oracle Applications. The following IMP 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

DataMerge (addmimp) runs during an upgrade to import seed data needed by Oracle Applications. A message like the following 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 error may not occur again if you immediately restart AutoUpgrade or the AutoUpgrade worker. Persistent DataMerge errors 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.

Correcting Worker Errors

When a worker fails its job, you can attempt to fix the failed job while the other workers and the manager are still running. Use the worker log files to determine the problem and restart the worker while the manager is running.

Note: Changes made to environment variables while AutoUpgrade is running do not take effect until all workers are restarted.

Additional Information: See Troubleshooting in the *Oracle Applications Maintenance Procedures* for information on handling a failed worker. See also AD Controller (adctrl) in Chapter 1.

Restarting AutoUpgrade

If AutoUpgrade failed, or if you aborted it, you must run it again. Make sure that the environment is set up properly before you restart.

Additional Information: See Setting the Environment 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 the log file, AutoUpgrade adds a "**Start of AutoUpgrade Session**" message to the end of the file and appends messages from the new session as it generates them.

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

The default is to continue the previous session. Press [Return] to choose this action. AutoUpgrade determines where the last session stopped and restarts at that point. It retains all configuration information you entered in the last session.

Attention: If the machine failed while AutoUpgrade was running the upgrade, the AutoUpgrade restart files may have been corrupted.

Start New Session

If you enter No at the Continue prompt, AutoUpgrade asks you to confirm that you do not want to continue the previous session. It then restarts from the beginning. You must re-enter any previous configuration information.

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 the previous session. Alternatively, you can restore the last saved database and file system, then start AutoUpgrade again from the beginning.**

6

Reports

This chapter describes the patch history database, how it is enabled, the patch history interface, and the reports that show patch history information. This information assumes familiarity with DBA tasks and the AutoPatch utility.

This chapter contains the following sections:

The Patch History Database

The patch history database and interface do not change the process of applying a patch. You still apply patches with AutoPatch, but after each driver is run on each node, AutoPatch uploads the patch information to the Oracle Applications database. After applying patches, you use OAM to view the patch history information.

The database stores the following key information for all patches:

- Patch number
- Patch type (such as mini-pack or maintenance pack)
- Driver file name

- Type of driver file (copy, database, or generate)
- Platform
- APPL_TOP on which the patch was applied
- Contents of the patch
- Language of the patch
- Servers on which the patch was applied
- Bug fixes included in the patch
- Whether the fix was applied successfully
- Reason a bug fix was not applied, if any

If the patch is a maintenance pack, the database also stores the maintenance pack level.

Patch History Database and AutoPatch Modes

AutoPatch stores the patch information in the database in different ways, depending on which AutoPatch mode you use when applying the patch. If the patch is not successful, patch information is not written to the database or to the patch history file.

Additional Information: See AutoPatch Modes in Chapter 4.

Normal mode

When you apply a patch, information about the patch is written to a *patch history file* in the APPL_TOP/admin/<SID> directory. In prior releases, the patch history file was named applptch.txt. In Release 11.5.9, there are two patch history files in this directory: adpsvYYYYMMDDhhmiss.txt and javaupdatesYYYYMMDDhhmiss.txt. (Note that in these filenames, hh is in 24-hour time format.) The javaupdates file contains the information on changes to Java files, and the adpsv file contains information on changes to all other files.

AutoPatch applies the patch, and then writes the patch history information to the database, appending it to any existing patch history information.

If there is an existing patch history file (applptch.txt, adpsv<date>.txt, or javaupdates<date>.txt file) when applying a patch in normal mode, the information from the patch history file is uploaded to the database. The patch history files are then deleted if the upload is successful. If the upload is not successful, the files are not deleted. The AutoPatch log file records whether the upload was successful.

Test mode

The patch history file is not changed. AutoPatch does not upload any patch history files and does not upload patch history information to the database.

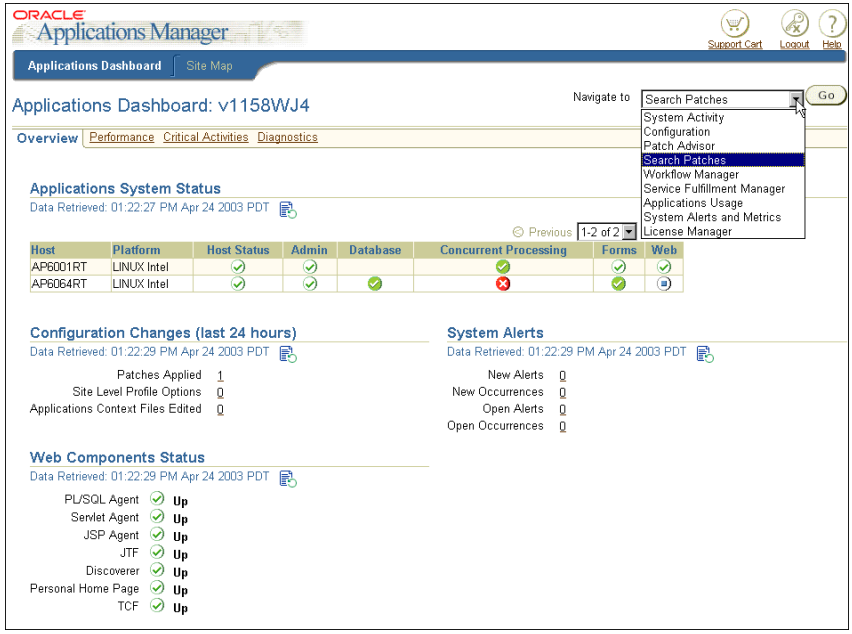
Pre-install mode

After applying the patch, AutoPatch writes patch history information to a patch history file and uploads the contents of this file to the database the next time it runs in normal mode.

> **Note:** Whether patch history information is written to the database depends on which mode (normal, test, or pre-install) you are running. Interactive and non-interactive mode do not affect whether patch history information is written.

Patch Search

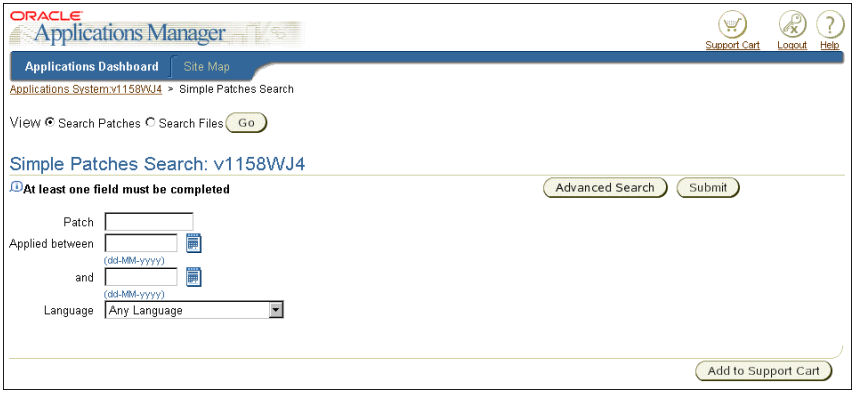
You can navigate to the Search Patches main screen from the Oracle Applications Manager Dashboard.



You can navigate to the Search Patches main screen from the Oracle Applications Manager Site Map.

Additional Information: Oracle Applications Manager in *Oracle Applications System Administrator's Guide* describes the layout of the OAM screens.

The first screen is the Simple Patches Search screen:



Use the View option buttons and the Go button at the top of the screen to toggle between the Search Patches and the Search Files screens. Click the Advanced Search button to go the Advanced Patch Search screen.

To perform a simple patch search, complete at least one of the following fields and then click Submit:

- Patch

  Enter the patch number.

- Applied between <begin date> and <end date>

  Search for patches that were applied during a specified period of time. Click the calendar icon to choose date, or enter the date directly in the field. For example, you can enter only a begin date to see all patches applied from that date through today's date. You can enter only an end date to see all patches applied up to that date. Or enter the begin date and the end date to see all patches applied between the two dates.

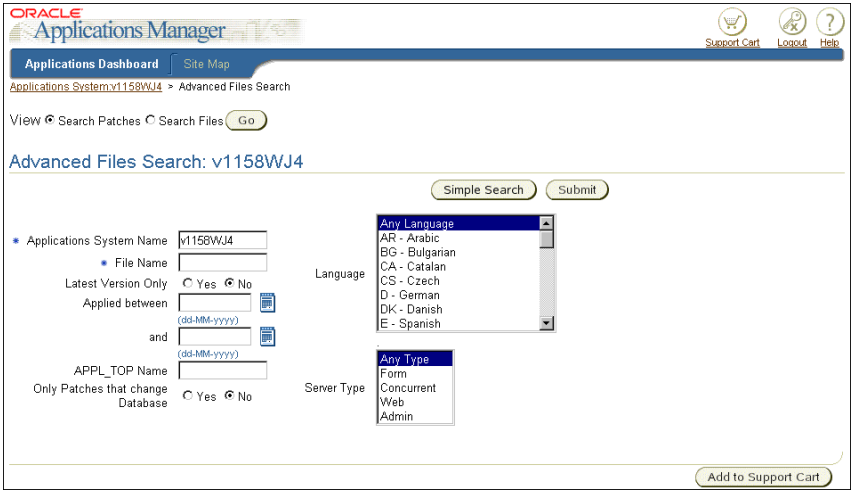
- Language

  You can choose only one language in this field. Use the Advanced Patches Search screen to choose multiple languages.

Advanced Patches Search

Click the Advanced Search button in the Simple Patches Search screen to see the Advanced Patches Search screen. From this screen you can:

- Search for applied patches.
- Access the Simple Patches Search screen.
- Access the Simple Files Search screen.
- Access patch history migrated from another Applications system.



In addition to the fields on the Simple Patches Search screen, the Advanced Patches Search screen has further criteria to narrow the results of a query:

- Applications System Name (required)

  Defaults to the name of your Applications system. If you have migrated patch history information from another system, and want to search those records, enter the name of that system now.

  > **Additional Information:** See Exporting Patch History Information in *Oracle Applications Maintenance Procedures*.

- Patch

  Enter the patch number.

- Product

  Enter the short name of the product that owns the patch, such as AD or GL. This field is not case-sensitive.

- Applied between <begin date> and <end date>

  Search for patches that were applied during a specified period of time. Click the calendar icon to choose the date, or enter the date directly in the field. For example, you can enter only a begin date to see all patches applied from that date through today's date. You can enter only an end date to see all patches

applied up to that date. Or enter the begin date and the end date to see all patches applied between the two dates.

- APPL_TOP name

  Enter the name of the APPL_TOP where the patches were applied.

- Only Patches that change Database

  Choose Yes to search for patches that have database drivers, and No to search for patches that do not.

- Language

  Use the drop-down menu to limit the query to patches for a specific language. Unlike the Simple Files Search screen, you can choose more than one language in this field. Hold down the Ctrl key and click any number of languages from the list.

- Server Type

  Choose the types of server implemented on the APPL_TOP where the patches were applied. The options are Any Type, Forms server, Concurrent processing server, Web server, and Admin server. You can choose one or more servers. Hold down the Ctrl key and click the servers on the list. Any Type returns patch information for all servers.

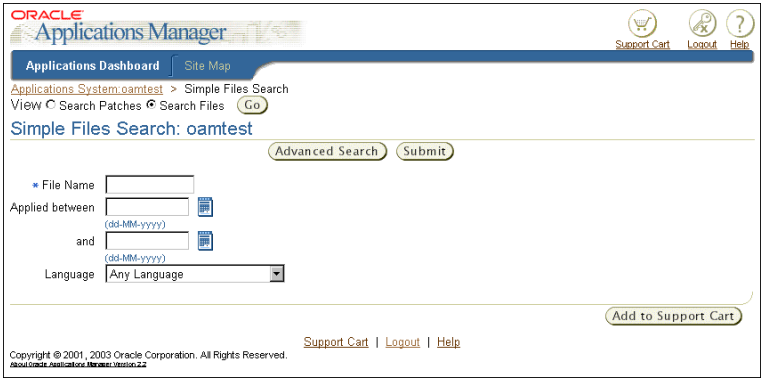
You must enter a value in at least one of these fields. Click Submit.

File Search

Click the Search Files button on the Simple Patches Search screen or the Advanced Patches Search screen to see the Simple Files Search screen. You can also access the Simple Files Search screen from the OAM Site Map.

From the Simple Files Search screen you can:

- Search for files that have been patched.
- Access the Simple Patches Search screen.
- Access the Advanced Files Search screen.



The Simple Files Search screen has the following input fields:

- File name (required)

  Enter a file name, but not a directory path. This field is case-sensitive.

- Applied between <begin date> and <end date>

  Search for files that were updated during a specified period of time. Click the calendar icon to choose the date, or enter the date directly in the field. For example, you can enter only a begin date to see all files updated from that date through today's date. You can enter only an end date to see all files up to that date. Or enter the begin date and the end date to see all files updated between the two dates.

- Language

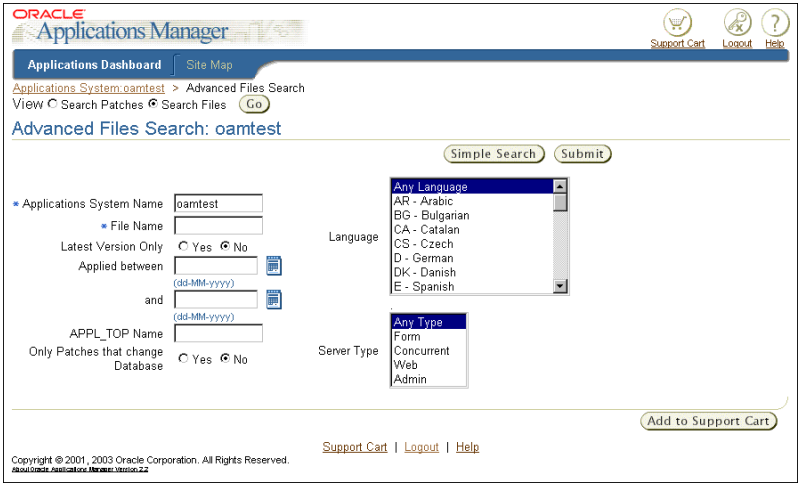
  You can choose only one language in this field. Use the Advanced Patches Search screen to choose multiple languages.

Enter a value in File name field — the other fields are optional. Click Submit.

Advanced Files Search

Click the Advanced Search button on the Simple Files Search screen to see the Advanced Files Search screen. From this screen you can:

- Search for files that have been patched.
- Access the Simple Files Search screen.
- Access the Simple Patches Search screen.
- Access the information migrated from another Applications system.



In addition to the fields on the Simple Files Search screen, the Advanced Files Search screen has further criteria to narrow the results of a query:

- Applications System Name (required)

  Defaults to the name of your Applications system. If you have migrated patch history information from another system, and want to search those records, enter the name of that system now.

- File name (required)

  Enter a file name, but not a directory path. This field is case-sensitive.

- Latest Version Only

  Choose Yes to see information on the latest version of the file only. Choose No to see information for all versions of the selected file.

- Applied between <begin date> and <end date>

  Search for patches that were applied during a specified period of time. Click the calendar icon to choose the date, or enter the date directly in the field. For example, you can enter only a begin date to see all patches applied from that date through today's date. You can enter only an end date to see all patches applied up to that date. Or enter the begin date and the end date to see all patches applied between the two dates.

- APPL_TOP name

  Enter the name of the APPL_TOP that contains the files.

- Only Patches that change Database

  Choose Yes to search for files contained in patches that have database drivers. Choose No to search for patches that do not have database drivers.

- Language

  Use the drop-down menu to limit the query to patches for a specific language. Unlike the Simple Files Search screen, you can choose more than one language in this field. Hold down the Ctrl key and click any number of languages from the list.

- Server Type

  Choose the types of server implemented on the APPL_TOP where the patches were applied. The options are Any Type, Forms server, Concurrent processing server, Web server, and Admin server. You can choose one or more servers. Hold down the Ctrl key while choosing two or more servers from the list. Any Type returns patch information for all servers.

Enter the Applications System Name and a value in File name field — the other fields are optional. Click Submit.

Patch History Reports

When you search the patch history database using any search method, the results appear on screen as a report. You can see the following reports after a successful search:

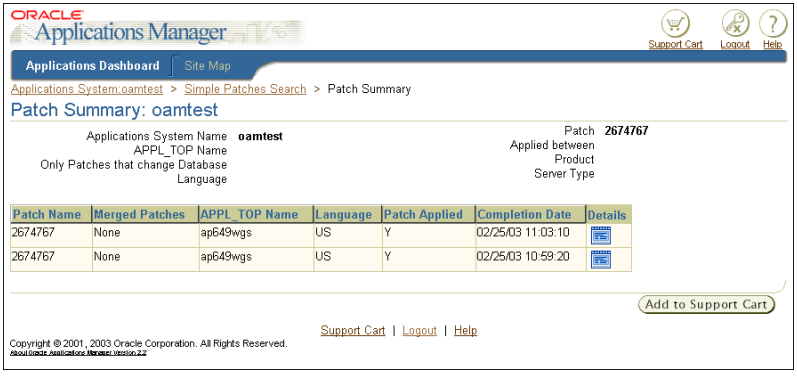
- Patch Summary
- Patch Details
- Files Copied
- Bug Fixes
- Action Summary
- File History

The top portion of each report displays the search criteria, and the bottom portion displays the results of the search. The navigation path at the top of the report is a link. Click the link to see the respective report.

Click the heading of a column to sort the information based on the content in that column. The sortable columns have column headings that appear 3-dimensional. The sort alternates between ascending and descending each time you click the column heading.

Patch Summary Report

The Patch Summary report displays the results of a query from either the Simple Patches Search screen or the Advanced Patches Search screen. Each line item represents an applied patch, and each page can contain up to 25 line items. If the Patch Summary report is more than one page long, use the Previous and Next links or the drop-down list to see other pages.



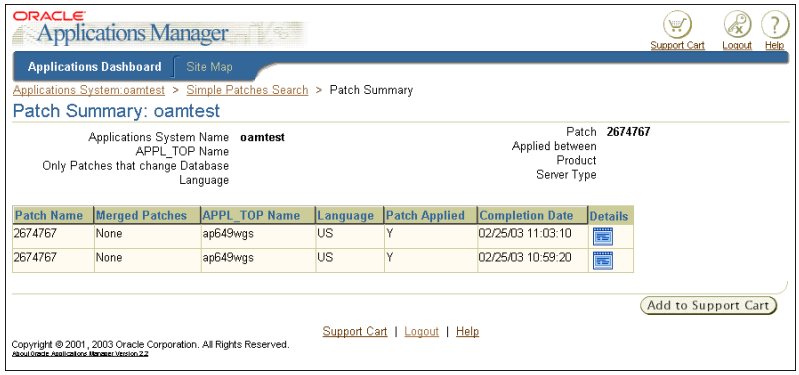
The details provided for each patch are:

Term	Definition
Patch Name	The patch number.
Merged Patches	If this is a merged patch, this field contains a comma-separated list of all patch numbers contained in the merged patch.
APPL_TOP Name	The APPL_TOP where the patches were applied.
Language	The language of the patch.
Patch Applied	Y indicates that the patch was applied successfully and N that it was not.
Completion Date	The date and time the patch was completed.
Details	Clicking this link accesses the Patch Details report.

If there are no patches matching the query criteria, the Patch Name column contains the message "The above criteria resulted in no rows".

Patch Details Report

The Patch Details report provides details for a specific patch. Click the Details link in a selected row of the Patch Summary report to open the Patch Details report. The patch summary information from the Patch Summary report appears at the top of the Patch Details report.



This report contains the following information:

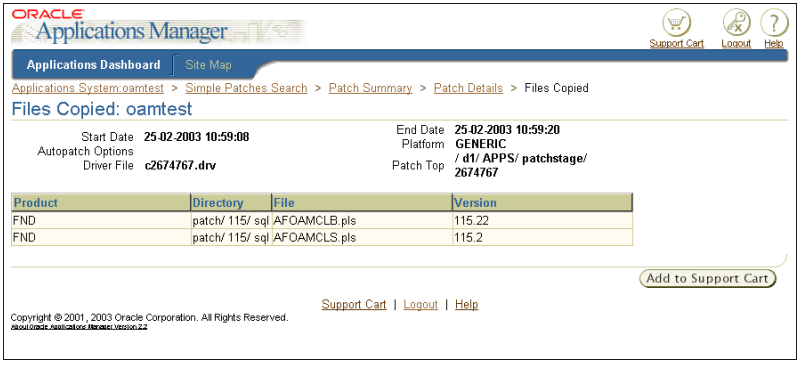
Term	Definition
Select	Determines which driver file details are presented in the Files Copied report, the Bug Fixes report, or the Action Summary report.
Driver File	The name of the AutoPatch driver file.
Start Date	The date and time the application of the driver file began.
End Date	The date and time the application of the driver file completed.
AutoPatch Options	Any AutoPatch command line options used to run the driver file.
Platform	The platform of the driver file.
Patch Top	The location of the driver file when it was run.

To see additional details for a patch, click on one of the following buttons:

Term	Definition
Files Copied	Shows Files Copied report.
Bug Fixes	Shows the Bug Fixes report.
Action Summary	Shows the Action Summary report.

Files Copied Report

This report lists all files copied to the file system as a result of the actions in the selected driver file. Each line item in this report represents a copied file. Choose a driver file in the Patch Details report and click the Files Copied button to see the Files Copied report.



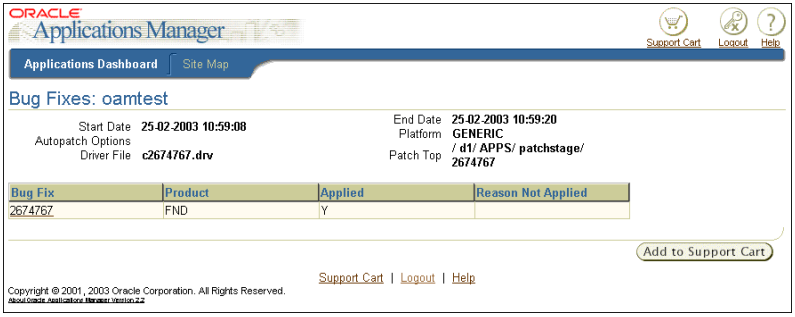
This report provides the following information about the files copied:

Term	Definition
Product	The short name for the product that owns the file.
Directory	The directory path where the file was copied.
File	The name of the file.
Version	The version number of the copied file.

If there are no files copied in the patch, no rows are displayed.

Bug Fixes Report

This report lists all bug fixes included in the selected driver file. Each line item represents a bug fix. Choose a driver file in the Patch Details report and click the Bug Fixes button to see the Bug Fixes report.

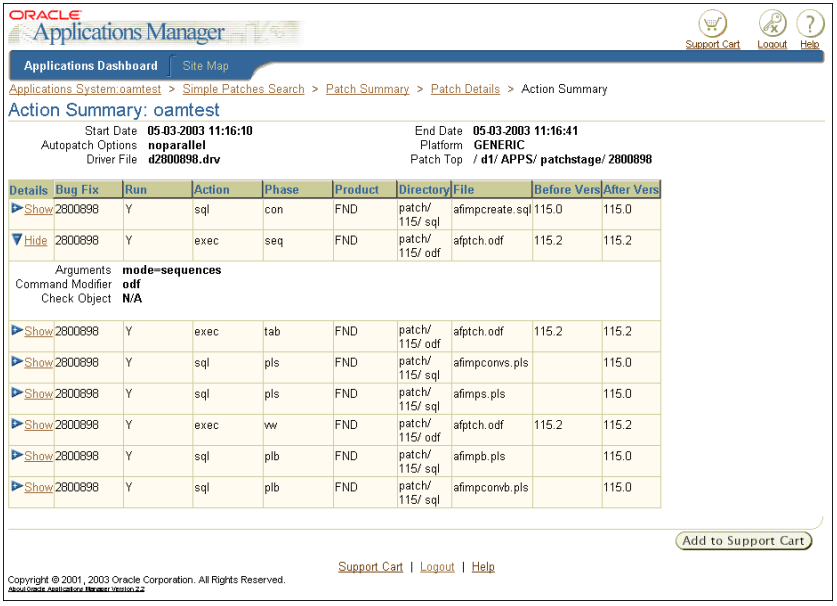


This report provides the following information about bug fixes:

Term	Definition
Bug Fix	The bug number of the bug fixed as a result of the selected driver file. The items in this column are links. Clicking an item accesses the Action Summary report.
Product	The product short name for the product whose bug was fixed.
Applied	Whether the bug fix was applied. If a bug fix was applied, you can click the bug number to see the Action Summary Report.
Reason Not Applied	If the bug fix was not applied, the reason is given here.

Action Summary Report

This report provides summary information for the actions of a selected driver file. Each line item represents a performed action.You can view the Action Summary report by either selecting a driver file in the Patch Details report and clicking the Action Summary button, or by clicking a bug fix number in the Bug Fix column of the Bug Fixes report.



The Action Summary report provides the following summary information:

Term	Definition
Details	Toggle between Show and Hide. Show expands the action item and provides more details about the action. Hide contracts and hides the action details.
Bug Fix	The bug number of the bug fixed as a result of the selected driver file.
Run	Whether the action was executed.
Action	The type of action performed on the patched file.
Phase	The phase in which the action occurred, if any.
Product	The product short name for the product that owns the file referenced by the action.
Directory	The directory path for the file referenced by the action.
File	The name of the file referenced by the action.
Before Vers	The version of the file before the patch.

Term	Definition
After Vers	The version of the file after the patch.

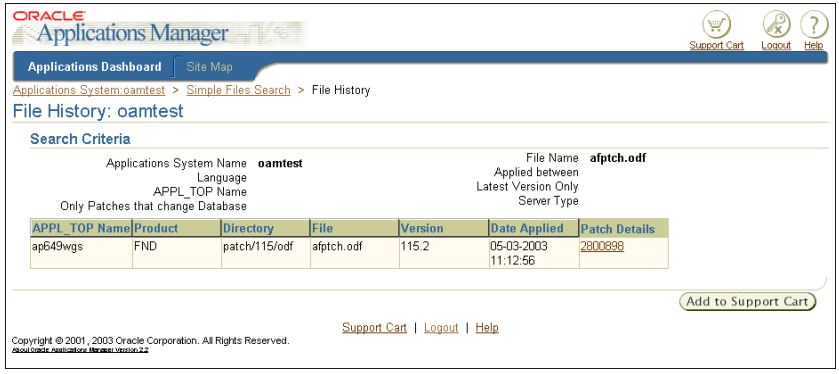
Action Details Clicking the Show link displays more detailed information about each action:

Term	Definition
Arguments	Extra arguments for SQL and EXEC commands.
Command Modifier	What type of SQL or EXEC command was run.
Check Object Name	The name of the database object to check for, along with name and password of the schema where AutoPatch looks for the check object. (This is "none none none" for most SQL commands and is not specified for EXEC commands.)

N/A in the report represents any action details that are not specified. For example, N/A in the Arguments field means no additional arguments were specified.

File History Report

The File History report displays the results of a query submitted through either the Simple Files Search screen or the Advanced Files Search screen. The query criteria entered in the search screens are at the top of the File History report. Each line item represents the file changed by a patch.



The details provided for a file are:

Term	Definition
APPL_TOP Name	The APPL_TOP containing the files.
Product	The product short name for the product that owns the file.
Directory	The directory path where the file is located.
File	The name of the file.
Version	The version number of the file.
Date Applied	The date this version of the file was applied by a patch.
Patch Details	Click on the patch number to see the details report for the patch that applied this version of the file.

If no files were patched, "The above criteria resulted in no rows" appears in the APPL_TOP Name column.

Patch Advisor

Search Patches provides screens to search and report on patches that have been applied to your system. Patch Advisor provides screens to determine patches that have not been applied to your system, but that should be applied to keep the system current. It does this by comparing the patches you have already applied against a list of all recommended Oracle Applications patches. This list of recommended patches is in a *patch information bundle file* that you download from Oracle*MetaLink* before running the Patch Advisor.

The Patch Advisor does not report on all available patches, but only on the patches that provide pro-active maintenance. The set of recommended patches includes maintenance pack, family packs, mini-packs, and high-priority patches.

Additional Information: See Patches in Chapter 4.

Before running Patch Advisor, you download a ZIP file called InfoBundle11i.zip from Oracle*MetaLink* and put it in the patch advisor staging directory. To download the InfoBundle11i.zip file, log into Oracle*MetaLink* and navigate to http://updates.oracle.com/download/InfoBundle11i.zip.

This patch information bundle zip file is updated daily and contains the list of recommended patches as well as metadata for these patches. The patch information bundle metadata contain the readme and LDT file for each recommended patch.

The patch metadata LDT files are FNDLOAD data files included in the top-level directory of all recent patches. The LDT files contain pre-requisite patch information and a manifest of all files in the patch with their version numbers. The patch information bundle metadata also include information about the relationships between patches, such as which mini-packs are contained in the recommended maintenance pack.

The Patch Advisor loads the patch information bundle data, including LDT files and readme files, into the Oracle Applications database. The Patch Advisor can analyze multiple requests against the metadata. For example, you can narrow the comparison to report only on recommended Human Resources patches, or to report only on high-priority patches.

Patch Advisor consists of a user interface, several concurrent programs, and several database tables. You use the interface to set up the Patch Advisor staging directory, manage patch filters, submit concurrent requests, and view recommended patches. The concurrent programs perform and monitor the following tasks:

- Upload patch information from Infobundle11i.zip to Patch Advisor tables

  Patch Advisor loads the patch information bundle metadata, including LDT files and readme files, into the Oracle Applications database.

- Recommend patches based on the current environment and patch information bundle

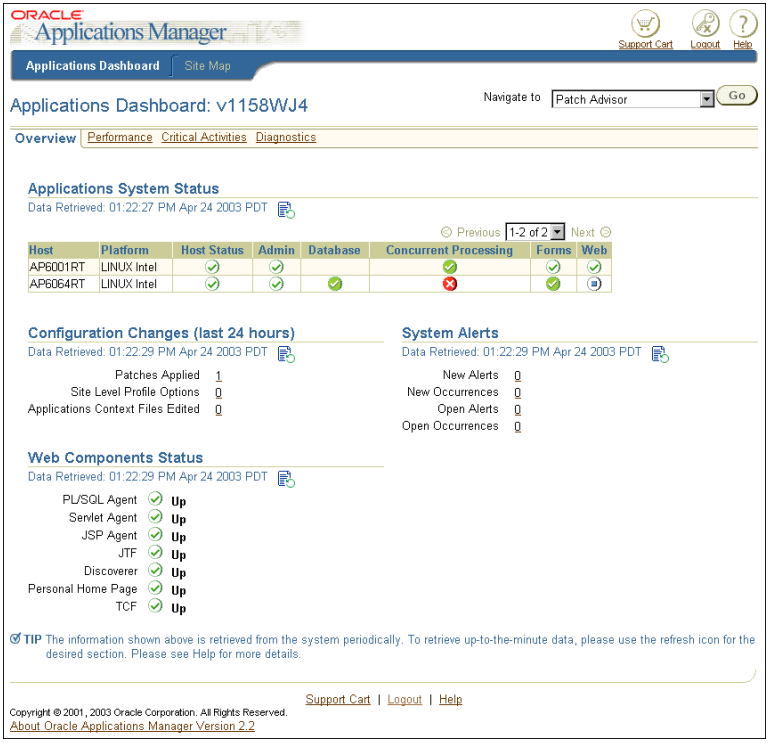
  Patch Advisor can analyze multiple requests against the metadata. For example, you can narrow the comparison to report only on recommended Human Resources patches, or to report only on high-priority patches.

- Analyze lists of patches after downloading them from Oracle*MetaLink*

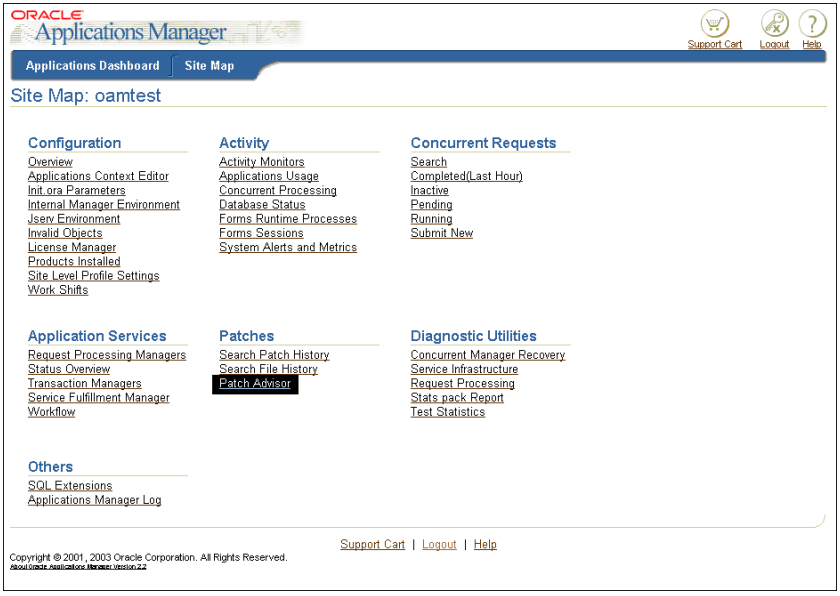
  You can also use Patch Advisor to upload the metadata for a specific patch or set of patches, and then view information reported from the metadata. For example, you can upload the metadata for a patch, and then view any requisite patches that have not yet been applied and the impact of applying this new patch.

Accessing Patch Advisor

You can navigate to the Patch Advisor main screen from the Oracle Applications Manager Dashboard.



You can navigate to the Patch Advisor main screen from the Oracle Applications Manager Site Map.



Choosing Patch Advisor from the Oracle Applications Dashboard or Site Map brings up the Patch Advisor main screen. From this screen, you can set up a staging directory, manage patch filters, submit and review concurrent requests, and view recommended patches.

Defining the Staging Directory

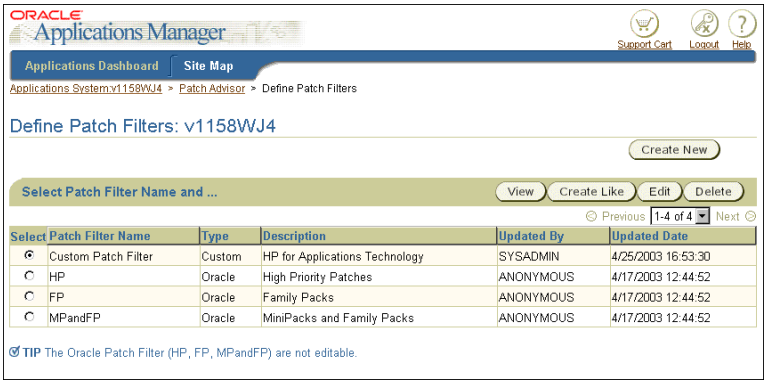
The staging directory is where you store files used by Patch Advisor. Patch Advisor also uses this directory to create temporary files and subdirectories during processing. These temporary files and directories are deleted after processing.

Note: We recommend you pick a staging directory once and use the same directory each time you run Patch Advisor.

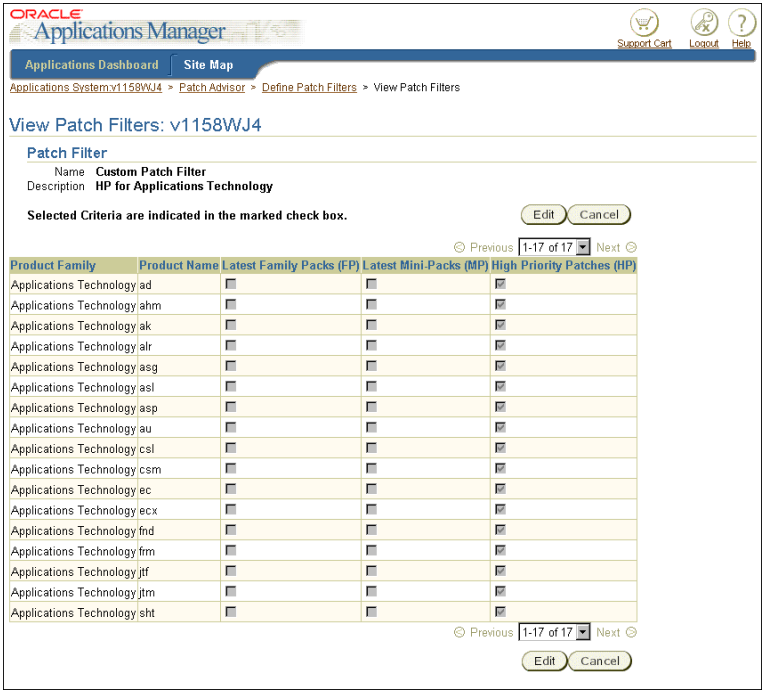
Defining Filters

The InfoBundle11i.zip file contains information for all recommended patches for all products. If the Patch Advisor were to compare patches in the patch history database against all metadata in this file, the number of recommended patches in the report would be large, and the results might not be useful. To avoid this, Patch Advisors provides filters so that only those patch types and products in the metadata that apply to your system are included in the comparison.

Clicking the Define Patch Filters link shows all the filters created for the current system. Patch Advisor has three pre-seeded filters, and you can create custom filters from this Define Filters screen.

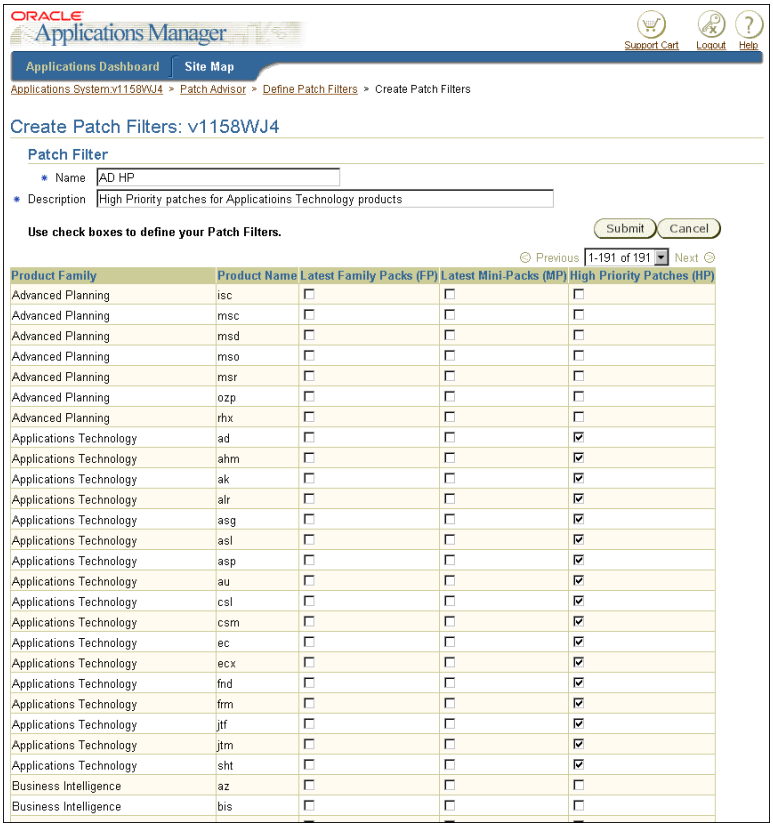


The three pre-seeded filters are MP and FP (mini-Packs and family packs), HP (high-priority patches), and FP (family packs). Patch Advisor creates these three filters by choosing only the mini-packs and family packs, or high-priority patches, or family packs that apply to the set of products you have licensed on your system. For example, if you select the HP filter from the screen, then choose View, a screen similar to the following shows that this filter only contains high-priority patches for the products you licensed.

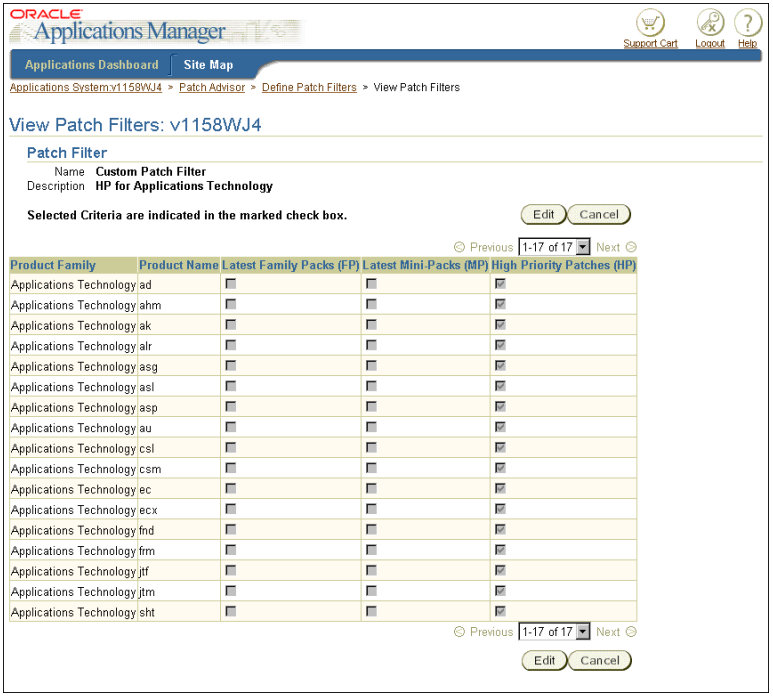


Note that the pre-seeded filters contain "Oracle" in the type column on this screen, and other filters you create contain "Custom" in this column. You cannot edit or delete these three pre-seeded filters, but you can use these three filters as templates to create new filters.

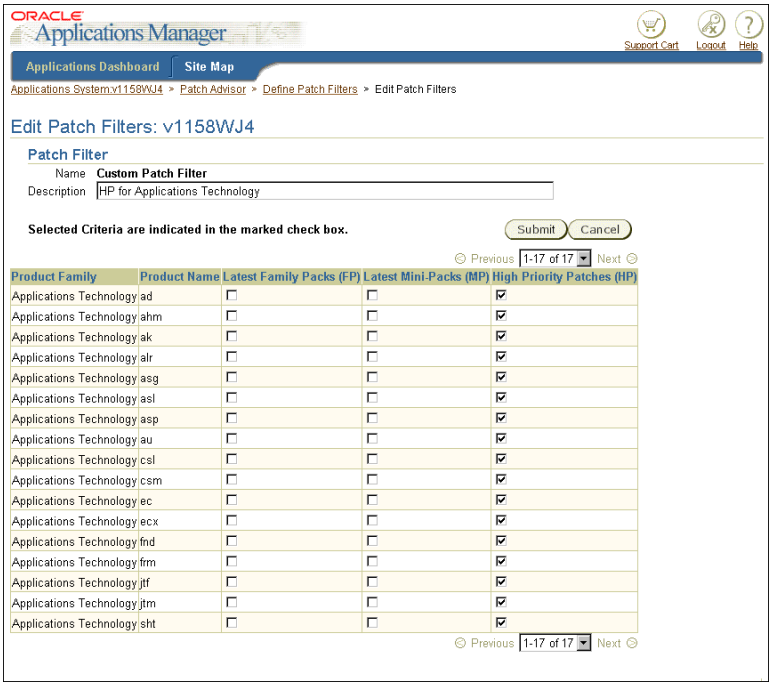
Click Create New to create a new custom filter. You must enter a unique name and a description for each new custom filter. All licensed products are listed on the left of the Create New screen. There are three columns for each product: Latest Family Packs, Latest Mini-packs, and High Priority Patches. Check the appropriate boxes next to each product to include the patches of each type in the new filter.



Choose a filter and click View to see the composition of the filter. All products that are in the filter are listed on the left of the screen, and the check boxes indicate which patches for which products are contained in the filter.



Clicking Edit on this screen allows you to edit the filter by checking or clearing the boxes. (You cannot edit the three pre-seeded filters.)



Submitting Requests

After setting up the Patch Advisor staging area (and optionally creating custom filters), you can submit requests for processing or view previous requests from the Patch Advisor main screen.

The Submit column on the Patch Advisor main screen contains the following actions:

- Upload Patch Information Bundle

  This uploads metadata from the InfoBundle11i.zip file to the Oracle Applications database. You can choose to schedule this upload for a later time or for recurrent runs. If there is no new or updated data in the Infobundle11i.zip file, no data will be uploaded to the database.

- Analyze Specific Patch(es)

  This generates recommendations for specific patches. After downloading specific patches from Oracle*MetaLink* and placing them in the staging area, you can analyze these patches from this screen by entering the patch numbers. You can schedule the analysis for a later time.

- Recommend Patch(es)

  This generates recommendations for specific patches. You choose one of the three pre-seeded filters or any custom filter you created in the Define Patch Filters screen. Patch Advisor uses the filter and compares the patch history database against the metadata patch list to recommend which patches you should apply. You can also schedule this task to run at a later time.

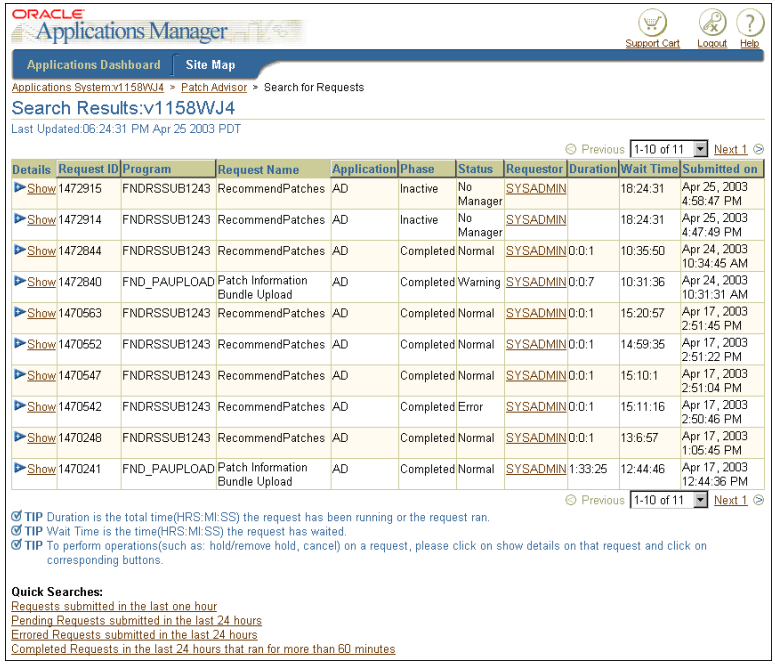
  The View column on the Patch Advisor main screen provides screens that report on previous Submit actions. The View column parallels the Submit

column. For example, if you upload a patch information bundle, you can then view the information about the upload (and about all other previous uploads).

Viewing Requests

The View column the Patch Advisor main screen provides screens that report on previous Submit actions. The View column parallels the Submit column. For example, if you upload a patch information bundle, you can then view the information about the upload (and about all other previous uploads).

The view screens for patch upload requests, specific patch analysis requests, and patch recommendation requests contain similar information. You can also review all requests, which include patch upload, specific patch analysis, and patch recommendation requests. The following view screen is for patch upload requests.



Each time you submit a request to upload a patch information bundle, to analyze specific patches, or to recommend patches, the Patch Advisor creates request ID. The request ID is shown in the second from right column on the screen. The row for each request ID also displays:

- Details column

  You can toggle between showing all details for the request or hiding the details and seeing only the summary line for each detail.

  If there are any errors reported in the Status column, click the View Diagnostics, Manager Log, Request Log, and Output buttons for more details. For reported errors in Specific Patch(es) Analysis Requests and Patch Recommendation Requests, click the SubRequest button. You can drill down to the subrequest that has "error" in the status column, and then click the Request Log button for error details.

- Program, Description, and Application columns

  The Application column is always AD for all Patch Advisor requests. The Program and Description columns identify the program that was run.

- Phase column

  The phase is either completed or incomplete. Incomplete requests are still running.

- Status column

  The status is either normal, warning, or error. You can find more information about why a status was reported by toggling to Show in the Details column, and clicking the View Diagnostics, Manager Log, Request Log, and Output buttons.

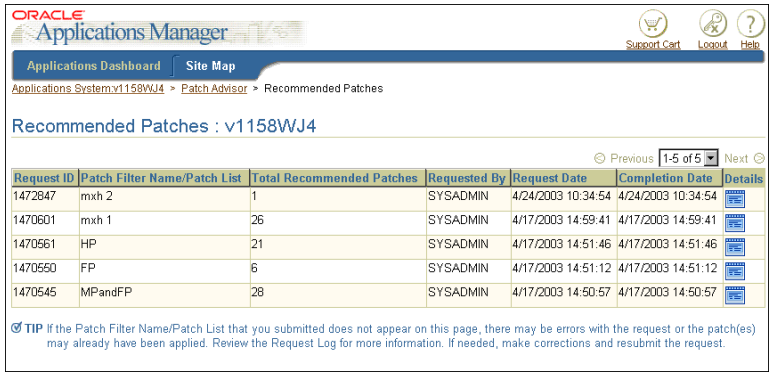
  For Specific Patch(es) Analysis Requests and Patch Recommendation Requests, click the SubRequest button. If there were any errors, you can drill down to the subrequest that caused the error, and then view the Request Log for details.

- Requestor, Duration, Wait Time, and Submitted on columns

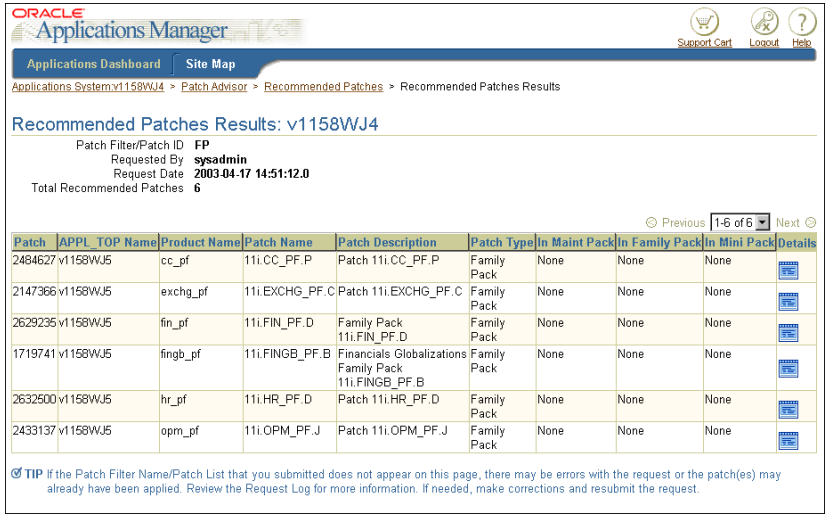
  These columns show who submitted the request, how long the request took to process, the wait time before OAM began processing the request, and the date the request was submitted.

Viewing Recommended Patches

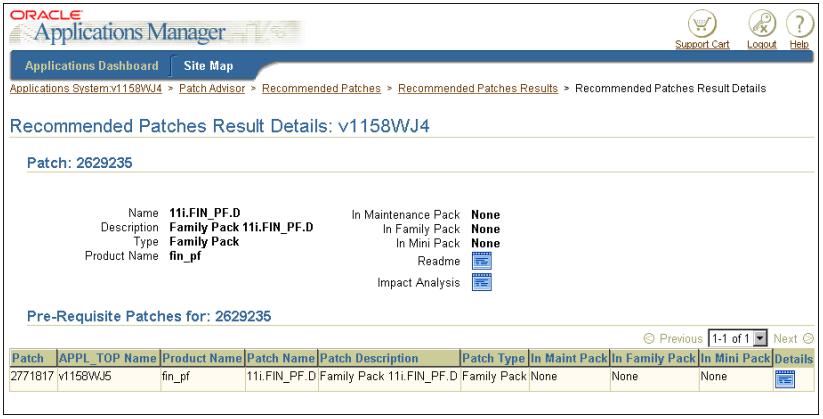
After setting up and submitting a request, the final action is to view the recommended patches. If you submitted a specific patch to analyze, you also click Recommended Patches to view the patch information.



The Patch Filter Name is the name you provided when generating recommendations (or the names of the specific patches you chose to analyze). The screen lists the total number of recommended patches and other information about the run. Click the report icon in the Details column to view the details.



The Recommended Patches Results screen provides the list of recommended patches, patch name, patch type, and tells if the patch is contained in any maintenance pack, family pack, or mini-pack. To see more information about a patch, click the report icon in the Details column of this screen.



From the Details screen of the Recommended Patches Results, you can view the readme file for each patch and view the impact analysis. The impact analysis tells the products and files that are affected by this patch. You can analyze which files are new, which files are changed, and which files are ignored when applying the patch. You can also view an prerequisite patches required by this patch.

Note that if the patch filter or patch list that you submitted does not appear on the Results page, there may be errors with the request, or the patches may already be applied, or no patches were recommended for the selected criteria. Review the request log for more information. If there are errors, make corrections and resubmit the request.

Job Timing Report

AutoUpgrade, AutoPatch, and AD Administration each produce a Job Timing report named adt<session_id>.lst. It is located in $APPL_TOP/admin/<SID>/out (UNIX) or %APPL_TOP%\admin\<SID>\out (Windows). The report provides summary timing information for jobs run by parallel workers, such as:

- Jobs run successfully on the first try
- Failed jobs that were deferred and then run successfully
- Failed jobs that were restarted and then run successfully
- Failed jobs that were skipped
- Time-consuming jobs

- Summary information for each parallel phase

You can also generate a report manually to view timing statistics from a prior session:

UNIX:

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

Windows:

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

The <output file> in these commands should not have an extension. The adtimrpt.sql script creates two files: an .lst file, which is the timing report, and a .csv file, which is currently not used.

AD Configuration (adutconf)

This utility is a SQL script that reports standard information about the installed configuration of Oracle Applications. Run this task in order to debug or document the status of your installation. Running AD Configuration generates a file (adutconf.lst) that contains the following:

- SQL*Plus PAUSE and NEWPAGE settings
- Rollback segment information
- Information about the product group
- Whether Multi-Org is installed
- List of operating units
- Whether Multiple Reporting Currency (MRC) is installed
- List of registered products
- Information on all registered schemas
- Information about all installed products, including shared and dependent products
- Status of localization modules
- The base language and other installed languages

- NLS init.ora settings

Log in as applmgr and set the environment as described in Getting Started in Chapter 1. Use the following command to run this script. The output file is written to adutconf.lst in the current working directory.

UNIX:

```
$ cd $APPL_TOP/admin/<SID>/out
$ sqlplus <APPS schema username>/<APPS schema password> \
  @$AD_TOP/sql/adutconf.sql
```

Windows:

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

AD File Identification (adident)

Use this utility to identify the version and translation level of Oracle Applications files. It is useful when collecting information about your site for Oracle Support Services.

Before starting adident, log in as applmgr and complete the steps in the Setting the Environment section of Chapter 1. Use the following command to run the program:

UNIX:

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

Windows:

```
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 Windows), library file (extension .a for UNIX and .lib for Windows), dynamic link library (.dll for Windows), or executable program (.exe for Windows). 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:

UNIX:

```
$ 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
  ....
```

Windows:

```
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).

Index

A

B

C

L

M

O

P

R

S

T

U

V

W